

# Generalized Advanced Propeller Analysis System (GAPAS)

## Volume II – Computer Program User Manual

December 1986

Prepared by  
L. Glatt  
D.R. Crawford  
J. B. Kosmatka  
R. J. Swigart  
E. W. Wong

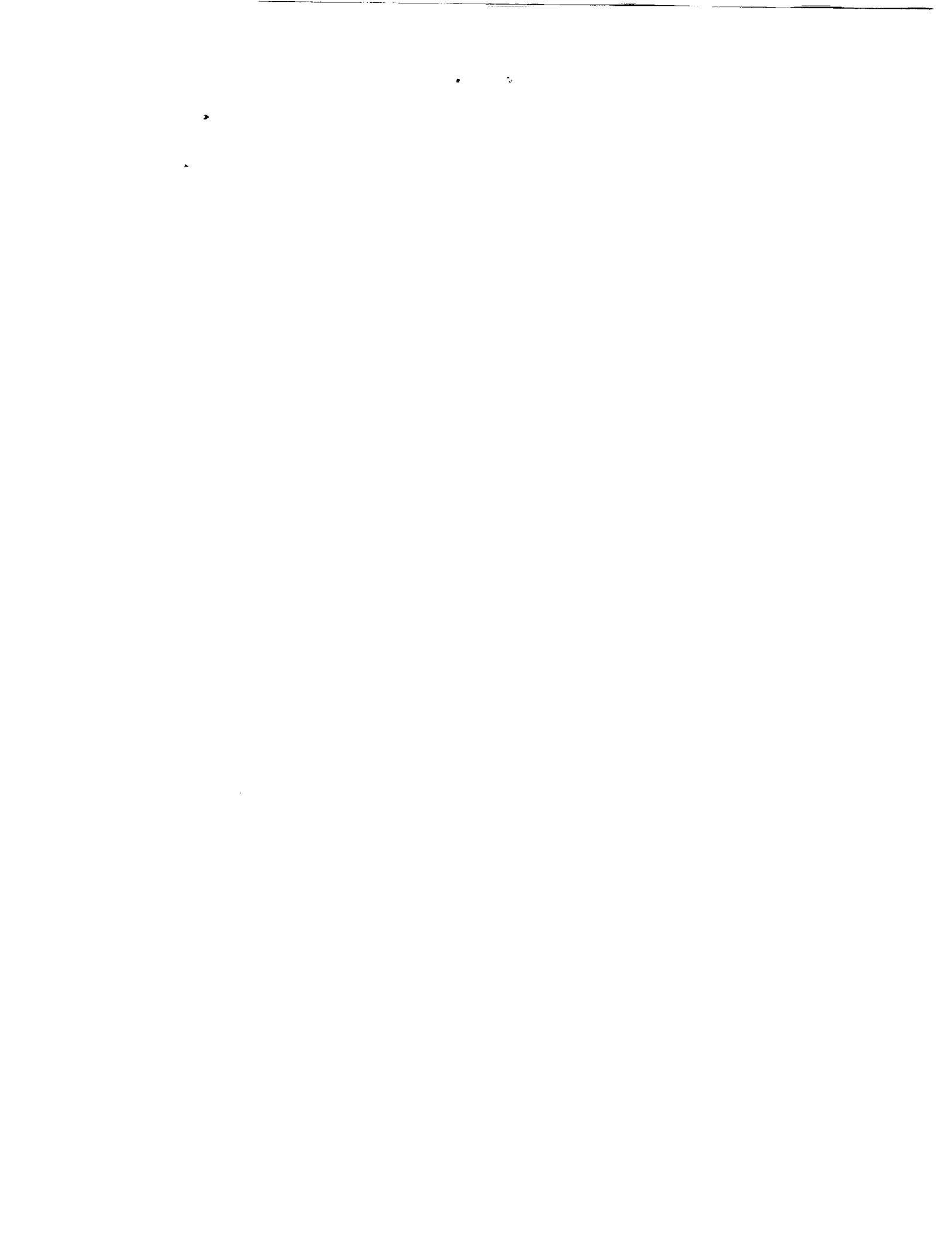
**TRW Space & Defense Sector**  
Redondo Beach, CA

Prepared for  
**National Aeronautics and Space Administration**  
NASA-Lewis Research Center  
Contract NAS3-22251

(NASA-CR-185277) GENERALIZED ADVANCED  
PROPELLER ANALYSIS SYSTEM (GAPAS). VOLUME 2:  
COMPUTER PROGRAM USER MANUAL (TRW Defense  
and Space Systems Group) 484 p CSCL 01C

N90-29394

Unclass  
63/07 0305448



# **Generalized Advanced Propeller Analysis System (GAPAS)**

## **Volume II – Computer Program User Manual**

**December 1986**

Prepared by  
L. Glatt  
D.R. Crawford  
J. B. Kosmatka  
R. J. Swigart  
E. W. Wong

**TRW Space & Defense Sector**  
Redondo Beach, CA

Prepared for  
**National Aeronautics and Space Administration**  
NASA-Lewis Research Center  
Contract NAS3-22251





## SUMMARY

This report describes the Generalized Advanced Propeller Analysis System (GAPAS) computer code. GAPAS has been developed to analyze advanced technology multi-bladed propellers which operate on aircraft with speeds up to Mach 0.8 and altitudes up to 40,000 feet. GAPAS includes technology for analyzing aerodynamic, structural and acoustic performance of propellers. The computer code has been developed for the CDC 7600 computer and is currently available for industrial use on the NASA Langley computer.

A description of all the analytical models incorporated in GAPAS is included. Sample calculations are also described as well as users requirements for modifying the analysis system. Computer system core requirements and running times are also discussed.



## CONTENTS

	Page
1. INTRODUCTION	1-1
2. MODULE DESCRIPTION	2-1
2.1 Geometry Generator Module	2-1
2.1.1 Problem Definition	2-1
2.1.2 Propeller Geometry	2-4
2.1.3 Coordinate Systems and Transformation Matrices	2-11
2.1.4 Cubic Spline Interpolation	2-14
2.1.5 Blade Geometry Definition	2-19
2.1.6 Intersection of a Wire with a Plane Perpendicular to an Arbitrary Line	2-22
2.1.7 Structural Module Requirements	2-22
2.1.8 Propeller Performance Modules Requirements	2-26
2.1.9 Blade Reflection - Iterative Calculation	2-26
2.1.10 G400 Interface	2-28
2.1.11 Geometry Generator Subroutine Summary	2-32
2.2 Aircraft Flow Field	2-32
2.2.1 Problem Definition	2-32
2.2.2 Aircraft Flow Field Methodology	2-37
2.2.3 Numerical Technique	2-37
2.2.4 Velocity Components in the Propeller Plane	2-38
2.2.5 Options and Limitations	2-39
2.2.6 Input Parameters	2-40
2.2.7 Subroutine Descriptions and Calling/Called Routines	2-42
2.3 Aerodynamic Performance Module	2-42
2.3.1 Problem Definition	2-42
2.3.2 Method of Solution	2-43
2.3.3 Blade Geometry	2-45
2.3.4 Governing Equations	2-50
2.3.5 Boundary Conditions	2-52
2.3.6 Detailed Solution Methodology	2-53
2.3.6.1 Thrust Coefficient	2-55
2.3.6.2 Power Coefficient	2-57
2.3.6.3 Radial Force Coefficient	2-58
2.3.7 Numerical Techniques	2-58
2.3.7.1 Convergence Analysis	2-61
2.3.7.2 Truncation Error	2-63
2.3.7.3 Error of Integration	

## CONTENTS (Continued)

		Page
2.3.8	Dictionary of Input Variables	2-65
2.3.9	Program Capabilities; User Options	2-67
2.3.10	Program Validation	2-68
2.4	Airfoil Loading Module	2-72
2.4.1	Airfoil Analysis Method: TRANSEP	2-72
2.4.1.1	Problem Formulation	2-75
2.4.1.2	Numerical Analysis	2-76
2.4.1.3	Numerical Stability	2-90
2.4.1.4	Numerical Studies	2-96
2.4.1.5	Viscous Boundary Layer and Wave Drag Correction	2-98
2.4.1.6	Massive Separation Model	2-102
2.4.1.7	Input Variable Description	2-110
2.4.1.8	Output Description	2-118
2.4.2	Airfoil Data Banks	2-121
2.4.2.1	Clark-Y Series	2-121
2.4.2.2	NACA 16 Series	2-124
2.5	Acoustics Module	2-128
2.5.1	Problem Definition	2-128
2.5.2	Subsonic Propeller Noise (SPN)	2-128
2.5.2.1	Acoustic Formulation	2-132
2.5.2.2	Frames of Reference	2-143
2.5.2.3	Computational Strategy	2-147
2.5.2.4	Retarded Time Equation	2-149
2.5.2.5	Full Blade Formulation	2-152
2.5.2.6	Mean Surface Approximation	2-155
2.5.2.7	Compact Chord Approximation	2-161
2.5.2.8	Point Source Approximation	2-162
2.5.2.9	Roots of Retarded Time Equation	2-166
2.5.2.10	Acoustic Spectrum Calculation	2-169
2.5.2.11	SPN Input	2-170
2.5.2.12	Program Validation	2-172
2.5.3	Elliptic Coordinate System	2-184
2.5.4	Transonic Propeller Noise Module	2-192
2.5.4.1	Acoustic Formulation	2-196
2.5.4.2	Frames of Reference	2-198
2.5.4.3	Computational Strategy	2-200
2.5.4.4	Retarded Time Equation	2-202

## CONTENTS (Continued)

	Page	
2.5.4.5	TPN Input	2-204
2.5.4.6	TPN Program Validation	2-207
2.6	Structural Beam Module	2-222
2.6.1	Basic Assumptions Used in the Analysis	2-232
2.6.2	Geometric Preliminaries	2-237
2.6.2.1	Transformation Relations Between the Coordinate Systems	2-238
2.6.2.2	Initial Curvatures and Twist of Beam	2-241
2.6.2.3	Ordering Scheme for a Long Slender Beam	2-242
2.6.3	Development of Strain Components	2-244
2.6.3.1	Outline of the Calculation of the Deformations and Strains	2-244
2.6.3.2	Undeformed Base Vector Development	2-246
2.6.3.3	Deformed Base Vector Development	2-247
2.6.3.4	Strain Components	2-257
2.6.4	Derivation of the Equations of Motion by Hamilton's Principle	2-259
2.6.4.1	Strain Energy Contributions	2-260
2.6.4.2	Kinetic Energy Contributions	2-269
2.6.4.3	External Work Contributions	2-276
2.6.4.4	Summary of the Partial Differential Equations of Motion and Boundary Conditions	2-277
2.6.5	Finite Element Discretization of the Equations of Motion Using Hamilton's Principle	2-279
2.6.5.1	Finite Elements Associated With the Strain Energy Variation	2-284
2.6.5.2	Finite Elements Associated With the Kinetic Energy Variation	2-286
2.6.5.3	Finite Elements Associated With the Virtual Work Variation	2-287
2.6.5.4	Summary of Finite Element Equations	2-287
2.6.6	Method of Solution	2-288
2.6.6.1	Static Analysis	2-288
2.6.6.2	Structural Dynamic Analysis	2-290

## CONTENTS (Continued)

	Page	
2.6.7	Determination of the Shear Center and Structural Constants of a Cross Section of Arbitrary Shape	2-291
2.6.7.1	Basic Assumptions Used in the Analysis	2-292
2.6.7.2	Total Potential Energy of a Tip Loaded Cantilever Beam	2-292
2.6.7.3	Solution of the St. Venant Functions by the Finite Element Method	2-297
2.6.7.4	Location of the Shear Center	2-299
2.6.7.5	Determination of the Torsional Constant and the Warping Constants	2-302
2.6.7.6	Sample Calculations	2-303
2.6.8	Numerical Results	2-307
2.6.9	General Description of Program Structure	2-322
3.	GAPAS SYSTEM	3-1
3.1	System Design	3-1
3.2	GAPAS Data Base	3-2
3.3	GAPAS Executive System Control Statements	3-4
3.3.1	Control Statement Format	3-4
3.3.2	Description of GAPAS Control Statements	3-6
3.4	Input/Output Requirement for Functional Modules	3-69
3.5	System Running Procedure	3-92
3.6	Input File Description	3-103
3.7	Computer Resources Requirements	3-114

## CONTENTS (Continued)

	Page
APPENDIX	
A    FINITE ELEMENT MATRICES ASSOCIATED WITH THE BEAM MODEL DEVELOPMENT	A-1
A.1    Finite Element Definition of the Strain Energy Variation	A-1
A.2    Finite Element Definition of the Kinetic Energy Variation	A-14
A.3    Finite Element Definition of the External Work Variation	A-20
REFERENCES	R-1

## ILLUSTRATIONS

	Page	
2.1-1	GAPAS as Seen by Geometry Generator	2-2
2.1-2	SR-3 Propeller-Nacelle Geometry	2-3
2.1-3	Example of Transformation Matrices	2-5
2.1-4	Coordinate Transformations Used in GAPAS	2-6
2.1-5	Transformation From Wind Axes to Body-Fixed Frame	2-8
2.1-6	Transformation From Body-Fixed Frame to Prop-Hub Frame	2-8
2.1-7	Transformation from Prop-Hub Frame to Pitch-Change Axes	2-10
2.1-8	Summary of Coordinate Systems	2-12
2.1-9	Cubic Spline Interpolation	2-13
2.1-10	Sweep Angle Definitions for Clockwise Rotation	2-16
2.1-11	Sweep Angle Definitions for Counterclockwise Rotation	2-17
2.1-12	Airfoil Section Local Coordinate System (LCS)	2-18
2.1-13	SR-3 Propeller Geometry	2-20
2.1-14	Three-View of SR-3 Propeller Blade Cuts Perpendicular to Line of Shear Centers	2-21
2.1-15	Camber Line and Quarter-Chord Point	2-24
2.1-16	Find Control Point, Load Point and Trailing Vortex Point	2-25
2.1-17	Undeformed and Deformed Line of Shear Centers	2-27
2.2-1	Aircraft Flow Field Effects	2-33
2.2-2	Sample Body Cross Section Input Showing a Repeated Cross Section Description and Indicating the Input Sequence of Periphery Points	2-35
2.2-3	Distributed Sources at I Panel and Point Source at J Panel	2-35
2.3-1	Propeller Coordinates System	2-44



## ILLUSTRATIONS (Continued)

		Page
2.3-2	Vortex Lattice Elements for a Four-Blade Propeller	2-46
2.3-3	Variation of FUHINT Versus $\theta$ from 0 to 0.40 Radian	2-59
2.3-4	Variation of FUHINT Versus $\theta$ from 1 to 20 Radian	2-59
2.3-5	Aerodynamic Performance of SR-3 Propeller	2-69
2.3-6	Aerodynamic Performance of SR-3 Propeller	2-70
2.3-7	Aerodynamic Performance of SR-3 Propeller	2-71
2.4-1	Flow Field Subdivision for Coordinate Stretching	2-77
2.4-2	Typical Grid System (25 x 25)	2-79
2.4-3	Relationship Between Airfoil and Grid (Upper Surface)	2-86
2.4-4	Computational Grid - Infinity Points	2-88
2.4-5	Grid Pattern Versus $X_4$	2-97
2.4-6	Problem Formulation	2-104
2.4-7	$C_{p_{sep}}$ and $\Gamma$ Formulation	2-106
2.4-8	Separation Point and Pressure Behavior During Relaxation Process	2-108
2.5-1	Transformation of Coordinates	2-145
2.5-2	Geometry of Source and Observer Position Used for Calculation of Emission Time	2-150
2.5-3	Blade Surface Coordinate System	2-154
2.5-4	Roots of the Retarded Time Equation	2-168
2.5-5	Wing-Mounted Microphone Position 101/16 Propeller Acoustic Measurements Near Field	2-173
2.5-6	Acoustic Pressure Signature 101/16 Propeller Near Field (Run No. 11)	2-174

## ILLUSTRATIONS (Continued)

		Page
2.5-7	Overall Noise Spectra 101/16 Propeller Near Field (Run No. 11)	2-175
2.5-8	Acoustic Pressure Signature 101/16 Propeller Near Field (Run No. 12)	2-176
2.5-9	Overall Noise Spectral 101/16 Propeller Near Field (Run No. 12)	2-177
2.5-10	Acoustic Pressure Signature 101/16 Propeller Near Field (Run No. 13)	2-178
2.5-11	Overall Noise Spectra 101/16 Propeller Near Field (Run No. 13)	2-179
2.5-12	Acoustic Verification Results SR-3 Propeller	2-181
2.5-13	Propeller Plane Near-Field Pressure Signature at 0.8 D Tip Clearance Radial Location; SR-3 Propeller, $N_b = 2$ , $M_0 = 0.203$ , 6700 rpm, $M_{TH} = 0.687$	2-182
2.5-14	8P Harmonic Sound Pressure Level at 0.8 D Tip Clearance; SR-3 Propeller, $N_b = 2$ , $M_0 = 0.203$ , 6700 rpm, $M_{TH} = 0.687$	2-183
2.5-15	8P Harmonic Sideline Directivity at 4.4 D Tip Clearance SR-3, $N_b = 4$ , $M_0 = 0.321$ , 8550 rpm, $M_{TH} = 0.901$	2-185
2.5-16	8P Harmonic Sideline Directivity at 4.4 D Tip Clearance SR-3, $N_b = 2$ , $M_0 = 0.203$ , 7500 rpm, $M_{TH} = 0.761$	2-186
2.5-17	Typical Airfoil Section in Complex z-Plane	2-189
2.5-18	Graph of Typical Transonic Retarded Time Equation	2-203
2.5-19	Exploded View of Retarded Time Equation	2-205
2.5-20	Pressure Transducer Positions SR-3 Propeller	2-208
2.5-21	Acoustic Pressure Signature SR-3 Propeller Near Field	2-209
2.5-22	Overall Noise Spectra SR-3 Propeller Near Field	2-210

## ILLUSTRATIONS (Continued)

		Page
2.5-23	Acoustic Pressure Signature SR-3 Propeller Near Field	2-211
2.5-24	Overall Noise Spectra SR-3 Propeller Near Field	2-212
2.5-25	Acoustic Pressure Signature SR-3 Propeller Near Field	2-213
2.5-26	Overall Noise Spectra SR-3 Propeller Near Field	2-214
2.5-27	Acoustic Pressure Signature SR-3 Propeller Near Field	2-215
2.5-28	Overall Noise Spectra SR-3 Propeller Near Field	2-216
2.5-29	Overall Sound Pressure Level SR-3 Propeller Near Field	2-217
2.5-30	Definition of Loading Parameter, $a$	2-218
2.5-31	Retarded Time Equation	2-219
2.5-32	Retarded Time Equation	2-220
2.6-1	Advanced Propeller (Prop-Fan)	2-223
2.6-2	Propeller Fixed and Beam Element Coordinate Systems	2-224
2.6-3	Inertial and Hub-Fixed Coordinate Systems	2-233
2.6-4	Hub-Fixed and Propeller-Fixed Coordinate Systems Including Sweep ( $\beta_S$ ), Precone ( $\beta_C$ ), and Pitch ( $\beta_P$ ) Adjustments	2-234
2.6-5	Element and Curvilinear Coordinate Systems of Pretwisted Beam Element	2-235
2.6-6	Beam-Type Finite Element	2-236
2.6-7	Position of a Material Particle Before and After the Deformation	2-245
2.6-8	Curvilinear Coordinate System After Deformation and Euler Angles	2-250
2.6-9	Cantilever Beam	2-293
2.6-10	Shear Center Location on the Beam Cross Section	2-300

## ILLUSTRATIONS (Continued)

		Page
2.6-11	Analysis of an Isotropic, Rectangular Cross Section	2-304
2.6-12	L-Shaped Cross Section	2-305
2.6-13a	Profile of SR-3 Turbo-Propeller at Location 25% from Hub	2-306
2.6-13b	Warping Function Distribution of a NASA SR-3 (25% from Blade Root)	2-308
2.6-14a	Profile of SR-3 Turbo-Propeller at Blade Root	2-309
2.6-14b	Profile of SR-3 Turbo-Propeller at Blade Tip	2-310
2.6-15	Tip-Loaded Analysis of Curved Cantilever Beam	2-311
2.6-16	Tip Displacements for Tip-Loaded Curved Cantilever Beam. Applied axial force (top) and in-plane transverse shear force (bottom).	2-312
2.6-17	Tip Displacements for Tip-Loaded Curved Cantilever Beam. Applied out-of-plane transverse force (top) and torsion moment (bottom).	2-313
2.6-18	Tip Displacements for Tip-Loaded Curved Cantilever Beam. Applied out-of-plane bending moment (top) and in-plane bending moment (bottom).	2-314
2.6-19	Natural Frequency Calculation of a Clamped-Clamped Circular Arc	2-316
2.6-20	Natural Frequency Calculation of a Cantilever Beam with In-Plane Curvature	2-317
2.6-21	Natural Frequency Calculation of a Cantilever Beam with Out-of-Plane Curvature	2-319
2.6-22	Natural Frequency Calculation of Pretwisted Cantilever Beam	2-320
2.6-23	Uniform Cantilevered Beam Under Transverse Loading	2-321
2.6-24	Flowchart of Nonlinear Structural Dynamic Analysis	2-323

## ILLUSTRATIONS (Continued)

	Page
3.4-1	Definition of ATRAN(I) and BTRAN(I) 3-86
3.5-1	Compilation of GAPAS Functional Modules 3-94
3.5-2	Linking GAPAS 3-95
3.5-3	Overlay Structure 3-96
3.5-4	Execution of GAPAS 3-102
3.6-1	Input File for NASA Lewis SR-3 Propeller 3-104
3.6-2	Input File for A Single-Pass Mode Operation 3-111

## TABLES

		Page
2.3-1	Values of FUHINT, FVHINT, and FWHINT at Different Revolutions of Integration	2-60
2.3-2	Values of $C_T$ , $C_D$ , and $\eta$ at Different Revolutions of Integration SR-2, $J = 3.06$ , $\beta = 60^\circ$	2-65
2.5-1	OASPL Results - 101/16 Propeller Near-Field Conditions	2-177
3.3-1	ANOPP CS Initialization and System Parameters	3-7
3.3-2	SETSYS CS System Parameters	3-10
3.3-3	RSTRT CS Initialization and System Parameters	3-66
3.7-1	GAPAS Limitations and Requirements	3-115

## ACKNOWLEDGEMENTS

We would like to acknowledge Professors Ken Korkan, Texas A&M University, Barnes McCormick, Penn State University, and Peritz Friedmann, University of California, Los Angeles, for their helpful suggestions during the period of the contract.





## 1. INTRODUCTION

This report describes the Generalized Advanced Propeller Analysis System (GAPAS) computer program developed for the NASA Langley CDC 7600 computer system. GAPAS has been developed in a unified but modular form. The current version of GAPAS contains a set of computer codes which addresses propeller performance, airfoil loading, aircraft flowfield effects, propeller acoustics, and propeller structural analysis. At the present time an aeroelasticity module for analyzing advanced propellers is unavailable, however, many of the required links to the other modules have already been provided. In addition, this version of GAPAS will only address solid propellers of one material. This limitation can be removed by the appropriate modification of the Geometry Generator module.

Chapter 2 describes the details of each of the individual modules including a summary of the governing equations, boundary conditions, initial conditions, numerical techniques, options, limitations, etc. Chapter 3 describes the GAPAS executive system, compiler options, input/output variables, tape and file assignments, storage requirements, sample deck setups for test cases, typical running times, and operating limitations.

Although the current version of GAPAS contains only one analysis code for each discipline, the code has been developed in modular form and the existing modules can either be modified or replaced with newer state-of-the-art analysis tools. Additional analysis codes can also be incorporated such that the user has a choice of which codes to use in each discipline. As an example, the user may prefer to use a less sophisticated analysis tool for some preliminary design studies, after which, he may then utilize one of the sophisticated analysis codes for a more detailed analysis.



## 2. MODULE DESCRIPTION

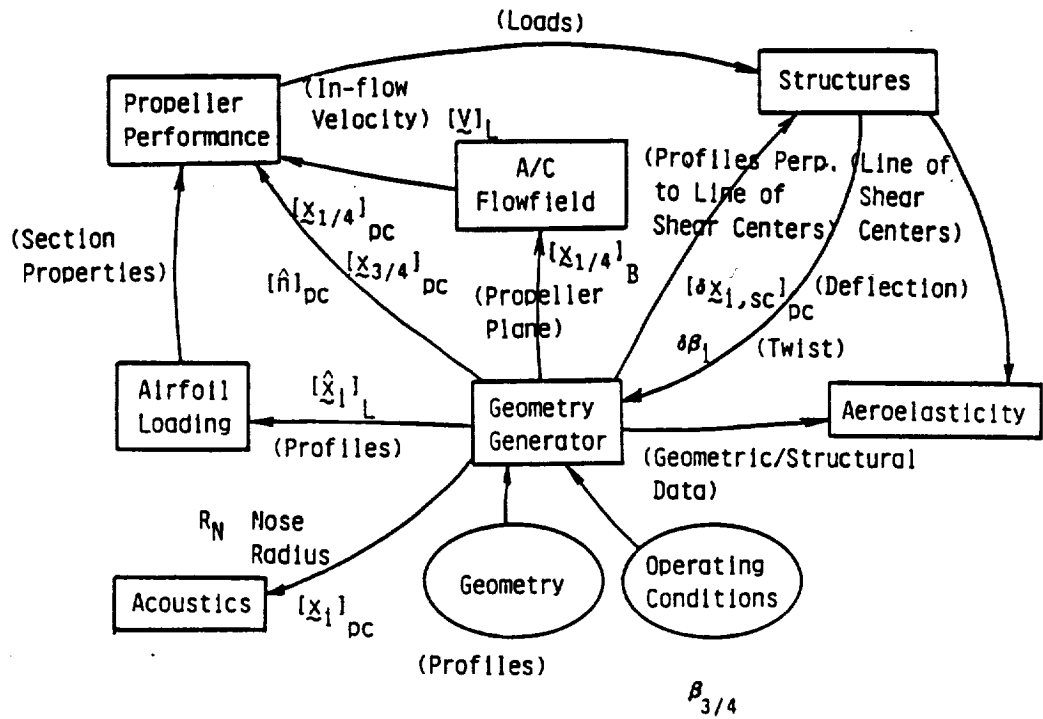
### 2.1 GEOMETRY GENERATOR MODULE

#### 2.1.1 Problem Definition

The geometry generator module is necessary to define the shape of the the propeller and to calculate quantities that are passed to the aircraft flow field, propeller performance, airfoil loading, acoustics, structures and aeroelasticity modules through the data base. The interplay among the modules as seen by the geometry generator module is indicated schematically in Figure 2.1-1, where the symbols represent position vectors, normal vectors, deflections and angles. The geometry generator module was developed at TRW based on the geometrical requirements of the other modules.

#### 2.1.2 Propeller Geometry

The propeller geometry is typically specified by local airfoil profiles that are input perpendicular to a given arbitrary line. In the cases considered thus far, the arbitrary line has been taken to be the pitch-change axis (PCA), but it can be any smoothly varying line in space -- an important generalization for a future design mode and when defining the blade after aeroelastic deformation. The leading edge of the profile with respect to the arbitrary line is input along with the local chord length and the local twist angle relative to the plane of rotation. The input profiles are given with arbitrary chordwise spacing at each input slice, but are then redefined with spacing appropriate to the program option being exercised. The coordinates at comparable points on each profile are connected by "wires" from the hub to the tip, as depicted in Figure 2.1-2 for the SR-3 propeller blade. Also shown in the figure is the geometry of the nacelle defined in the aircraft flow field module. Taking slices, we can find the intersection of each wire with a plane defined perpendicular to a particular line. Then, connecting the wire intersections, we can define the profile in the plane of the slice. The transformation matrices that are used to express the coordinates in another reference frame are accessible through the data base and are transferred to the modules that require them. They define the in-plane and out-of-plane sweep angles, and the local twist angle.



- List of symbols for Figure 2.1-1.
- $\{x_{1/4}\}_{pc}$ ,  $\{x_{1/4}\}_B$  quarter-chord point (defines the propeller plane)
  - $\{x_{3/4}\}_{pc}$  control point on 3/4-chord drive
  - $\{\hat{n}\}_{pc}$  normal vector at control point
  - $\{x_i\}_L$ ,  $\{x_i\}_{pc}$  airfoil profile points
  - $\beta_{3/4}$  twist angle at 3/4 radius location
  - $\{v\}_L$  inflow velocity in the propeller plane
  - $\delta\beta_i$  twist increment due to aerodynamic and inertial loads
  - $\{\delta x_{i,sc}\}_{pc}$  deflection of line of shear centers due to aerodynamic and inertial loads
- Subscripts
- pc quantity measured in pitch-change frame
  - B quantity measured in body-fixed coordinates
  - L quantity measured in local frame

Figure 2.1-1. GAPAS as Seen by Geometry Generator

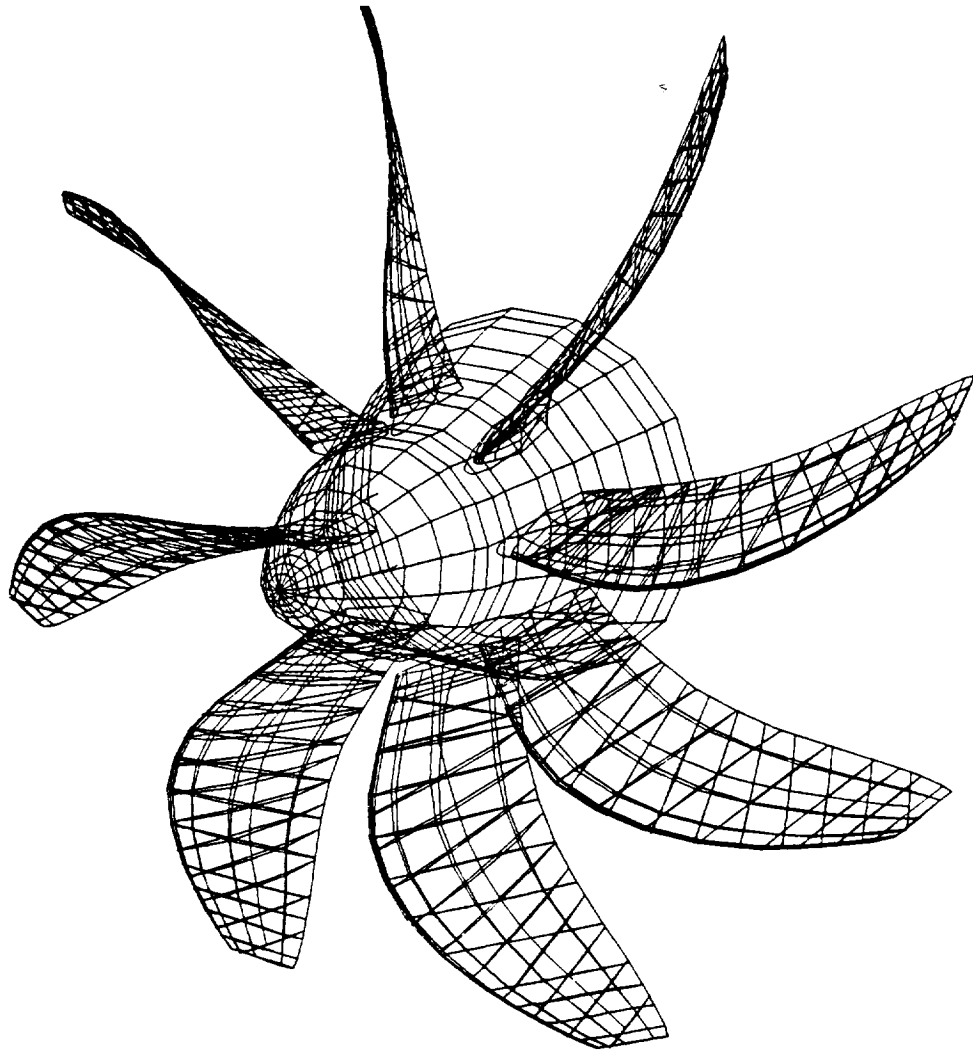


Figure 2.1-2. SR-3 Propeller-Nacelle Geometry

In the performance mode it is important to have enough wires in the nose region so that the profiles calculated from slices perpendicular to the quarter chord can be analyzed by the airfoil loading module. On the other hand, the structural module operates better with clockwise elements that are roughly equally spaced in order to calculate an accurate value for the profile shear center.

### 2.1.3 Coordinate Systems and Transformation Matrices

It is important to be able to transform the coordinates from one reference frame to another. Each transformation matrix is a combination of rotations about the x-, y-, or z-axes, represented by the matrices  $L_1$ ,  $L_2$ , and  $L_3$ ,

$$L_1(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \quad (2.1-1)$$

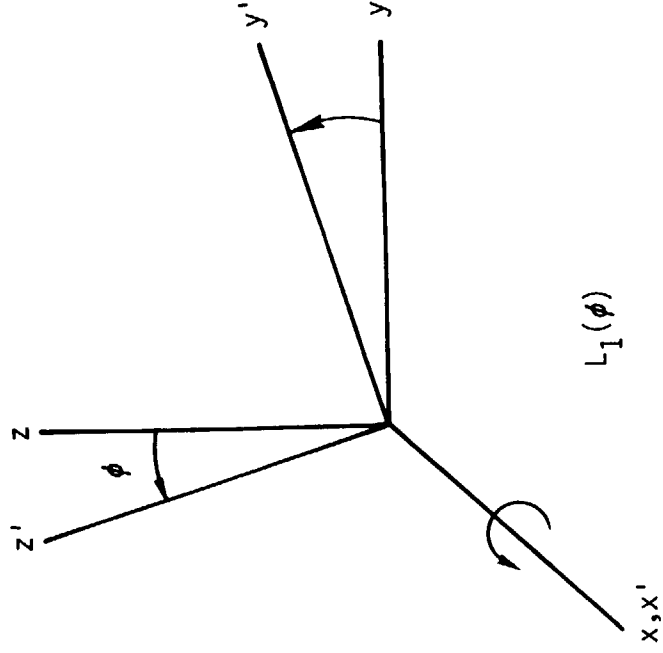
$$L_2(\phi) = \begin{bmatrix} \cos(\phi) & 0 & -\sin(\phi) \\ 0 & 1 & 0 \\ \sin(\phi) & 0 & \cos(\phi) \end{bmatrix} \quad (2.1-2)$$

and

$$L_3(\phi) = \begin{bmatrix} \cos(\phi) & \sin(\phi) & 0 \\ -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.1-3)$$

Figure 2.1-3 shows an example of rotation about the x-axis,  $L_1(\phi)$ .

The transformations necessary to proceed from the wind-axes frame to the local coordinates are indicated in Figure 2.1-4. Starting with a wind axis frame where the x-axis is aligned with the velocity vector of the center of gravity of the body, the y-axis is horizontal to the right, and the z-axis is downward to give a right-handed coordinate system, we can define the transformation to the body-fixed reference frame in terms of the sideslip angle,  $\beta$ , and the angle of attack,  $\alpha$ ,



$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = L_1(\phi) \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x \\ y \cos\phi + z \sin\phi \\ -y \sin\phi + z \cos\phi \end{bmatrix}$$

Figure 2.1-3. Example of Transformation Matrices

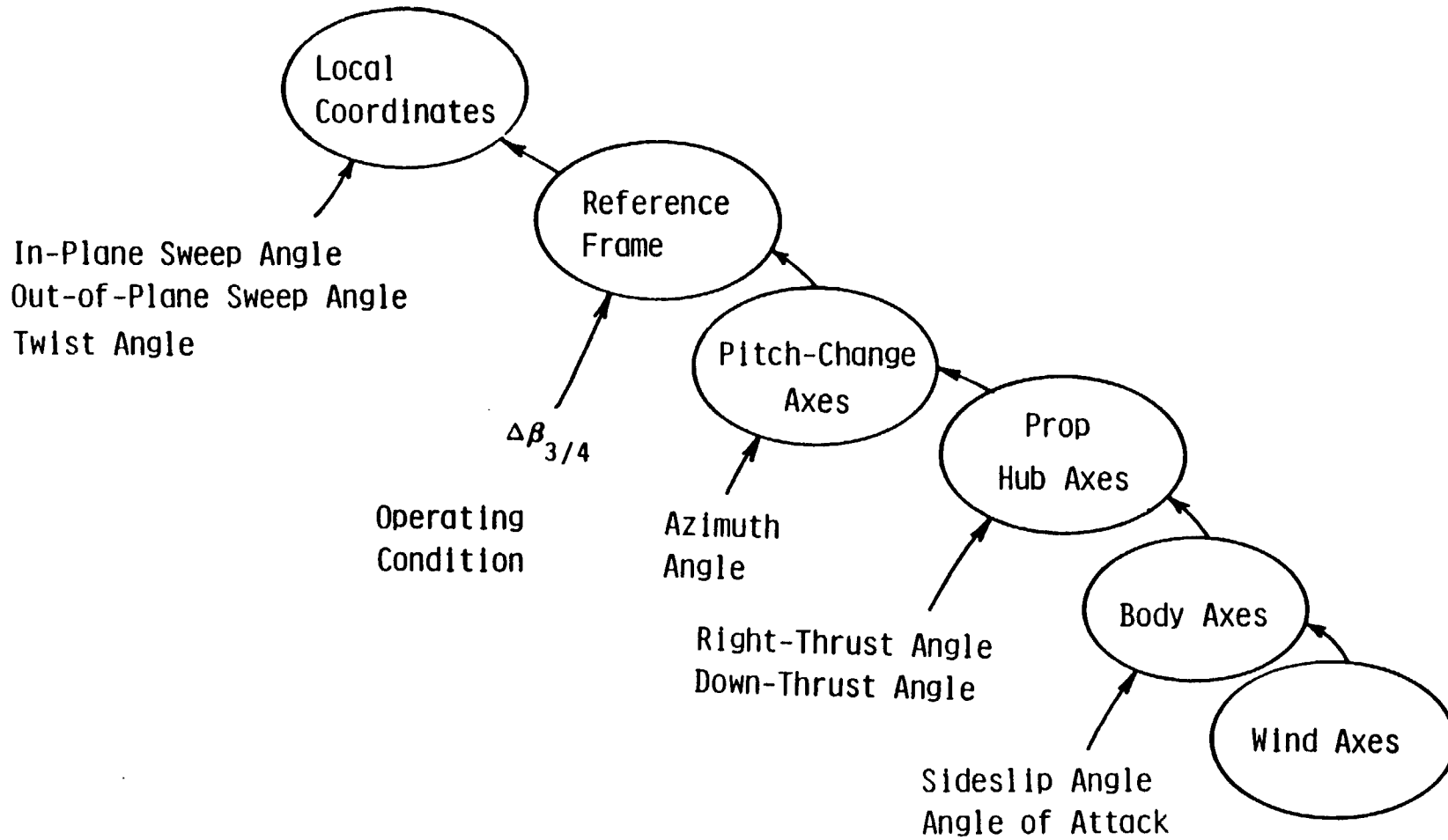


Figure 2.1-4. Coordinate Transformations Used in GAPAS



$$L_{B,W} = L_2(\alpha) L_3(-\beta) \quad (2.1-4)$$

That is, we yaw to the left about the z-axis with angle  $\beta$  and pitch up about the y-axis with angle  $\alpha$ . Multiplied out with the row by column rule for matrix multiplication, this transformation matrix becomes

$$L_{B,W} = \begin{bmatrix} \cos(\alpha)\cos(\beta) & -\cos(\alpha)\sin(\beta) & -\sin(\alpha) \\ \sin(\beta) & \cos(\beta) & 0 \\ \sin(\alpha)\cos(\beta) & -\sin(\alpha)\sin(\beta) & \cos(\alpha) \end{bmatrix} \quad (2.1-5)$$

Since the transformation matrices are orthonormal, the inverse transformation (for the airplane body-fixed axes to the wind axes) is easily found by taking the transpose of  $[L_{B,W}]$ ,

$$L_{W,B} = L_{B,W}^T = L_3(\beta) L_2(-\alpha) \quad (2.1-6)$$

or

$$L_{W,B} = \begin{bmatrix} \cos(\alpha)\cos(\beta) & \sin(\beta) & \sin(\alpha)\cos(\beta) \\ -\cos(\alpha)\sin(\beta) & \cos(\beta) & -\sin(\alpha)\sin(\beta) \\ -\sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix} \quad (2.1-7)$$

The geometry of the transformation from the wind axes to the body-fixed frame is described in Figure 2.1-5.

The propeller-hub axes are required in order to define the propeller plane and determine the induced velocity components for use in the propeller performance analysis. The right-thrust angle,  $\beta_r$ , and down-thrust angle,  $\alpha_d$ , define the transformation to the propeller hub axes from the body-fixed axes. Therefore,

$$L_{PH,B} = L_2(-\alpha_d)L_3(\beta_r) \quad (2.1-8)$$

The angles involved in the transformation from the body-fixed frame to the propeller-hub axes are shown in Figure 2.1-6.

The transformation from the prop-hub axes to the pitch-change axes first involves a quarter-rotation about the z-axis so that the y-axis

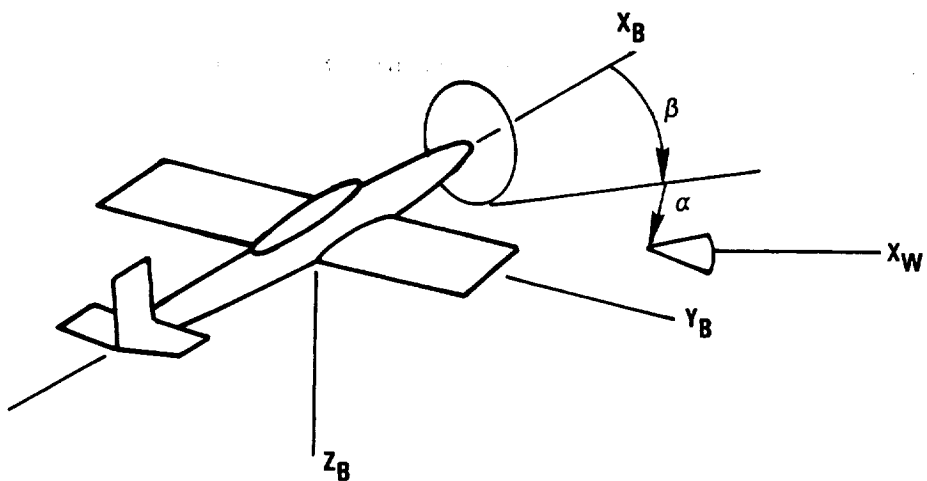


Figure 2.1-5. Transformation From Wind Axes to Body-Fixed Frame

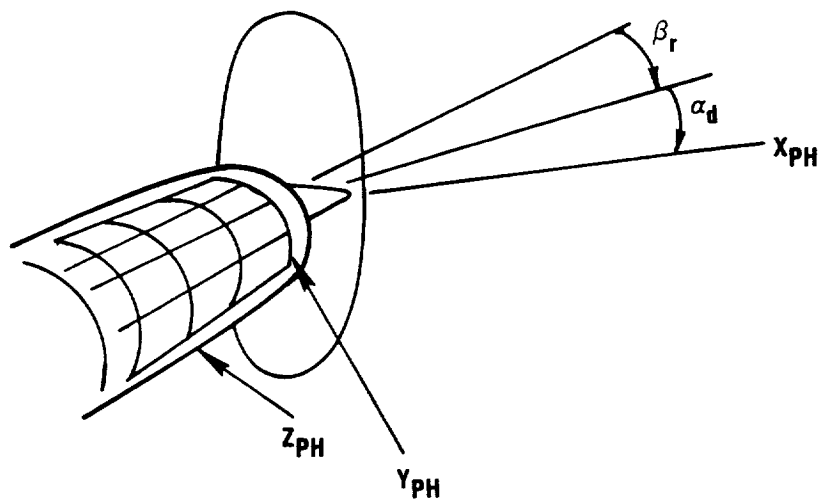


Figure 2.1-6. Transformation From Body-Fixed Frame to Prop-Hub Frame

points along the rotational axis of the propeller. Then, the blade is rotated through the azimuth angle,  $\psi$ ,

$$L_{PC,PH} = L_2(\psi) L_3\left[-\frac{\pi}{2}\right] \quad (2.1-9)$$

The propeller geometry is then described with respect to an arbitrary line defined in this coordinate system. The profiles perpendicular to this line are given in local coordinates with the leading edge, chord, and twist angle defined at specified values of the distance along the PCA. Once the blade is defined, an additional twist,  $\Delta\beta_{3/4}$ , about the pitch-change axis will be made to account for different blade operating conditions. The geometry of the transformation from the propeller-hub frame to the pitch-change axes is shown in Figure 2.1-7.

The transformation to the local frame is defined by finding the tangent vector to the arbitrary line and using the components of the tangent vector to define the in-plane and out-of-plane sweep angles. Finally, the twist angle is applied to give the transformation from the reference frame to the local frame. The way that the final twist angle is applied depends on the direction of rotation, since the profiles are always defined such that the local x-axis is along the profile from the leading to the trailing edge. The transformation from the reference frame to the local frame is

$$L_{L,PC} = L_3(-\beta) L_1(\phi_{out}) L_2(\psi_{in}) \quad (2.1-10)$$

when the propeller rotates in a clockwise direction when observed from the front -- like the SR-3. When the propeller rotates in a counterclockwise direction --- like the Hartzell 101/16 --- the transformation from the reference frame to the local frame is

$$L_{L,PC} = L_3(-\beta) L_1(-\phi_{out}) L_2(-\psi_{in} - \pi) \quad (2.1-11)$$

where the local profile is aligned so that the x-axis of the profile starts at the leading edge and goes to the trailing edge, while the local y-axis points toward the suction side of the profile.

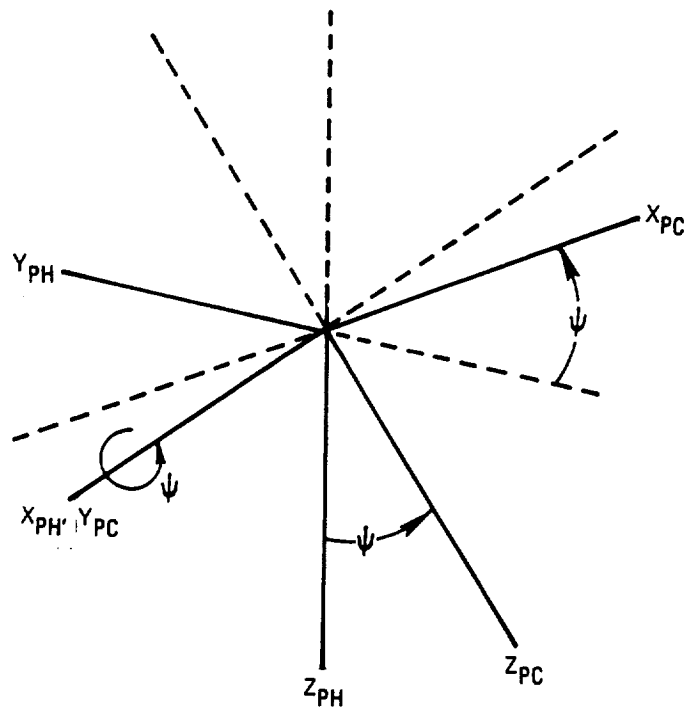


Figure 2.1-7. Transformation from Prop-Hub Frame to Pitch-Change Axes

The transformations are summarized in Figure 2.1-8. The in-plane and out-of-phase sweep angles,  $\psi_{in}$ , and  $\phi_{out}$ , are shown in figures in Section 2.1.5 for clockwise and counterclockwise rotation.

#### 2.1.4 Cubic Spline Interpolation

There are many applications of the subroutines SETSPL and CSPLINE to setup and interpolate with cubic splines. The usual method of application is to find the spline-fit coefficients using the approximate arc length,  $s$ , as the independent variable. Following Figure 2.1-9, the "arc-length" is defined such that

$$s_{i+1} = s_i + \sqrt{dx_i^2 + dy_i^2 + dz_i^2} , \quad (2.1-12)$$

where

$$dx_i = x_{i+1} - x_i \quad (2.1-13a)$$

$$dy_i = y_{i+1} - y_i \quad (2.1-13b)$$

$$dz_i = z_{i+1} - z_i . \quad (2.1-13c)$$

If  $\xi$  is a dependent variable that can be either  $x$ ,  $y$ , or  $z$ , we assume that each component  $\xi$  has a cubic form within each segment,

$$\begin{aligned} \xi(s) = \xi_i + \xi_i' * ds(i) + (1/2!)*\xi_i''(i)*ds(i)^2 + \quad (2.1-14) \\ + (1/3!)*\xi_i''' * ds(i)^3, \end{aligned}$$

where

$$ds(i) = s - s_i, \quad (2.1-15)$$

and where  $s$  is the independent variable. From segment to segment, the values of the function, first derivative, and second derivative are assumed to be continuous. The third derivative is allowed to be discontinuous, thus the definition of the "knot".

Expressing the third derivative  $\xi_i'''$  in terms of the second derivative at the endpoints  $\xi_i''$ ,  $\xi_{i+1}''$  and requiring continuity of the first derivative  $\xi'(s)$  at the endpoints, we obtain a tridiagonal equation for  $\xi_i'''$  in terms of the functional values at the knots.

W: Wind Axes

B: Body Axes

$$\underline{\underline{L}}_{B,W} = \underline{\underline{L}}_2(\alpha) \underline{\underline{L}}_3(-\beta)$$

Angle of Attack      Sideslip Angle

PH: Propeller Hub Axes

$$\underline{\underline{L}}_{PH,B} = \underline{\underline{L}}_2(-\alpha_D) \underline{\underline{L}}_3(\beta_R)$$

Downthrust Angle      Right-Thrust Angle

PC: Pitch Change Axes

$$\underline{\underline{L}}_{PC,PH} = \underline{\underline{L}}_2(\psi) \underline{\underline{L}}_3(-\frac{\pi}{2})$$

Azimuth Angle

R: Reference Axes

$$\underline{\underline{L}}_{R,PH} = \underline{\underline{L}}_3(-\Delta\beta_{3/4}) \quad \text{or} \quad \underline{\underline{L}}_3(\Delta\beta_{3/4})$$

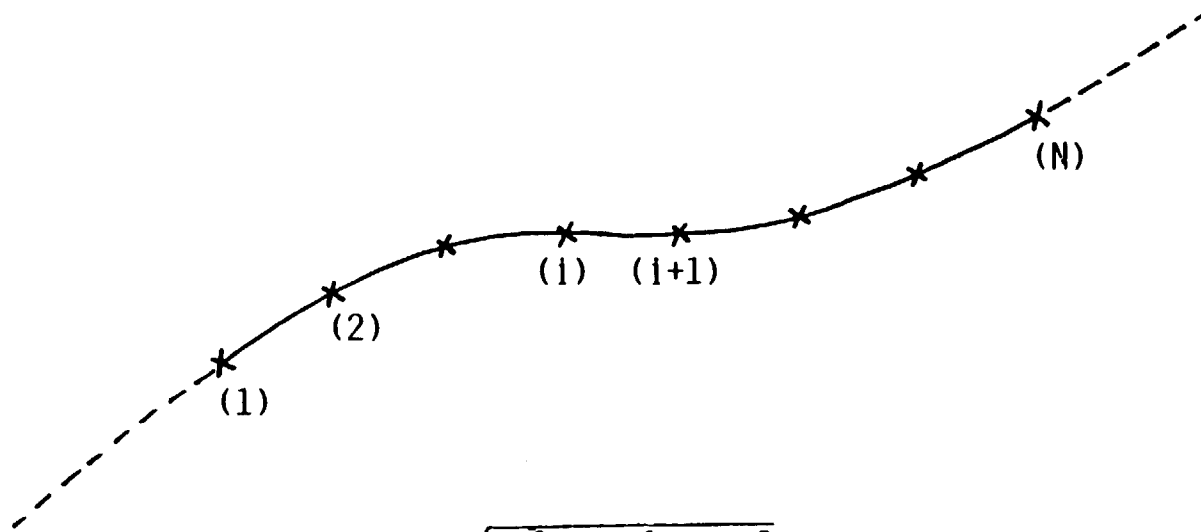
CW      CCW  
Operating Angle at 3/4 Radius

L: Local Axes

$$\underline{\underline{L}}_{L,R} = \underline{\underline{L}}_3(-\beta) \underline{\underline{L}}_1(\phi_{out}) \underline{\underline{L}}_2(\psi_{in}) \quad \text{or} \quad \underline{\underline{L}}_3(-\beta) \underline{\underline{L}}_1(-\phi_{out}) \underline{\underline{L}}_2(-\psi_{in}-\pi)$$

Twist Angle      In-Plane Sweep Angle  
Out of Plane Sweep Angle

Figure 2.1-8. Summary of Coordinate Systems

Applications

"Wires"

Arbitrary Line

Local Profiles

Pressure, Skin Friction,

Shear Center

$$s_i = s_{i-1} + \sqrt{\Delta x_i^2 + \Delta y_i^2 + \Delta z_i^2}$$

$$\{x(s)\}_{pc} = \begin{pmatrix} x(s) \\ y(s) \\ z(s) \end{pmatrix}$$

For each component,  $\xi$ , where  $\xi$  can be  $x$ ,  $y$  or  $z$

$$\xi = \xi_i + \xi_i'(s - s_i) + \frac{1}{2} \xi_i''(s - s_i)^2 + \frac{1}{3!} \xi_i'''(s - s_i)^3$$

Match  $\xi$ ,  $\xi'$ ,  $\xi''$  at "knots" (where  $\xi'''$  is discontinuous)

## Boundary Conditions

- (1) Cantilever ( $\xi' = \text{const}$ )
- (2) Free ( $\xi'' = 0$ )
- (3) "Not-a-knot" (continuous  $\xi'''$  at  $i = 2$  or  $i = N - 1$ )

Figure 2.1-9. Cubic Spline Interpolation

$$\begin{aligned}
 h_{i-1} \xi_{i-1}'' + 2(h_{i-1} + h_i) \xi_i'' + h_i \xi_{i+1}'' &= \quad (2.1-16) \\
 &= 6 \left[ \frac{\xi_{i+1} - \xi_i}{h_i} - \frac{\xi_i - \xi_{i-1}}{h_{i-1}} \right]
 \end{aligned}$$

where  $h_i = s_{i+1} - s_i$ ,  $h_{i-1} = s_i - s_{i-1}$ .

Finally, the boundary conditions at each end of the spline are assumed to satisfy one of follow conditions:

- (1) Cantilever condition ...  $\xi_i'$  or  $\xi_N'$  equal to a given constant.
- (2) Free-end condition ...  $\xi_i''$  or  $\xi_N''$  equal to zero.
- (3) "Not-a-Knot" condition ...  $\xi_1'' = \xi_2''$  or  $\xi_{N-2}'' = \xi_{N-1}''$ .

When extrapolating, we take the slope equal to a constant, and extrapolate from the end point.

The calculation of the spline-fit coefficients is performed in subroutine SETSPL. the interpolation/extrapolation using the spline-fit coefficients is performed in subroutine CSPLINE.

### 2.1.5 Blade Geometry Definition

The three-dimensional blade geometry is developed using a five-step process as outlined below.

- (1) Define an arbitrary line and input profiles perpendicular to this line. Typically this line is the pitch-change axis (PCA) or the line of shear centers (when considering a deflected blade). The spline-fit coefficients for the arbitrary line are set up, since it will be necessary to define planes perpendicular to it.
- (2) At a given radial station along the arbitrary line, the transformation from the pitch-change frame to the local frame is defined in terms of the in-plane sweep angle,  $\psi_{in}$ , the out-of-plane sweep angle,  $\phi_{out}$ , and the local twist angle  $\beta$ . First the coordinates are rotated about the y-axis through the in-plane sweep angle,  $\psi_{in}$ , so that the local tangent vector is in the new y-z plane. Then, the coordinates are rotated about the new x-axis through the out-of-plane sweep angle,  $\phi_{out}$ , so that the tangent vector to the defining arbitrary line is along the local z-axis. Finally, the coordinate system is rotated through the local twist angle,  $\beta$ , so that the new x-axis is parallel to the local chord of the airfoil. The transformation matrix from the PC-frame to the local frame is



$$L_{L,PC} = L_3(-\beta) L_1(\phi_{out}) L_2(\psi_{in}) \quad (2.1-17)$$

for the case of clockwise rotation like the SR-3 while for counterclockwise rotation

$$L_{L,PC} = L_3(-\beta) L_1(-\phi_{out}) L_2(-\psi_{in}-\pi) \quad (2.1-18)$$

The in-plane and out-of-plane sweep angles are schematically represented in Figures 2.1-10 and 2.1-11 for clockwise and counterclockwise rotation.

- (3) The local nondimensional profile is defined with respect to the leading edge of the airfoil,

$$\left\{ \begin{matrix} \hat{x} \\ \hat{y} \\ 0 \end{matrix} \right\} = \left\{ \begin{matrix} \hat{x} \\ \hat{y} \\ 0 \end{matrix} \right\} \quad (2.1-19)$$

where  $\hat{x}$  and  $\hat{y}$  are the coordinates measured from the nose of the airfoil, nondimensionalized with respect to the local chord,  $c$ . The leading edge point  $\{x_{LE}\}_L$  is defined relative to the point where the arbitrary point pierces the local plane,  $\{x_{out}\}_L$  by the leading-edge alignment, LEA, and the horizontal alignment, HA

$$\left\{ x_{LE} - x_{out} \right\}_L = \left\{ \begin{matrix} -LEA \\ HA \\ 0 \end{matrix} \right\} \quad (2.1-20)$$

The defining "arbitrary" line that defines the propeller blade sections is typically the pitch-change axis. The generalization of the arbitrary line is important for the case where the blade deflects and twists under the inertial and aerodynamic loads. In that case, the profiles will be redefined with respect to the line of shear centers and will be assumed to be undeformed as the line of shear centers deflects and twists. The propeller blade section is defined in local coordinates as indicated in Figure 2.1-12.

Although the defining profiles were input with arbitrary spacing, we can redefine the profiles in terms of user-specified "nicely spaced" coordinates. Each profile will have the same nondimensional spacing which will then be connected to form a three-dimensional surface. The profiles are fitted with planes and interpolated to determine these new local coordinates. The leading edge radius is also calculated and stored in the data base.

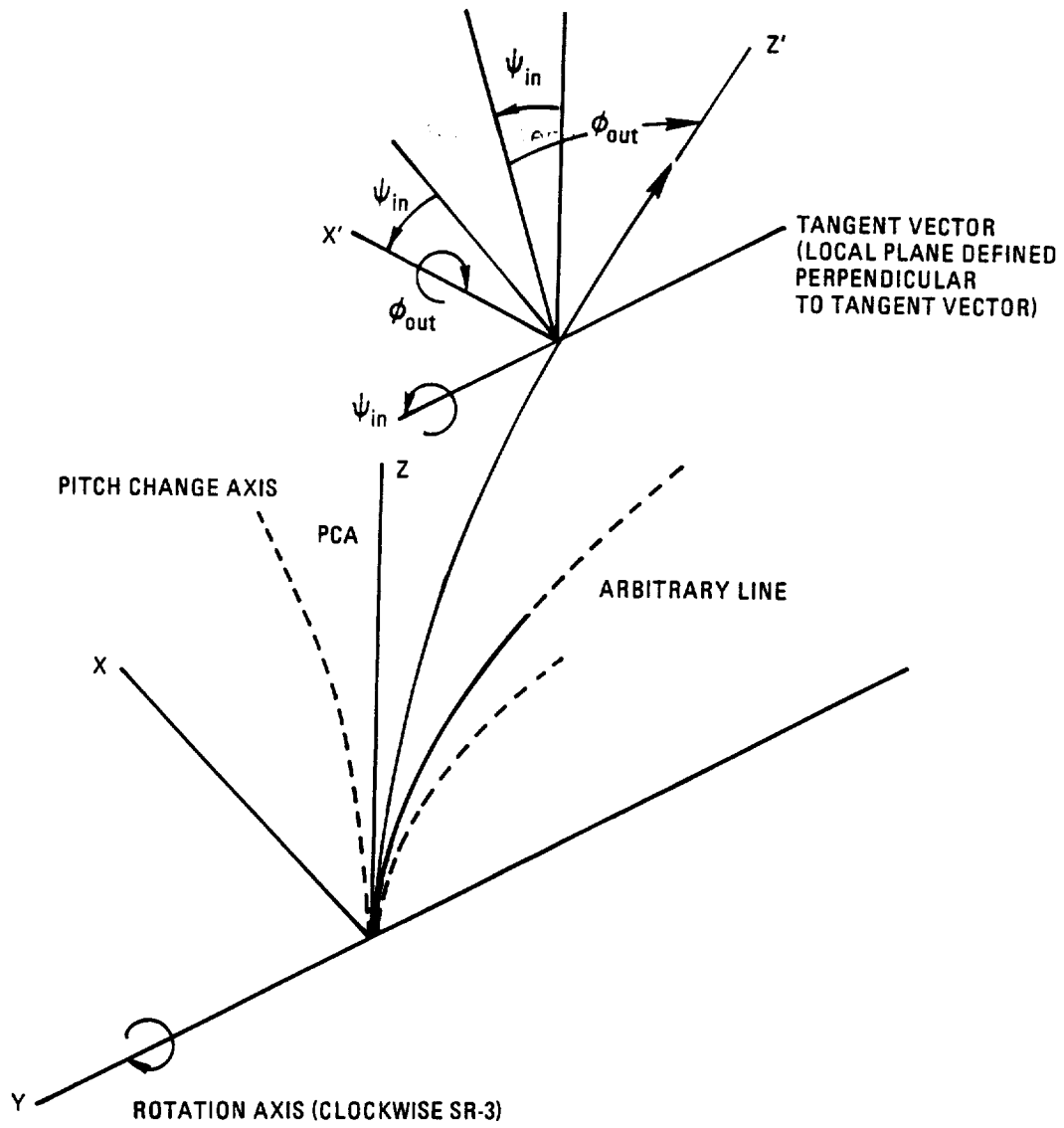


Figure 2.1-10. Sweep Angle Definitions for Clockwise Rotation

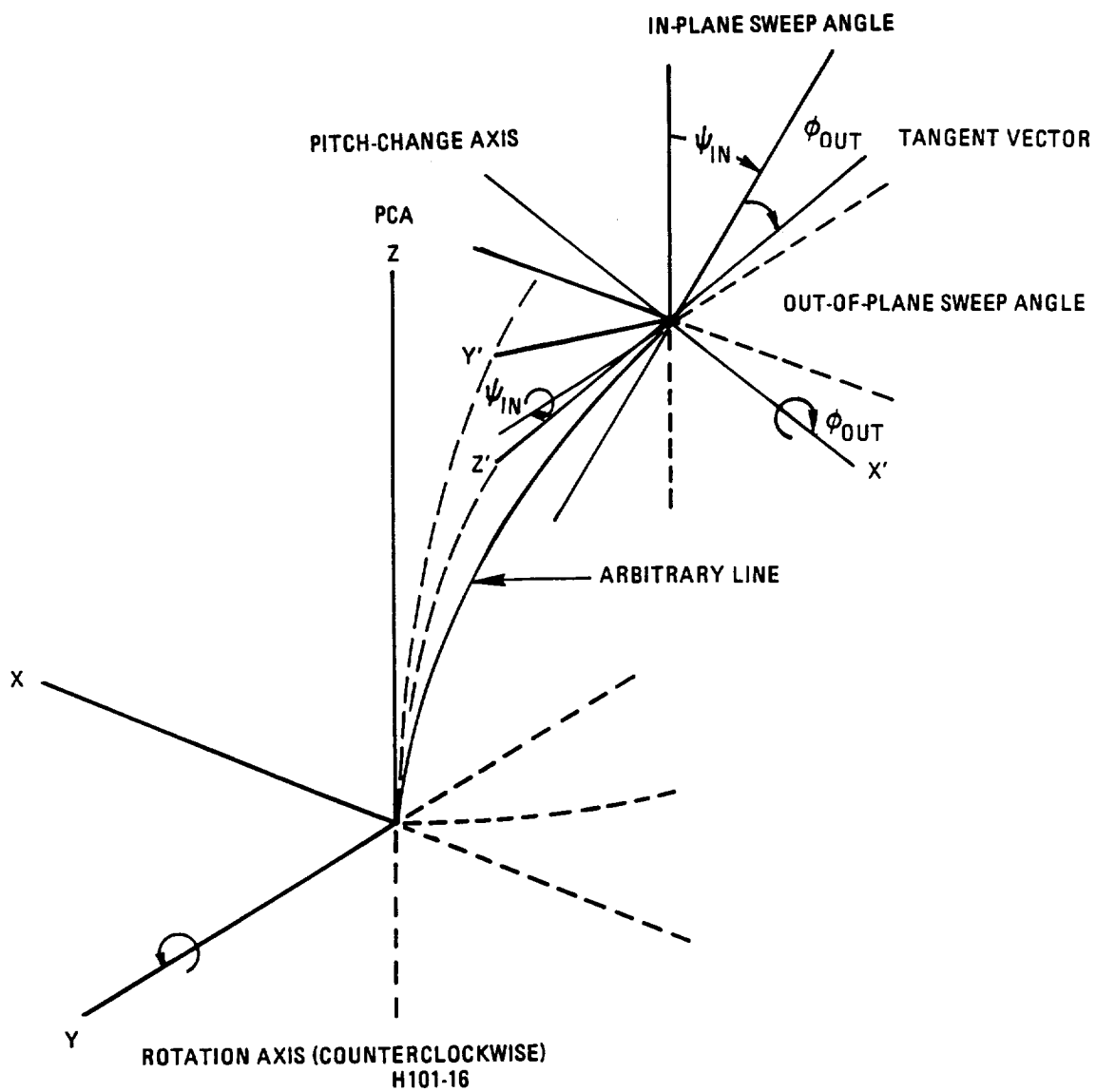


Figure 2.1-11. Sweep Angle Definitions for Counterclockwise Rotation

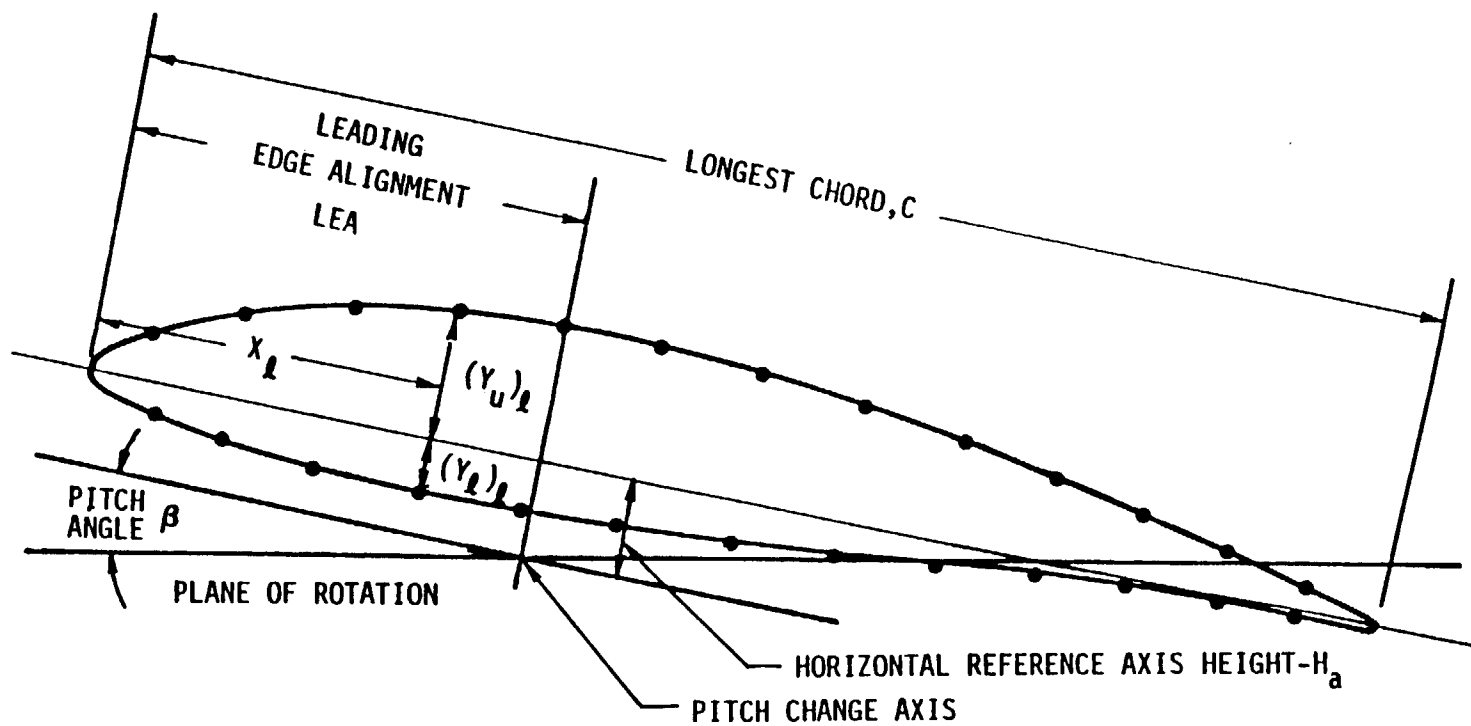


Figure 2.1-12. Airfoil Section Local Coordinate System (LCS)

- (4) The "global coordinates" of the "nicely spaced" profile are defined by

$$\{ \tilde{x} \}_{PC} = \{ \tilde{x}_{arb} \}_{PC} + L_{PC,L} \left[ c \{ \hat{\tilde{x}} \}_L + \{ \tilde{x}_{le} - \tilde{x}_{arb} \}_L \right] \quad (2.1-21)$$

where  $\{ \tilde{x}_{out} \}$  is the reference point on the arbitrary line,  $L_{PC,L}$  is the transformation matrix from the local coordinate frame to the pitch-change frame (equal to the transpose of  $L_{L,PC}$ ),  $c$  is the local chord length,  $\{ \hat{\tilde{x}} \}$  is the nondimensional point on the profile, and  $\{ \tilde{x}_{le} \}$  is the leading edge location. The subscripts PC and L refer to the reference frame of measurement - pitch-change or local.

- (5) Define the "wires" connecting profile to profile. The spline-fit coefficients are set up and stored for future use since it will be necessary to take cuts perpendicular to a different arbitrary line depending on which module is being utilized.

The geometry for the SR-3 propeller following these steps is shown in Figures 2.1-13 and 2.1-14 where the profiles perpendicular to the line of shear centers are also indicated.

#### 2.1.6 Intersection of a Wire with a Plane Perpendicular to an Arbitrary Line

Once the cubic spline is set up for an arbitrary line (subroutine SETARB), it is a straightforward matter to find the tangent vector at any point on the line defined by the distance  $R$  along the pitch-change axis ( $z$ -axis). This is done by finding the root of a function defined by

$$F(s) = z(s) - R_1 = 0. \quad (2.1-22)$$

The value of  $s$  is varied until the function is equal to zero within a specified tolerance. The other two coordinates can be calculated by interpolation using the same value of  $s$  and the appropriate spline-fit coefficients. The derivatives,  $x'(s)$ ,  $y'(s)$ , and  $z'(s)$  determined from this interpolation are used to establish the unit tangent vector at the point on the arbitrary line.

$$\{ \hat{\tilde{x}} \} = \frac{1}{d} \begin{Bmatrix} x' \\ y' \\ z' \end{Bmatrix} \quad (2.1-23)$$

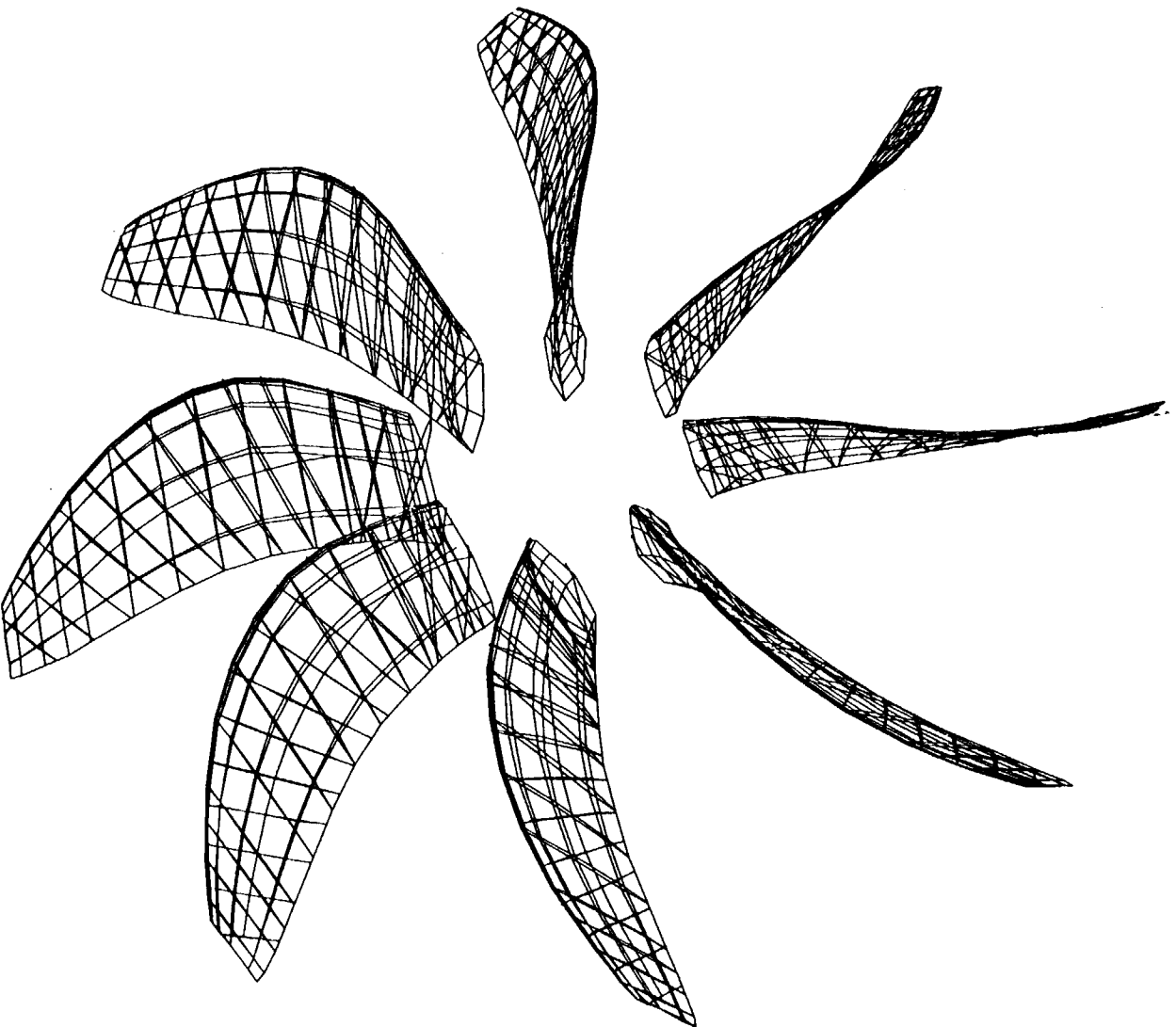


Figure 2.1-13. SR-3 Propeller Geometry

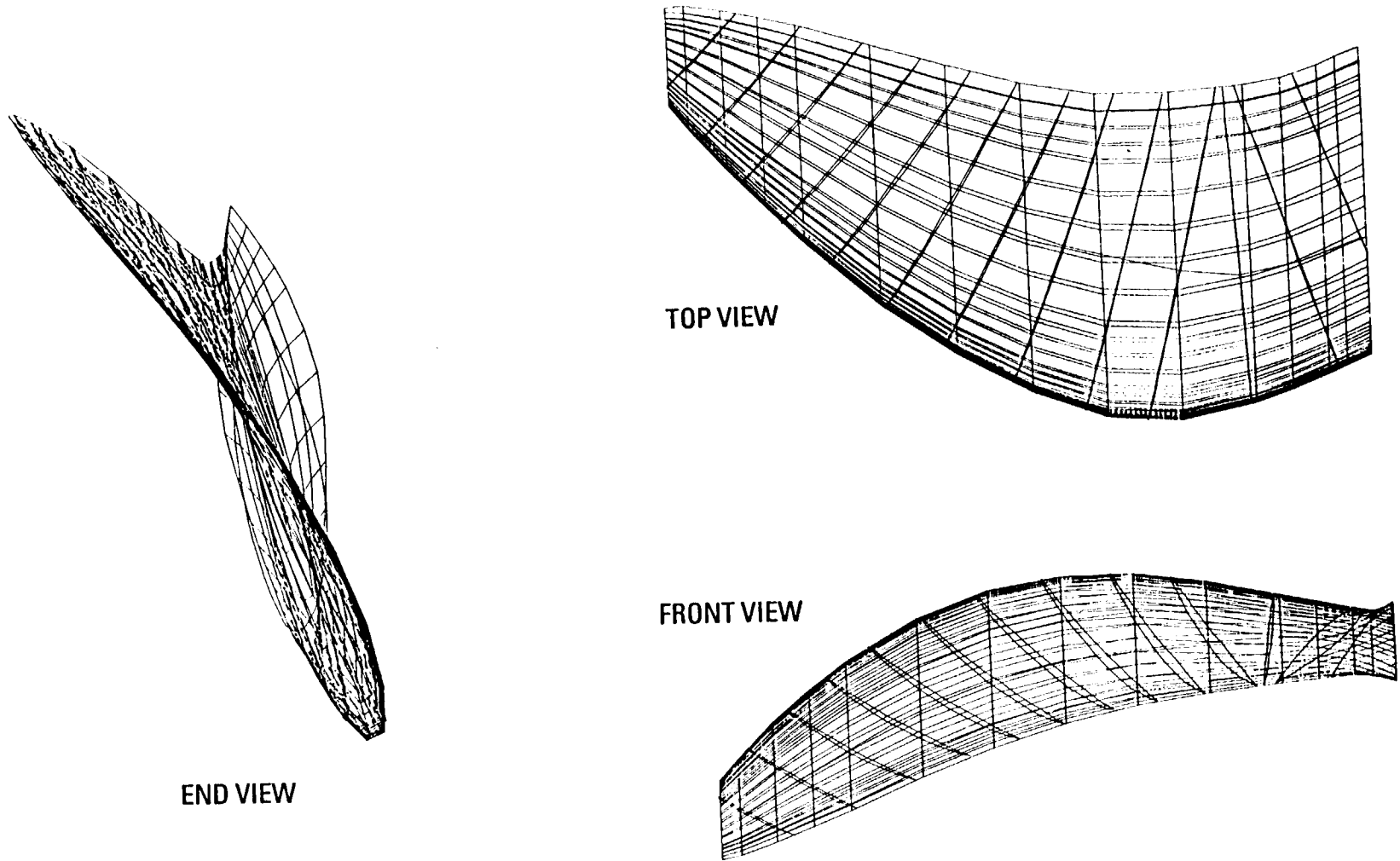


Figure 2.1-14. Three-View of SR-3 Propeller Blade Cuts Perpendicular to Line of Shear Centers

where

$$d = \sqrt{(x')^2 + (y')^2 + (z')^2}.$$

Then, the plane perpendicular to the point  $\{x_{out,(i)}\}$  is determined by another function

$$F_1(s) = \{x(s) - x_{arb}(i)\} \cdot \{\hat{t}\} = 0, \quad (2.1-24)$$

where  $\{x(s)\}$  is the point defined by the wire spline-fit coefficients and  $\{\hat{t}\}$  is the tangent vector. Using the function subroutine ZEROIN, we can find the root of this function by varying the value of  $s$ .

#### 2.1.7 Structural Module Requirements

The structures beam module requires a consistent set of profiles perpendicular to the line of shear centers as part of setting up the beam elements. The methodology for obtaining these profiles is based on an iterative scheme consisting of the following five steps.

- (1) An initial guess is made for the line of shear centers. As a first approximation, it is presumed to be the line of midchords. Based on this line, the spline-fit coefficients are determined.
- (2) The profiles perpendicular to this line at each of the input stations ( $R_j$ ) are then determined.
- (3) The shear center of this profile, using the methods described in the structures section, is then determined.
- (4) Check the new location of the shear center with the previous location and iterate until a specified tolerance in the change in shear center location is made.
- (5) Set up the new spline coefficients for the new line of shear centers and return to Step (2), if the convergence criteria are not met.

#### 2.1.8 Propeller Performance Modules Requirements

In order to calculate the propeller performance and acoustic signature of the propeller, it is necessary to calculate specific input to these modules. These include the locations of the trailing vortex points and control points for input into the Chang-Sullivan propeller performance module and to determine the appropriate section profiles for airfoil



loading and acoustics modules. The methodology incorporated on the geometry module follows a 12 step approach:

- (1) Read the specified output radial stations,  $R_0$ .
- (2) Setup a line through the quarter-chord points based on the initial profiles. The spline-fit coefficients are used to setup the tangent vector,  $\{\hat{t}\}_{PC}$ , at each radial station.
- (3) Profiles are found perpendicular to the quarter-chord line by finding the location where each wire pieces the plane. That is, for each point on a wire  $\{x(s)\}_{PC}$ , we vary  $s$  until

$$\{x(s) - x_{1/4}\}_{PC} \cdot \{\hat{t}\}_{PC} = 0. \quad (2.1-25)$$

This calculation is performed in using function subroutine SLICE.

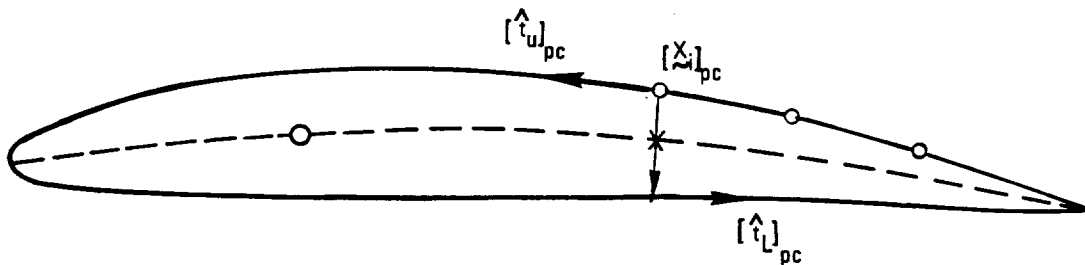
- (4) For each profile we find the leading edge  $\{x_{le}\}$  trailing edge,  $\{x_{te}\}$  chord length  $c$ , twist angle  $\beta$ , and transformation matrix,  $L_{L,PC}$ , from the pitch-change frame to the local frame.
- (5) Set up the spline-fit coefficients defining the profile shape starting at the trailing edge moving around the nose along the upper surface and back to the trailing edge along the lower surface (subroutine SETSPL). A finite trailing edge is allowed in the analysis.
- (6) Determine the camber line for the profile. In particular, the quarter-chord point on the camber line is the location where the trailing vortices are assumed bound to the propeller. The camber line is determined by varying the independent parameter  $s$  (approximate arc length as explained in Section 2.1.4) until

$$F(s) = \{x(s) - x_i\} \cdot \{\hat{t}_u - \hat{t}_L(s)\} = 0 \quad (2.1-26)$$

where, from Figure 2.1-15,  $\{x(s)\}$  is a point on the lower surface of the airfoil,  $\{x_i\}$  is a "knot" on the upper profile,  $\{\hat{t}_u\}$  is the tangent vector on the upper surface and  $\{\hat{t}_L(s)\}$  is the tangent vector on the lower surface. This defines a vector  $\{x(s) - x_i\}$  that is perpendicular to the average of the two tangent vectors. The point halfway between the upper and lower points is on the camber line. The calculation is performed in subroutine camber using function subroutine FCAMB and ZEROIN.

- (7) Define profiles perpendicular to the quarter-chord line midway between the previous profiles. This airfoil section is required by the airfoil loading module to determine the loading and moment.

FIND CAMBER LINE AND 1/4-CHORD POINT ON CAMBER LINE



$$F(S) = [\underline{X}(S) - X_i] \cdot [\underline{t}_u - \underline{t}_L(S)] = 0$$

TRAILING VORTICES BOUND TO THE PROPELLER BLADE AT 1/4-CHORD LOCATION

Figure 2.1-15. Camber Line and Quarter-Chord Point

- (8) Define the camber line on this airfoil section in a manner similar to Steps (5) and (6).
- (9) Find the control point and normal vector at the three-quarter point on the camber line for use in the Chang-Sullivan performance module (Figure 2.1-16). The boundary condition required to determine the strengths of the unknown bound vortices requires that the normal component of the velocity is zero at this control point. The inflow velocity components are required in the propeller plane in order to satisfy the above boundary conditions. The quarter-chord load point is used to define the propeller plane, which is then transferred through the data base to the aircraft flow field module. After the velocity components are calculated by the aircraft flow field module, they are transformed to the global coordinates used by the propeller performance module (Chang-Sullivan)

$$\{ \underline{V} \}_{CS} = L_{CS,PC} \{ \underline{V} \}_{PC} \quad (2.1-27)$$

where the transformation matrix from the pitch-change frame (PC) to the Chang-Sullivan frame (CS) is

$$L_{CS,PC} = L_3(\pi/2) L_1(\pi/2) \quad (2.1-28)$$

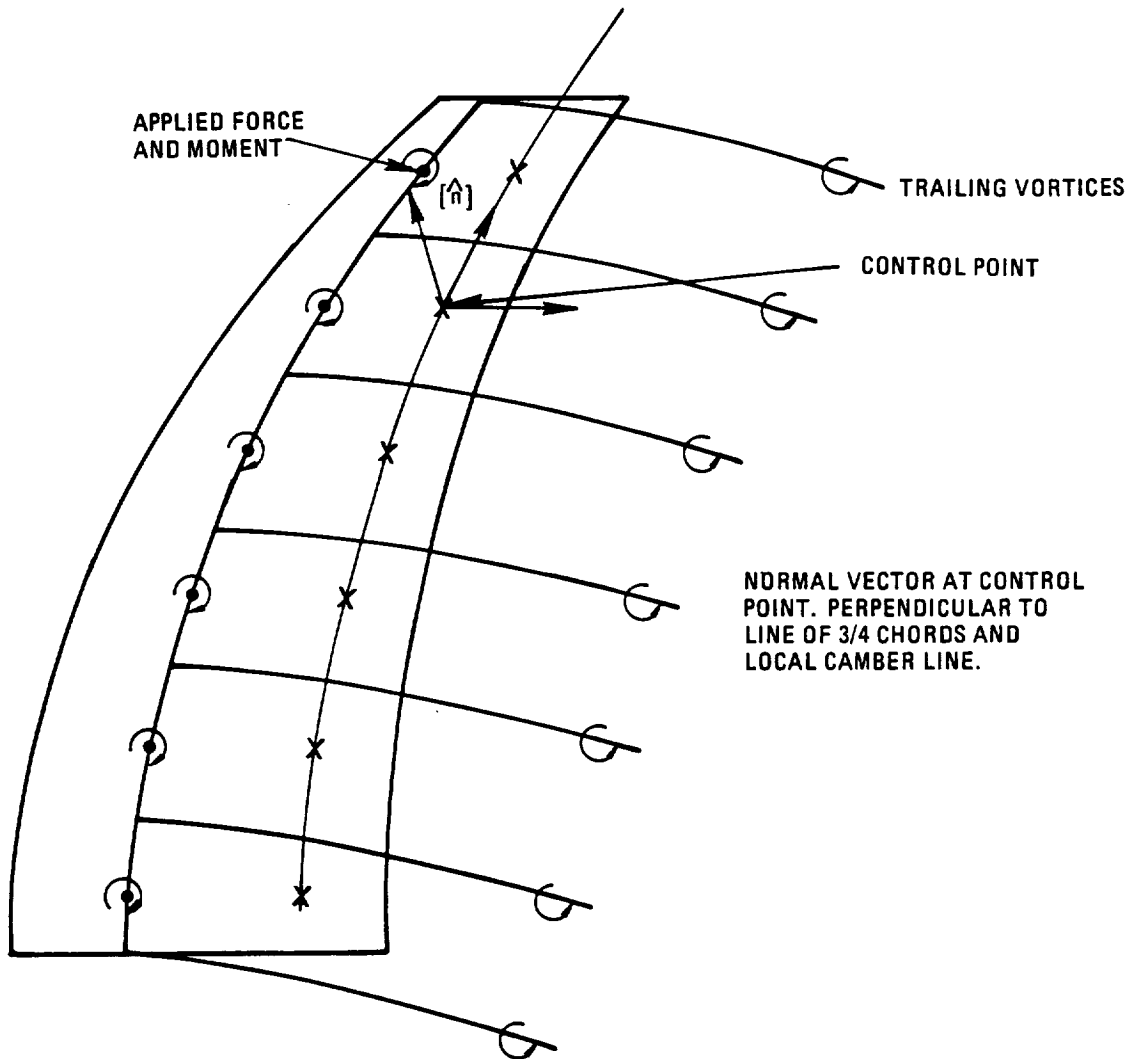


Figure 2.1-16. Find Control Point, Load Point and Trailing Vortex Point

- (10) The transformation matrix  $L_{L,pc}$  is saved in order that components of the local aerodynamic forces and moments can be transformed to global (pitch-change) coordinates for use in the structures module.
- (11) Execution of the acoustics module requires the definition of the profiles perpendicular to the pitch-change axis, in particular the local nose radius,  $R_N$ , and the thickness to chord ratio,  $t/c$ .
- (12) Set up elliptic coordinates for interpolation of the surface pressure distribution for the acoustics module. The elliptic coordinates are described in Section 2.5.3 of this report.

#### 2.1.9 Blade Reflection - Iterative Calculation

The line of shear centers and the profiles perpendicular to it are described in Section 2.1.7 of this manual and Section 2.2.1 of the GAPAS final report. When the line of shear centers is constructed, the profiles normal to this line are stored so that the blade geometry can be reconstructed after the beam elements are deflected and twisted. In the iterative scheme, the loads computed by the propeller performance module are passed through the data base to the structures module. The structures module then calculates the deflection and twist of the beam elements based on the aerodynamic and inertia loads. Now, the new propeller blade is constructed using the new line centers as the defining line with the previously stored profiles constructed perpendicular to this line with an incremental local twist angle added to account for the twist on the beam. New loads can be found from the performance module and the next iteration on the deflected shape of the beam elements (line of shear centers) can be found. This procedure is continued until convergence based on the incremental displacement is attained. Sample calculation with the undeflected shape and the first three iterations are shown in Figure 2.1-17. These results indicate that there are some problems with the convergence of this preliminary version of the iterative scheme.

#### 2.1.10 G400 Interface

the G400 interface is not completely implemented into GAPAS. However, some of the required geometrical and structural properties are calculated and stored in the data base. A major effort will be required to allow GAPAS to interface efficiently with the stand-alone version of the G400

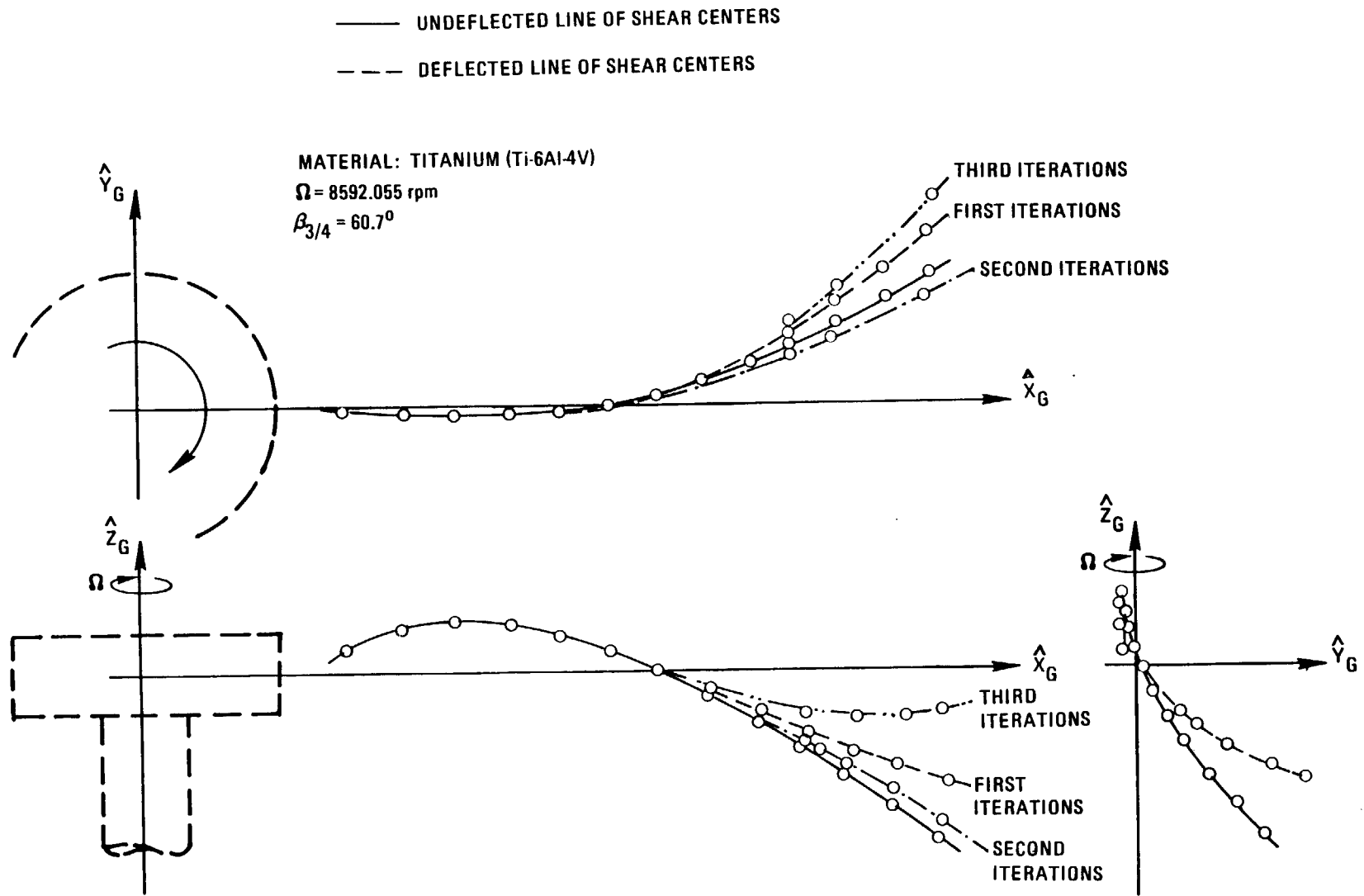


Figure 2.1-17. Undeformed and Deformed Line of Shear Centers

code. This will require a clarification of the input variable definitions, including the input parameters for the various options and flag settings. Radius, number of segments, mass, inertia, torsion stiffness, etc. have been written into arrays, but other quantities such as flatwise modal frequencies, explicit edgewise spring rates, etc., have not been computed.

#### 2.1.11 Geometry Generator Subroutine Summary

The geometry generator module contains the subroutines described below:

- GEOGEN - The main driver routine.
- INPUT - This subroutine reads the input data for defining profiles and calls routines to redefine coordinates and set up spline-fit coefficients for the wires. Called by GEOGEN.
- NICE - Subroutine for generating "nicely spaced" wire coordinates. Find leading and trailing edges of the profiles. Called by INPUT.
- FCHORD - Function to minimize finding the longest chord. Called by NICE.
- FCOORD - Function used by ZEROIN to find the "nicely spaced" coordinates. Called by NICE
- SETARB - Subroutine to set up splines for arbitrary line. Called by INPUT, GGENDR.
- SPWIRE - Subroutine to spline fit the wire data in pitch change frame. This routine finds the sweep angles and the transformation matrix from the local frame to the pitch change frame,  $L_{pC,L}$ . The profiles are input perpendicular to arbitrary line #1. Called by INPUT.
- LPERP - Subroutine to calculate the transformation matrix for the plane perpendicular to the given arbitrary line at the input radius. Called by SPWIRE, PROFIL.
- FRAD - Function to calculate the difference between input radius and calculated radius versus interpolation distance,  $s$ . Used by ZEROIN, called by LPERP.

- SETWIR - Subroutine to set up splines of the wires. Called by SPWIRE.
- PROFIL - Subroutine to calculate propeller section properties for a slice perpendicular to an arbitrary line. Called by GGENDR.
- SLICE - Subroutine to find the intersection of a wire with the plane perpendicular to an arbitrary line. Called by PROFIL.
- FSLICE - Function to calculate the difference between input radius and calculated radius versus interpolation variable s. Called by SLICE.
- CAMBER - Subroutine to calculate the camber line in an arbitrary slice. Called by PROFIL.
- FCAMB - Function to find the location of the camber line. Called by CAMBER.
- CLINE - Function subroutine to find the camber-line location at the nondimensional x-coordinate. Called by GGENDR using ZEROIN.
- PSLICE - Subroutine to output the slice information. Called by GGENDR.
- ROTATE - Subroutine to rotate the input blade coordinates through the angle DBET34. Higher values of DBET34 increase the pitch setting. Called by INPUT.
- SETSPL - Subroutine to set up coefficients for cubic spline interpolation. Parameters allow for cantilever, free or "not-a-knot" end boundary conditions. Called by NICE, SETARB, SETWIR, PROFIL, CAMBER, GGENDR, G400.
- CSPLINE - Subroutine to interpolate (or extrapolate) cubic spline function. Called by FCHORD, NICE, FCOORD, FRAD, LPERP, FSLICE, SLICE, FCAMB, CLINE, GGENDR, FROUT, G400.
- INTEG - Subroutine to integrate a spline function. Called by G400.
- SL - Subroutine to calculate the rotation matrix  $L_n(x)$ , representing the angular rotation x about the n-axis. Called by LPERP, SPWIRE, PROFIL.

- MVMULT - Subroutine to multiply matrix [A] by vector {x}:  $\{y\} = [A]\{x\}$ .  
Called by SPWIRE, PROFIL, FSLICE, GGENDO, G400.
- MMMULT - Subroutine to multiply matrix [A] by matrix [B]:  $[C] = [A][B]$ .  
Called by LPERP, SPWIRE, PROFIL.
- TRANSP - Subroutine to calculate the transpose of matrix [A]. Called by  
SPWIRE, GGENDR.
- CROSS - Subroutine to calculate the cross product  $\{c\} = \{a\} \times \{b\}$ . Called  
by GGENDR, VOLUME.
- DOT - Function subroutine to calculate the dot product  $DOT = \{a\} \cdot \{b\}$ .  
Called by VOLUME.
- XNORM - Function subroutine to calculate the norm of a vector. Called by  
GGENDR.
- FMIN - Function subroutine to find an approximation to a point where F  
(from external subroutine) attains a minimum on the interval (AX,  
BX). Called by NICE (FCHORD).
- ZEROIN - Function subroutine to compute a zero of the function F(x)  
(external function) in the interval (AX, BX). Called by NICE  
(FCOORD), LPERP (FRAD), SLICE (FSLICE), CAMBER (FCAMB), GGENDR  
(CLINE, FCIRC), G400 (FROUT).
- G400 - Subroutine to calculate the input parameters for the G400 PROP  
aeroelasticity module. Called by GGENDR.
- SEGVOL - Subroutine to calculate the segment volume. Called by G400.
- VOLUME - Subroutine to calculate the volume of a tetrahedron. Called by  
SEGVOL.
- FROUT - Function subroutine to find the value of s at nondimensional  
radius ROUT. Called by G400.
- FCIRC - Function subroutine to determine the location of the 3/4-chord  
control point on a circular arc for the Chang-Sullivan  
performance module. Called by GGENDR using ZEROIN.
- SHRCTR - Driver subroutine for the shear center iteration. Called by  
GGENDR.



- CAPSHR - Subroutine to calculate the section properties and shear center location of an arbitrary airfoil section (solid or multicelled) composed of one isotopic material. This is a finite element code that models the cross section using quadrilateral plate finite elements. Called by SHRCTR.
- CENTER - Subroutine to calculate the center of a triangular finite element. Called by GAPSHR, QUAD.
- AREA - Subroutine to calculate the area of a triangular finite element. Called by GAPSHR, QUAD.
- INERT - Subroutine to determine the moments and products of inertia of a triangular element in a quadrilateral element. Called by GAPSHR.
- WRPCON - Subroutine to determine the third and fourth area moment integrals of a triangular element in a quadrilateral element N (for constants in G400 PROP aeroelasticity module). Called by GAPSHR.
- TEMAT - Subroutine to determine the T matrix for a triangular element in quadrilateral element N. Called by GAPSHR.
- QUAD - Subroutine to determine the stiffness and load matrices for the quadrilateral element N. Called by GAPSHR, QUAD.
- SYMSOL - Subroutine to solve a set of linear algebraic simultaneous equations. Called by GAPSHR.

## 2.2 AIRCRAFT FLOW FIELD

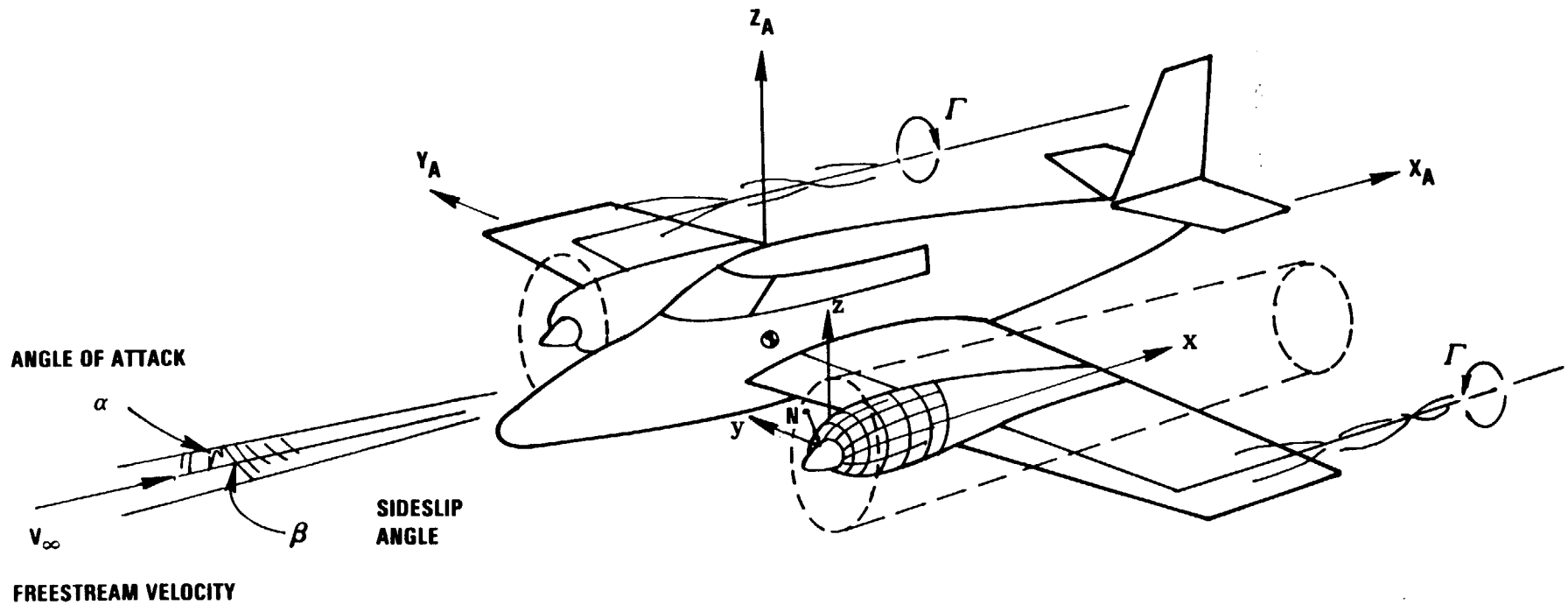
### 2.2.1 Problem Definition

The aircraft flow field module is used to calculate the axial, radial and tangential components of the velocity at arbitrary locations in the plane of the propeller using incompressible potential flow methods. The computational approach accounts for airplane angle of attack and sideslip, nacelle shape, and upwash due to the lifting wing, but neglects the velocity induced by the propeller. These components of the inflow velocity are required by other modules within GAPAS for the calculation of performance, airfoil loading, and acoustic signature of the propeller. The aircraft flowfield is shown schematically in Figure 2.2-1.

### 2.2.2 Aircraft Flow Field Methodology

Jumper's code (Refs. 2.2-1 and 2.2-2) is based on the panel methods of Hess and Smith (Ref. 2.2-3) and is written specifically to calculate the flow field in the propeller plane of a typical general aviation airplane. This code models the fuselage (and nacelles, if applicable) by source panels, the strengths of which are determined by satisfying the condition that the velocity normal to the surface vanish at the center of each panel. The boundary conditions are relaxed at panels simulating air inlets and outlets. The wing is replaced by a single horseshoe vortex. In satisfying boundary conditions at each panel, the velocities induced by the horseshoe vortex are included. The result is a set of simultaneous algebraic equations, equal in number to the number of panels, for the unknown source strengths. The equations are then solved using the Gauss-Seidel iteration method.

Knowing the panel source strengths, the program then calculates the velocity and pressure field over the body surface. Finally, velocity components are calculated at specified points in the propeller plane. The definition of the propeller plane was modified so that swept blades, such as the SR-3, could be considered. The velocity components were calculated at the quarter-chord location along the blade for specified angular position of the pitch-change axis.



**OBJECTIVE: CALCULATE THE AXIAL, TANGENTIAL AND RADIAL VELOCITY COMPONENTS AT A SPECIFIED POINT IN THE PROPELLER PLANE**

**REQUIRED:**

- PROPELLER PLANE DEFINITION
- VELOCITY CONTRIBUTIONS

- (1) FREESTREAM VELOCITY (OPERATING CONDITIONS)
- (2) FUSELAGE INDUCED VELOCITIES (SOURCES/SINKS)  
(NACELLE AND FUSELAGE GEOMETRY)
- (3) WING INDUCED VELOCITIES (VORTICES)  
(WING GEOMETRY)

Figure 2.2-1. Aircraft Flow Field Effects

The program logic of Jumper's code generates either triangular or quadrilateral panels depending upon the specification of points around a section contour. This is illustrated in Figure 2.2-2 taken from Reference 2.2-2. In this figure, a value of the logic parameter NFLAG of 1 signals either that the section is repeated or is the end of the body. The panels are generated and indexed locating the unit normal vector directed outward from the center of each panel. For a triangular panel, this procedure is straightforward since the points are always coplaner. For a quadrilateral panel, the unit vector is obtained from the cross product of the diagonals. Then a plane panel is generated by projecting the original points onto a plane normal to the unit vector which contains the average of the coordinates of the original points. Two or more successive points on a cross-section having the same coordinates results in the generation of one or more triangular panels.

Except for the mechanics of paneling the body, locating control points, and determining unit vectors normal to the panels, Jumper's program follows the approximate representation for distributed sources proposed by Hess and Smith (Ref. 2.2-3). Referring to Figure 2.2-3, the self-induced velocity at the centroid of a panel due to a continuous source strength in unit area,  $q(I)$ , distributed over the panel can be written immediately as,

$$\{ \underline{v}(I) \} = \frac{1}{2} q(I) \{ \hat{n}(I) \} \quad (2.2-1)$$

where  $\{\hat{n}(I)\}$  is the unit normal vector at the center of the  $I$ th panel. The velocity induced by the  $J$ th panel is approximated by assuming that the distributed source is concentrated at the centroid, of the panel, rather than integrating the distribution over the panel. Hess and Smith (Ref. 2.2-3) point out that this is acceptable if the distance between centroids is larger than four times the maximum panel diagonal. Otherwise, the quadrupole terms in the multipole expansion should be included, or the exact integrates should be calculated. Then, the velocities induced by the source panels are

$$\{ \underline{v}_S(I) \} = \frac{1}{2} q(I) \{ \hat{n}(I) \} + \sum_{J \neq I} \frac{S(J) q(J)}{4\pi r^2(I,J)} \{ \hat{e}(I,J) \} \quad (2.2-2)$$

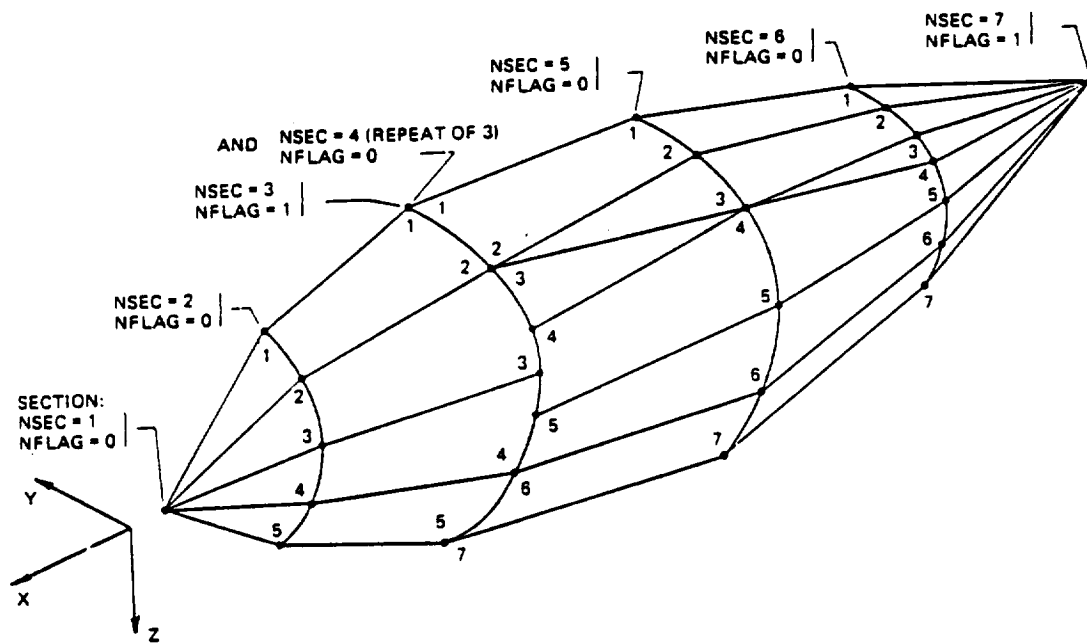


Figure 2.2-2. Sample Body Cross Section Input Showing a Repeated Cross Section Description and Indicating the Input Sequence of Periphery Points

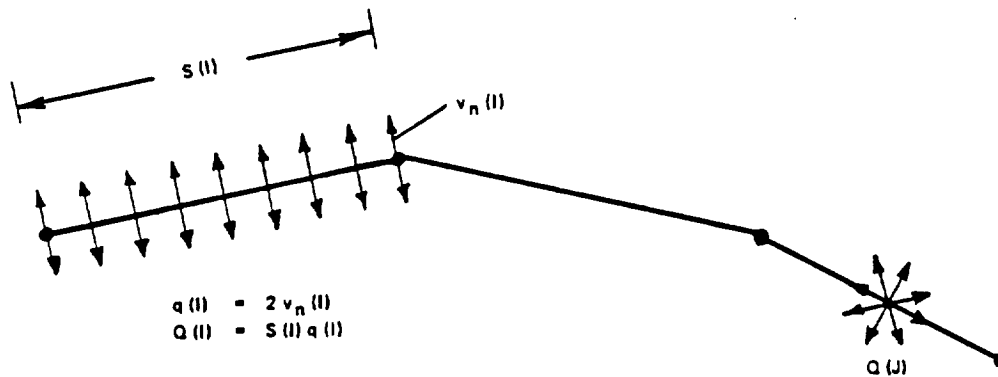


Figure 2.2-3. Distributed Sources at I Panel and Point Source at J Panel

where  $r(I,J)$  is the distance from the centroid of the  $J$ th panel to the centroid of the  $I$ th panel and  $\{\hat{e}(J)\}$  is the corresponding unit vector along  $r(J)$ .

The velocity induced by a vortex element of strength  $\{\Gamma\}$  is given by the Biot-Savart Law

$$\{ \underline{v}_V(p) \} = \int_{s_1}^{s_2} \frac{\{ \underline{x} - \underline{x}_p \} \times \{ \underline{\Gamma} \}}{4\pi | \underline{x} - \underline{x}_p |^3} ds \quad (2.2-3)$$

The induced velocity for a straight line vortex segment is calculated in subroutine VORTEX. Jumper's code replaces the wing with a single horseshoe vortex located at the quarter-chord of the wing. The spacing of the trailing vortices is  $\pi/4$  times the physical span of the wing. The strength of the vortex is determined from the operating conditions by noting that the lift,  $L$ , is

$$L = \rho V_\infty \Gamma \frac{\pi}{4} b = \frac{1}{2} \rho V_\infty^2 C_L S \quad (2.2-4)$$

where  $\rho$  is the air density,  $b$  is the span,  $V_\infty$  is the freestream velocity,  $C_L$  is the wing lift coefficient, and  $S$  is the reference wing area. The velocity components induced by the vortex are calculated in subroutine WINGV.

Assuming that the total velocity at any point is the sum of the freestream velocity, the velocity induced by the source panels, and the velocity induced by the horseshoe vortex,

$$\{ \underline{v} (I) \} = \{ \underline{v}_\infty + \underline{v}_S(I) + \underline{v}_V(I) \} \quad (2.2-5)$$

then the linear set of equations to be solved for the source strength  $q(I)$  is found by requiring that the normal component of velocity is equal to zero

$$\{ \hat{n}(I) \} \cdot \{ \underline{v}(I) \} = 0. \quad (2.2-6)$$

at each of the  $N$  control points.

Then,

$$\frac{1}{2} q(I) + \sum_{\substack{J \neq I \\ I=1}}^N \frac{S(J)}{4\pi r^2(I,J)} \left\{ \hat{n}(I) \right\} \cdot \left\{ \hat{e}(I,J) \right\} q(J) = \quad (2.2-7)$$

$$= - \left\{ \hat{n}(I) \right\} \cdot \left\{ \underline{v}_\infty + \underline{v}_V(I) \right\}$$

### 2.2.3 Numerical Technique

The system of N simultaneous equations for the unknown source strengths  $q(I)$  is solved using the Gauss-Seidel iterative technique. There were no problems with the convergence as long as the size of panels varies smoothly. Hess and Smith (Ref. 2.2-3) point out that the Gauss-Seidel method, which uses the improved values in the iterative scheme as soon as they are computed, always converges faster than the point Jacobi method, which obtains the new approximation entirely from the values of the previous iteration.

### 2.2.4 Velocity Components in the Propeller Plane

Jumper's code calculates the velocity  $\{\underline{v}\}$  at a point  $\{x_p\}$  in the propeller plane in the body reference frame

$$\left\{ \underline{v} \right\}_B = \begin{Bmatrix} v_x \\ v_y \\ v_z \end{Bmatrix}$$

where the x-axis is fixed with the body pointed forward out the nose, the y-axis points out the right wing and the z-axis points down. This calculation is made after convergence of the Gauss-Seidel iteration scheme to solve the linear set of equations. The velocity components required are in cylindrical coordinates in the plane of the propeller. To transform from the body axes to the propeller blade frame, first rotate about the z-axis through the right-thrust angle  $\beta_R$ , then rotate about the new y-axis through the downthrust angle  $-a_D$ , and finally rotate about the x-axis (rotational axis of the propeller) through the angle  $-\psi'$ . This is represented by

$$\left\{ \underline{v} \right\}_{PB} = L_1(-\psi') L_2(-a_D) L_3(\beta_R) \left\{ \underline{v} \right\}_B$$

$$\begin{pmatrix} -V'_z \\ -V'_\theta \\ -V'_R \end{pmatrix} = \begin{bmatrix} \cos\alpha_D \cos\beta_R & | & -\sin\beta_R & | \\ \cos\alpha_D \sin\beta_R \cos\psi' - \sin\psi' \sin\alpha_D & | & \cos\psi' \cos\beta_R & | \\ \sin\psi' \cos\alpha_D \sin\beta_R + \cos\psi' \sin\alpha_D & | & \sin\psi' \cos\beta_R & | \\ -\sin\alpha_D \cos\beta_R & & & \\ -\cos\psi' \sin\alpha_D \sin\beta_R - \sin\psi' \cos\alpha_D & & & \\ -\sin\psi' \sin\alpha_D \sin\beta_R + \cos\psi' \cos\alpha_D & & & \end{bmatrix} \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix} \quad (2.2-8)$$

The axial, tangential and radial velocity components ( $V'_z$ ,  $V'_\theta$ ,  $V'_R$ ) are calculated in subroutine VPROPS. To these velocity components we must add (or subtract) the rotational velocity components  $V_{\theta R} = \Omega R$  where  $\Omega$  is the rotational velocity in radius per second and  $R$  is the radial distance from the rotational axis to the point of interest in the propeller place.

### 2.2.5 Options and Limitations

When the body/wing combination has a plane of symmetry, the propeller has no side thrust angle and there is no sideslip, the computational efficiency can be improved by exercising the symmetric input option. Then, only points on the left half of each cross section need to be input. The right side panels are mirror images of those on the left and the number of equations to be solved is reduced by a factor of two.

The boundary conditions at panels representing inlets or outlets are relaxed such that the normal velocity is specified to be some fraction of the freestream velocity. In this way, cooling flow to the engine can be modeled. This aspect of the code was not exercised for this study although it is built into the code. Guidelines developed by Jumper (Refs. 2.2-1 and 2.2-2) should be used if this option is needed.

The code is written for incompressible potential flow. If the freestream Mach number is large enough, the axial direction should be stretched according to the Prandtl-Glauert rule

$$X = \frac{x}{\sqrt{1 - M_\infty^2}}$$

where  $M_\infty$  is the freestream Mach number and  $X$  is the transformed axial coordinate. However, it would probably only make sense to make this



transformation for bodies at zero angle of attack. This transformation is not incorporated into the code and would have to be made by hand.

For convergence of the Gauss-Seidel integration scheme it is suggested that the panel size change smoothly. That is, don't have large panels next to small panels. Also, the panels near the plane of the propeller have the most influence on the velocities there. Therefore, the paneling in this region should be considered carefully.

### 2.2.6 Input Parameters

The input parameters required by the aircraft flow field module are described in Section 3.4 of this manual. Some of the input parameters have been modified compared to the original revision of the JUMPER program, while a few have been added. Some of the variable names have also been changed to allow compatibility with the other modules. For instance, the sideship angle is now called YAW in the input instead of BETA, which might be confused with propeller twist angle. The propeller plane definition has been changed from the original, so now the number of azimuthal stations NAZIU is now an input parameter instead of DPSI, angular spacing.

Two parameters that have been added are RPM, the rotational speed in revolutions per minute and XREF, a scaling parameter. The rotational speed is used with the inflow velocity to give a first estimate of the total velocity at a point in the propeller plane. Then, using the transformation matrices from the pitch change frame to the local frame,  $L_{L,PC}$ , we can find the local angle of attack  $\alpha$ , from the local velocity components

$$\left\{ \begin{matrix} \underline{v} \end{matrix} \right\}_L = \left\{ \begin{matrix} v_x \\ v_y \\ v_z \end{matrix} \right\} = L_{L,PC} \left\{ \begin{matrix} v_\infty \\ v_{rot} \end{matrix} \right\}$$

and

$$\alpha = \tan^{-1} \left( \frac{v_y}{v_x} \right)$$

The scaling parameter XREF is useful when the coordinates of the body elements need to be scaled up or down without having to reenter the

coordinate data. The current input coordinates are all multiplied by XREF to obtain the required coordinates. Therefore, for example, Coordinates of the model can be used for full scale calculations using the XREF Scale factor.

In the present version of GAPAS the number of inlet and outlet panels are defined to be zero (NINFLO = 0). For non-zero NINFLO, the panel index INDEX, and flow rate, FLRATO, need to be specified. These are not implemented in the current revision of GAPAS. Therefore, do not attempt to run the program with non-zero values for NINFLO.

### 2.2.7 Subroutine Descriptions and Calling/Called Routines

The aircraft flow field module contains the subroutines described below. For a detailed description of the program implemented see Jumper (Refs. 2.2-1 and 2.2-2).

- JUMPER (PROP) - Main program to set up influence coefficients and solve for velocity components in the plane of the propeller. Calls to subroutines NPUT, VCOMP, WGEOM, PANEL, COEFIC, SOLVE, COFSYM, SOLSYM, VELOCI, VPROPS.
- INPUT - Subroutine to read and verify the input data. Called by JUMPER.
- VCOMP - Subroutine to compute the components of the freestream velocity. Called by JUMPER.
- WGEOM - Subroutine to establish the geometry and strength of the wing horseshoe vortex. Called by JUMPER.
- PANEL - Subroutine to generate the surface panels. For a symmetric body, the left half of the body is input and the right half is generated by mirror image, otherwise the entire body is input. Called by JUMPER.
- COEFIC - Subroutine to set up the system of linear equations to be solved used for a fully three-dimensional body. Called by JUMPER. Call to subroutine WINGV.
- SOLVE - Subroutine using Gauss-Seidel iteration scheme to solve the system of equations. Called by JUMPER.

- COFSYM - Subroutine to set up the linear equations for the left half of the body - used for bodies with a plane of symmetry. Called by JUMPER. Call to subroutine WINGV.
- SOLSYM - Subroutine to solve the set of equations when the body is symmetric using Gauss-Seidel iteration. Called by JUMPER.
- VELOCI - Subroutine to compute the body panel surface velocities and pressure coefficient. Called by JUMPER. Call to subroutine WINGV.
- VPROPS - Subroutine to compute the velocity components in the plane of the propeller. Called by JUMPER. Call to subroutine WINGV.
- WINGV - Subroutine to calculate the components of velocity induced at a point by a horseshoe vortex. Called by COEFIC, COFSYM, VPROPS.
- VORTEX - Subroutine to calculate the velocity components induced by a straight line vortex element using the Biot-Savart Law. Called by WINGV.

## 2.3 AERODYNAMIC PERFORMANCE MODULE

The primary aerodynamic performance analysis procedure and computer code selected for inclusion in GAPAS is that developed by Chang and Sullivan (Refs. 2.3-1 through 2.3-3) (the resulting computer code is called the Chang-Sullivan code).

### 2.3.1 Problem Definition

The problem addressed by the Chang-Sullivan methodology is that of calculation of the performance of a propeller of arbitrary blade shape. In addition, methodology is developed to study the performance of propeller proplets that are designed to improve propeller efficiency.

### 2.3.2 Method of Solution

The solution methodology utilizes the vortex-lattice method which approximates a lifting surface and its wake by a discrete vortex lattice system (Ref. 2.3-2). In the Chang-Sullivan methodology, the vortex-lattice method is extended to accommodate a general blade shape and to include viscous and compressible drag in the thrust, power, and radial force formulation.

The propeller is considered to be a thin blade of arbitrary planform, rotating with a constant angular velocity about a common axis in an unbounded fluid. The presence of solid boundaries, including the hub, is ignored. Since the propeller blade is assumed sufficiently thin, its presence in the fluid can be represented by a distribution of horseshoe vortices lying in the chord plane of each blade. There are  $M$  vortex lattices assumed along the blade radius, each having a constant circulation  $\Gamma_i$ . Steady flow conditions are assumed, so that the circulation is only a function of location along the radius.

The vortex wake of the propeller is assumed to be a constant pitch and diameter helix, since the induced velocity is small compared with the resultant velocity of the propeller, and there is no force existing in the radial direction on a helical surface. This assumption, which is coincident with Goldstein's helical vortex model, dispenses with the essential difficulty in lifting-line theory that the induced velocity and the geometry of the trailing helical vortex sheet are mutually dependent.

### 2.3.3 Blade Geometry

As shown in Figure 2.3-1, a Cartesian coordinate system (x,y,z) originates from the center of the hub. The x axis is defined as passing through the propeller blade, the z axis pointing in the positive downwind direction, and the y axis completes the right-hand system. The propeller is modeled with a single lifting line at the quarter chord of each blade and a set of control points along the three-quarter chord line. The propeller is assumed to have K blades. An index p is assigned to each blade, where p = 1, 2----K. The coordinates of the control points can be calculated by

$$YC_j = (YN_j + YN_{j+1})/2 + \left(\frac{C_j}{2}\right) \cos \beta_j$$

$$ZC_j = (ZN_j + ZN_{j+1})/2 + \left(\frac{C_j}{2}\right) \cos \beta_j$$

$$XC_j = \sqrt{r_j^2 - YC_j^2}$$

where

(XC<sub>j</sub>, YC<sub>j</sub>, ZC<sub>j</sub>) = control points of the vortex lattice element j

(XN<sub>i</sub>, YN<sub>i</sub>, ZN<sub>i</sub>) = nodal point i of the vortex element on the 1/4-chord line

$\beta_j$  = the angle of pitch

C<sub>j</sub> = mean chord of the vortex element

r<sub>j</sub> = radial distance of the control points.

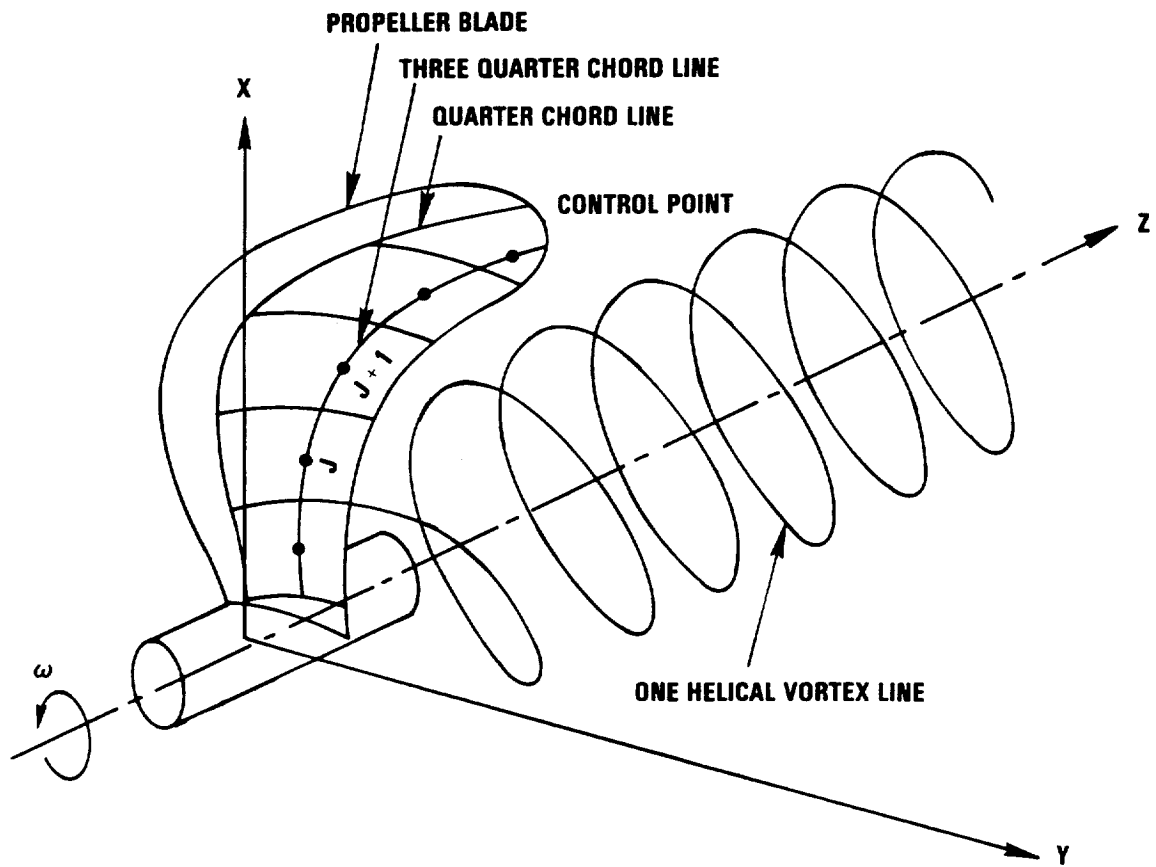


Figure 2.3-1. Propeller Coordinates System

With regard to multiple blades, we need to define a blade difference angle,  $\theta_p = \frac{2\pi}{K} (p-1)$ ,  $p = 1, 2, \dots, K$ , and  $p = 1$  implies the principal blade. A diagram showing one element in each of a four blade propeller is given in Figure 2.3-2.

#### 2.3.4 Governing Equations

With the relevant assumptions of the preceding sections, the propeller blade is replaced by a single swept bound vortex, placed at the 1/4-chord line, and helical trailing vortex lines, which are determined by the following equations:

$$x = \rho \cos \bar{\theta} \quad (2.3-1)$$

$$y = \rho \sin \bar{\theta} \quad (2.3-2)$$

$$z = \frac{V_\infty}{\omega R} \theta \quad (2.3-3)$$

where

$\omega$  = angular speed of the propeller

$R$  = radius of the blade

$\rho$  = radius of the helical vortex line

$\theta$  = angle of rotation

All the length quantities  $x$ ,  $y$ ,  $z$ , and  $\rho$  are nondimensionalized with respect to the propeller radius  $R$ . The velocity field induced by the vortex lines is obtained by the Biot-Savart law (Ref. 2.3-4).

$$\frac{\vec{v}}{V_\infty} = \frac{\Gamma}{4\pi R V_\infty} \int_0^\infty \frac{\vec{R} \times d\vec{\ell}}{|\vec{R}|^3} \quad (2.3-4)$$

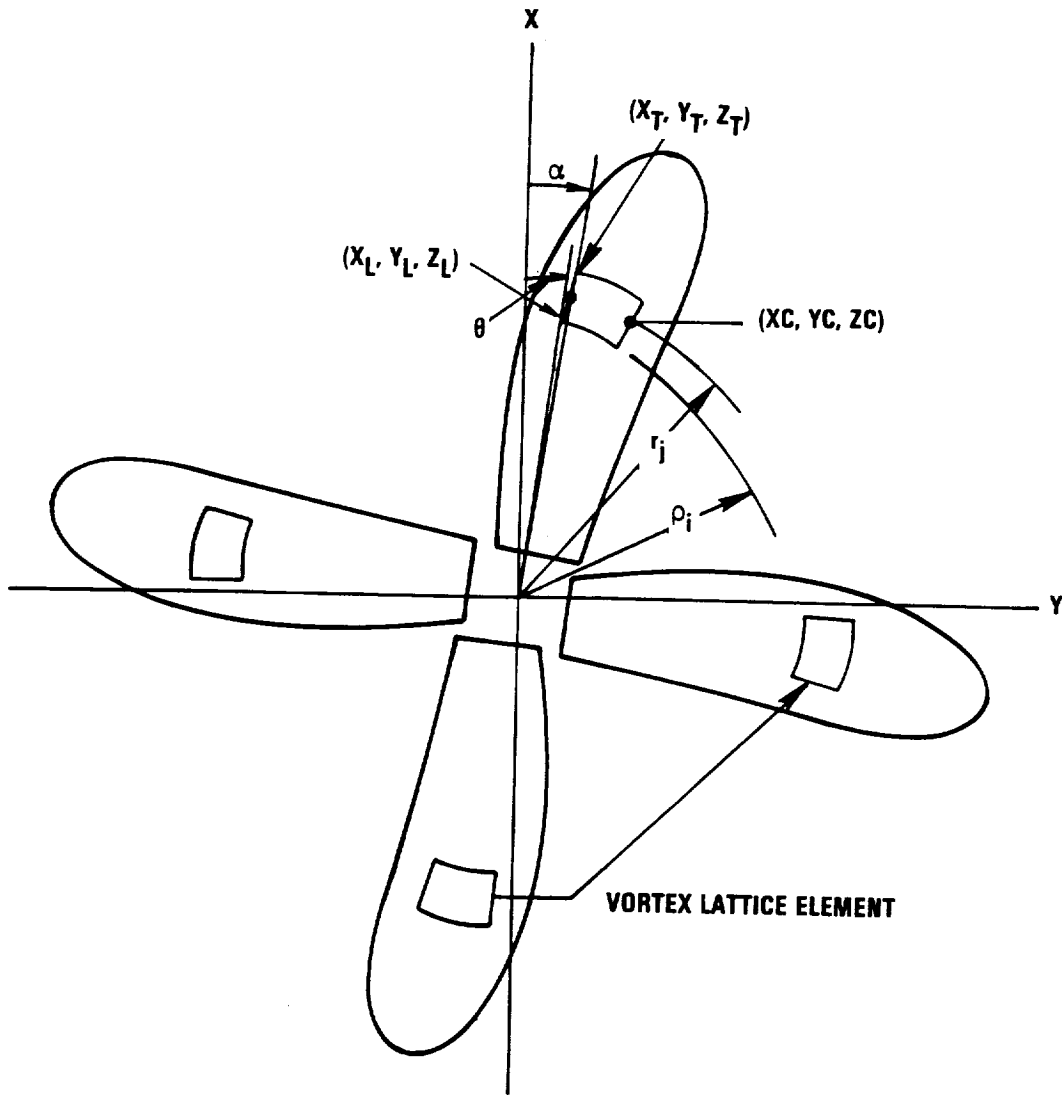


Figure 2.3-2. Vortex Lattice Elements for a Four-Blade Propeller



where

$\vec{v} = (u, v, w)$  is the induced velocity at a reference point  $P_j (x, y, z)$

$\Gamma$  = circulation of each element, ft/sec

$V_\infty$  = free stream velocity

$\vec{R}$  = distance from a reference point of the propeller to a point on the vortex line

$d\vec{\ell}$  = an elemental length of a helical vortex line vector

Now let the following equations

$$X' = r_j \cos\psi \quad (2.3-5)$$

$$Y' = r_j \sin\psi \quad (2.3-6)$$

$$Z' = Z_j \quad (2.3-7)$$

represent the coordinates of the control point where the induced velocity is sought. Then, by substituting Equations (2.3-1) through (2.3-3) and (2.3-5) through (2.3-7) into Equation (2.3-4), we obtain

$$\frac{\vec{v}}{V_\infty} = \frac{\Gamma}{4\pi R V_\infty} \int_0^\infty \frac{\vec{R} \times d\vec{\ell}}{|\vec{R}|^3}$$

where

$$\vec{R} = (\rho \cos \bar{\theta} - r_j \cos\psi) \hat{i} + (\rho \sin \bar{\theta} - r_j \sin\psi) \hat{j} + \left[ \frac{V_\infty}{\omega R} \bar{\theta} - Z_j \right] \hat{k}$$

The cross product  $\vec{R} \times d\vec{\ell}$  is given as follows:

$$\left\{ \left[ \frac{V_\infty}{\omega R} \right] \left[ \rho \sin \bar{\theta} - r_j \sin \psi - \left[ \theta - \frac{Z_j}{V_\infty / \omega R} \right] \cdot \rho \cos \bar{\theta} \right] \hat{i} + \left[ \frac{V_\infty}{\omega R} \right] \left[ -\rho \cos \bar{\theta} + r_j \cos \psi - \left[ \theta - \frac{Z_j}{V_\infty / \omega R} \right] \cdot \rho \sin \bar{\theta} \right] \hat{j} + \left[ \rho^2 - \rho r_j \cos (\bar{\theta} - \psi) \right] \hat{k} \right\} d\theta .$$

In addition,

$$|\hat{R}_1|^2 = \rho^2 + r_j^2 - 2\rho r_j \cos (\bar{\theta} - \psi) + \left[ \frac{V_\infty}{\omega R} \right]^2 \left[ \theta - \frac{Z_j}{V_\infty / \omega R} \right]^2$$

and

$$\bar{\theta} = \theta_p + \theta .$$

Then the induced velocities in the x, y, and z directions are

$$\frac{u}{V_\infty} = \frac{\Gamma_j}{4\pi R V_\infty} \int_0^\infty \sum_{n=1}^k \frac{\left[ \frac{V_\infty}{\omega R} \right] \left[ \rho \sin \bar{\theta} - r_j \sin \psi - \left[ \theta - \frac{Z_j}{V_\infty / \omega R} \right] \rho \cos \bar{\theta} \right] d\theta}{|\hat{R}_1|^3} \quad (2.3-8)$$

$$\equiv \frac{\Gamma_j}{4\pi R V_\infty} \text{FUH}$$

$$\frac{v}{V_\infty} \equiv \frac{\Gamma_j}{4\pi R V_\infty} \int_0^\infty \sum_{n=1}^k \frac{\left[ \frac{V_\infty}{\omega R} \right] \left[ -\rho \cos \bar{\theta} + r_j \cos \psi - \left[ \theta - \frac{Z_j}{V_\infty / \omega R} \right] \rho \sin \bar{\theta} \right] d\theta}{|\hat{R}_1|^3} \quad (2.3-9)$$

$$\equiv \frac{\Gamma_j}{4\pi R V_\infty} \text{FVH}$$

$$\frac{w}{V_\infty} = \frac{\Gamma_j}{4\pi R V_\infty} \int_0^\infty \sum_{n=1}^k \frac{[\rho^2 - \rho r_j \cos(\theta - \psi)] d\theta}{|R|^3} \quad (2.3-10)$$

$$\equiv \frac{\Gamma_j}{4\pi R V_\infty} \text{FWH.}$$

The velocity components induced at the reference point by the bound vortex can also be obtained by applying the Biot-Savart law. The results are derived in Reference 2.3-2 as

$$\frac{u}{V_\infty} = \frac{\Gamma}{4\pi R V_\infty} \{(Y_L - Y')(Z_T - Z_L) - (Z_L - Z')(Y_T - Y_L)\} I \equiv \frac{\Gamma}{4\pi R V_\infty} \text{FUB}$$

$$\frac{v}{V_\infty} = \frac{\Gamma}{4\pi R V_\infty} \{(Z_L - Z')(X_T - X_L) - (X_L - X')(Z_T - Z_L)\} I \equiv \frac{\Gamma}{4\pi R V_\infty} \text{FVB} \quad (2.3-11)$$

$$\frac{w}{V_\infty} = \frac{\Gamma}{4\pi R V_\infty} \{(X_L - X')(Y_T - Y_L) - (Y_L - Y')(X_T - X_L)\} I \equiv \frac{\Gamma}{4\pi R V_\infty} \text{FWB}$$

where

$$I = \frac{1}{ac - bb} \left[ \frac{a+b}{\sqrt{a+2b+c}} - \frac{b}{c} \right]$$

$$a = (X_T - X_L)^2 + (Y_T - Y_L)^2 + (Z_T - Z_L)^2$$

$$b = (X_T - X_L)(X_L - X') + (Y_T - Y_L)(Y_L - Y') + (Z_T - Z_L)(Z_L - Z')$$

$$c = (X_L - X')^2 + (Y_L - Y')^2 + (Z_L - Z')^2$$

and  $X_L$ ,  $X_T$ ,  $Y_L$ ,  $Y_T$ ,  $Z_L$ , and  $Z_T$  are coordinates of the vortex elements at the 1/4-chord line.

For a multiple blade propeller they are represented by:

$$X_L = \rho_i \cos (\theta + \theta_p) = XN_i$$

$$X_T = \rho_{i+1} \cos (\theta + \theta_p) = XN_{i+1}$$

$$Y_L = \rho_i \sin (\theta + \theta_p) = YN_i$$

$$Y_T = \rho_{i+1} \sin (\theta + \theta_p) = YN_{i+1}$$

$$Z_L = ZN_i$$

$$Z_T = ZN_{i+1}$$

### 2.3.5 Boundary Conditions

The condition of zero velocity normal to the chord plane needs to be satisfied at the control points on the blade. The resultant velocity at an arbitrary point is the vector sum of free stream velocity  $V_\infty$ , the induced velocity  $(u,v,w)$ , and the rotating velocity  $\omega r$ :

$$\vec{v} = (u - \omega r \sin \gamma) \hat{i} + (v + \omega r \cos \gamma) \hat{j} + (w + V_\infty) \hat{k} \quad (2.3-12)$$

where the angle  $\gamma$  is as shown in Figure 2.3-2. Let the surface normal of each vortex element be represented by

$$\vec{n} = n_{xj} \hat{i} + n_{yj} \hat{j} + n_{zj} \hat{k} \quad (2.3-13)$$

By substituting Equations (2.3-12) and (2.3-13) into the boundary condition  $\vec{v} \cdot \vec{n} = 0$  at the control points one obtains:

$$\left(\frac{u}{V_\infty}\right) n_{xj} + \left(\frac{v}{V_\infty}\right) n_{yj} + \left(\frac{w}{V_\infty}\right) n_{zj} = \left(\frac{\omega R}{V_\infty}\right) \left[ \frac{r_j}{R} \sin \theta_j \cdot n_{xj} - \frac{r_j}{R} \cos \theta_j \cdot n_{yj} \right] - n_{zj} \quad (2.3-14)$$

where

$$\begin{aligned} n_{xj} &= (Y_{N_{j+1}} - Y_{C_j})(Z_{N_j} - Z_{C_j}) - (Z_{N_{j+1}} - Z_{C_j})(Y_{N_j} - Y_{C_j}) \\ n_{yj} &= (Z_{N_{j+1}} - Z_{C_j})(X_{N_j} - X_{C_j}) - (X_{N_{j+1}} - X_{C_j})(Z_{N_j} - Z_{C_j}) \\ n_{zj} &= (X_{N_{j+1}} - X_{C_j})(Y_{N_j} - Y_{C_j}) - (Y_{N_{j+1}} - Y_{C_j})(X_{N_j} - X_{C_j}) \end{aligned} \quad (2.3-15)$$

The induced velocity at each control point is made up of the velocity components induced by the helical vortex lines and the bound vortex lines:

$$\begin{aligned} \frac{u_j}{V_\infty} &= \sum_{i=1}^M \frac{\Gamma_i}{4\pi R V_\infty} F_{U_{ij}} \\ \frac{v_j}{V_\infty} &= \sum_{i=1}^M \frac{\Gamma_i}{4\pi R V_\infty} F_{V_{ij}} \\ \frac{w_j}{V_\infty} &= \sum_{i=1}^M \frac{\Gamma_i}{4\pi R V_\infty} F_{W_{ij}} \end{aligned} \quad (2.3-16)$$

Substituting Equation (2.3.15) into Equation (2.3-14), the boundary condition at the control point  $j$  is

$$\sum_{i=1}^M \frac{\Gamma_i}{4\pi R V_\infty} [(FU_{ij})n_{xj} + (FV_{ij})n_{yj} + (FW_{ij})n_{zj}] \quad (2.3-17)$$

$$= \left(\frac{\omega R}{V_\infty}\right) \left(\frac{r_i}{R}\right) (\sin \theta_j \cdot n_{xj} - \cos \theta_j \cdot n_{yj}) - n_{zj}$$

$FU_{ij}$ ,  $FV_{ij}$ ,  $FW_{ij}$  are called influence coefficients, and are given by

$$FU_{ij} = FUH_{i+1,j} - FUH_{i,j} + FUB_{i,j}$$

$$FV_{ij} = FVH_{i+1,j} - FVH_{i,j} + FVB_{i,j}$$

$$FW_{ij} = FWH_{i+1,j} - FWH_{i,j} + FWB_{i,j}$$

### 2.3.6 Detailed Solution Methodology

The components of the influence coefficients are found by Equations (2.3-8) through (2.3-11). Therefore, the application of the vortex lattice method consists of first calculating the influence coefficients and then solving the system of equations (2.3-17) to find the circulation  $\Gamma_i/4\pi R V_\infty$ . Once the circulation is known, the force on each element is calculated by the Kutta-Joukowski law:

$$F_i = \rho \vec{v} \times d\vec{\Gamma}_i (2s) \quad (2.3-18)$$

in which the induced velocity is referred to at the midpoint of each vortex element on the 1/4-chord line.

### 2.3.6.1 Thrust Coefficient

Thrust is the propulsive force produced by the propeller. Thrust coefficient defined by

$$C_T = \frac{T}{\rho n^2 D^4}$$

where

$T$  = thrust force, lb

$\rho$  = air density (0.002378 slug/ft<sup>3</sup> at sea level at standard conditions)

$n$  = rotating speed, revolutions/sec

$D$  = diameter of the propeller, ft

The aerodynamic force acting on each element of length  $2s$  of the propeller is obtained as follows:

$\vec{f}^*$  can be represented by

$$\vec{f}^* = \frac{(XN_{i+1} - XN_i) \hat{i} + (YN_{i+1} - YN_i) \hat{j} + (ZN_{i+1} - ZN_i) \hat{k}}{2s} |\Gamma| \quad (2.3-19)$$

By substituting Equations (2.3-12) and (2.3-19) into Equation (2.3-18), one obtains

$$(F_x)_L = \rho \Gamma [(v + w r \cos \gamma)(Z_T - Z_L) - (V_\infty + w)(Y_T - Y_L)]$$

$$(F_y)_L = \rho \Gamma [(V_\infty + w)(X_T - X_L) - (u - w r \sin \gamma)(Z_T - Z_L)]$$

$$(F_z)_L = \rho \Gamma [(u - w r \sin \gamma)(Y_T - Y_L) - (v + w r \cos \gamma)(X_T - X_L)]$$

The drag force

$$\vec{F}_D = \frac{\vec{V}}{|\vec{V}|} \cdot D$$

$$(F_x)_D = D \left[ \frac{u - w \sin \gamma}{|\vec{V}|} \right]$$

$$(F_y)_D = D \left[ \frac{v + w \cos \gamma}{|\vec{V}|} \right]$$

$$(F_z)_D = D \left[ \frac{w + V_\infty}{|\vec{V}|} \right]$$

$$D = C_d \frac{1}{2} \rho V_{R_o}^2 C_f (2s)$$

$$L = C_l \frac{1}{2} \rho V_{R_o}^2 C_f (2s) = \rho V_{R_o} \Gamma_f (2s)$$

$$(F_x)_D = \frac{C_d}{C_l} \rho \Gamma_f (u - w \sin \gamma) (2s)$$

$$(F_y)_D = \frac{C_d}{C_l} \rho \Gamma_f (v + w \cos \gamma) (2s)$$

$$(F_z)_D = \frac{C_d}{C_l} \rho \Gamma_f (w + V_\infty) (2s)$$

The force components in x, y, z direction are then given by

$$F_x = \rho \Gamma \left[ (v + w \cos \gamma) (Z_T - Z_L) - (w + V_\infty) (Y_T - Y_L) + \frac{C_d}{C_l} (u - w \sin \gamma) (2s) \right]$$



$$F_y = \rho \Gamma \left[ (w+V_\infty)(X_T-X_L) - (u-wr \sin \gamma)(Y_T-Y_L) + \frac{C_d}{C_\ell} (v+wr \cos \gamma)(2s) \right]$$

$$F_z = \rho \Gamma \left[ (u-wr \sin \gamma)(Y_T-Y_L) - (v+wr \cos \gamma)(X_T-X_L) + \frac{C_d}{C_\ell} (w+V_\infty)(2s) \right]$$

The thrust force on each vortex element is in the negative Z-direction, i.e.,  $T = -F_z$ .

$T$  is nondimensionalized by  $\rho n^2 D^4$ . The thrust loading is obtained by

$$\frac{dC_T}{d\left(\frac{r_i}{R}\right)} = B\pi^3 \left(\frac{V_\infty}{\omega R}\right)^2 \left(\frac{\Gamma_i}{4\pi V_\infty R}\right) \left\{ \left[ \left(\frac{\omega R}{V_\infty}\right) \frac{r_i}{R} \cos \gamma + \frac{v}{V_\infty} \right] \left(\frac{X_T-X_L}{2s}\right) - \left[ \frac{u}{V_\infty} - \left(\frac{\omega R}{V_\infty}\right) \frac{r_i}{R} \sin \gamma \right] \left(\frac{Y_T-Y_L}{2s}\right) - \frac{C_d}{C_\ell} \left(1 + \frac{w}{V_\infty}\right) \right\}$$

The thrust coefficient of the propeller is calculated by integration of the thrust loading along the blade radius, using the equation

$$C_T = \int_{\text{HUB}}^1 \frac{dC_T}{d\left(\frac{r_i}{R}\right)} d\left(\frac{r_i}{R}\right)$$

### 2.3.6.2 Power Coefficient

By analogy with the last section, the power coefficient can be derived. Let  $Q_i$  be the aerodynamic torque on a blade element.

$$Q_i = F_{\theta i} \cdot r_i$$

where

$$F_{\theta i} = F_{y i} \cdot \cos \gamma - F_{x i} \cdot \sin \gamma$$

The power coefficient  $C_p$  of each blade element is defined by

$$C_p = \frac{P}{\rho n^3 D^5}$$

and

$$C_{p_i} = 2\pi C_{Q_i}$$

where

$$C_{Q_i} = \frac{Q_i}{\rho n^2 D^5}$$

By considering the drag effect, the power loading is determined by

$$\begin{aligned} \frac{dC_p}{d\left(\frac{r_i}{R}\right)} = & B\pi^4 \left(\frac{V_\infty}{\omega R}\right)^2 \left(\frac{r_i}{R}\right) \left(\frac{\Gamma_i}{4\pi V \omega R}\right) \left\{ \left[ \left(1 + \frac{w}{V_\infty}\right) \left(\frac{X_T - X_L}{2s}\right) - \right. \right. \\ & \left. \left. \left(\frac{u}{V_\infty} - \left(\frac{\omega R}{V_\infty}\right) \frac{r_i}{R} \sin\gamma\right) \left(\frac{Z_T - Z_L}{2s}\right) + \right. \right. \\ & \left. \left. \left(\frac{C_d}{C_l}\right) \left(\frac{v}{V_\infty} + \left(\frac{\omega R}{V_\infty}\right) \frac{r_i}{R} \cos\gamma\right) \right] \cos\gamma - \left[ \left(\frac{\omega R}{V_\infty}\right) \frac{r_i}{R} \cos\gamma + \frac{v}{V_\infty} \right] \left(\frac{Z_T - Z_L}{2s}\right) - \right. \\ & \left. \left. \left[1 + \frac{w}{V_\infty}\right] \left(\frac{Y_T - Y_L}{2s}\right) + \left(\frac{C_d}{C_l}\right) \left[\frac{u}{V_\infty} - \left(\frac{\omega R}{V_\infty}\right) \frac{r_i}{R} \sin\gamma\right] \right] \sin\gamma \right\} \end{aligned}$$

The power coefficient of the propeller is:

$$C_p = \int_{HUB}^1 \frac{dC_p}{d\left(\frac{r_i}{R}\right)} d\left(\frac{r_i}{R}\right)$$

### 2.3.6.3 Radial Force Coefficient

The propeller blade may be shaped in a variety of ways in modern design technology for energy saving purposes. A force may exist in the radial direction in any part of the blade for complex blade shapes. The force in the radial direction is represented by

$$F_R = F_x \cdot \cos\gamma + F_y \cdot \sin\gamma$$

In nondimensional form:

$$C_R = \frac{F_R}{\rho n^2 D^4}$$

$$\begin{aligned} \frac{dC_R}{d\left(\frac{r_i}{R}\right)} = & B\pi^3 \left(\frac{V_\infty}{\omega R}\right)^2 \left(\frac{\Gamma_i}{4\pi V_\infty R}\right) \left\{ \left[ \left( \frac{v}{V_\infty} + \left(\frac{\omega R}{V_\infty}\right) \frac{r_i}{R} \cos\gamma \right) \left( \frac{Z_T - Z_L}{2s} \right) - \left( 1 + \frac{w}{V_\infty} \right) \cdot \right. \right. \\ & \left. \left( \frac{Y_T - Y_L}{2s} \right) + \frac{C_d}{C_\ell} \left( \frac{u}{V_\infty} - \left(\frac{\omega R}{V_\infty}\right) \frac{r_i}{R} \sin\gamma \right) \right] \cos\gamma + \\ & \left[ \left( 1 + \frac{w}{V_\infty} \right) \left( \frac{X_T - X_L}{R} \right) - \left( \frac{u}{V_\infty} - \left(\frac{\omega R}{V_\infty}\right) \frac{r_i}{R} \sin\gamma \right) \left( \frac{Z_T - Z_L}{2s} \right) + \frac{C_d}{C_\ell} \cdot \right. \\ & \left. \left. \left( \frac{v}{V_\infty} + \left(\frac{\omega R}{V_\infty}\right) \frac{r_i}{R} \cos\gamma \right) \right] \sin\gamma \right\} \end{aligned} \quad (2.3-22)$$

and the radial force coefficient is determined by

$$C_R = \int_{\text{HUB}}^1 \frac{dC_R}{d\left(\frac{r_i}{R}\right)} d\left(\frac{r_i}{R}\right)$$

### 2.3.7 Numerical Techniques

#### 2.3.7.1 Convergence Analysis

The integral expressions of the induced velocities derived in the previous sections cannot be solved by analytical means. While Lamb (Ref. 2.3-5) has given an expression for a single infinite helical vortex in terms of Bessel functions, the difficulty of the evaluation of the principal value of an infinite integral along the blade makes the use of this method prohibitive.

When the induced velocity integral is solved numerically, two questions will naturally arise: (1) where should we truncate the integration, and (2) what will be the convergence characteristics if we truncate the integral at an upper integration limit? These two problems can be resolved by error analysis which is presented in this section.

In numerical analysis, error can be defined as: (1) truncation error, and (2) integration error. The former results from the truncation of integration, and the second comes strictly from the numerical integration formula.

Before performing the analysis, it is advisable to investigate the nature of variations of the integrands in the integral expression, corresponding to different values of  $\theta$ . Computer subroutines denoted by FUHINT, FVHINT, and FWHINT are written for calculating the integrands of the integral for estimating the induced velocities of  $u$ ,  $v$ ,  $w$ , respectively. The calculations are based on the SR-2 propeller at  $J = 3.06$ , and  $\beta_{3/4} = 60^\circ$ . The results of FUHINT are presented in Figures 2.3-3 and 2.3-4. Similar results are obtained for FVHINT and FWHINT. A general feature of these results is that drastic changes of FUHINT, FVHINT, or FWHINT versus the angle  $\theta$  occur in values of  $\theta < 0.4$  radian, which corresponds to 0.0636 revolution. At  $\theta = 20$  radians, FUHINT = -0.00030, FVHINT = -0.00070, FWHINT = 0.00001. Values of the integrands FUHINT, FVHINT, and FWHINT, at different revolutions are presented in Table 2.3-1.

As indicated by these results, the computer program of integration should be well programmed, and relatively small intervals should be used to guarantee the desired accuracy.

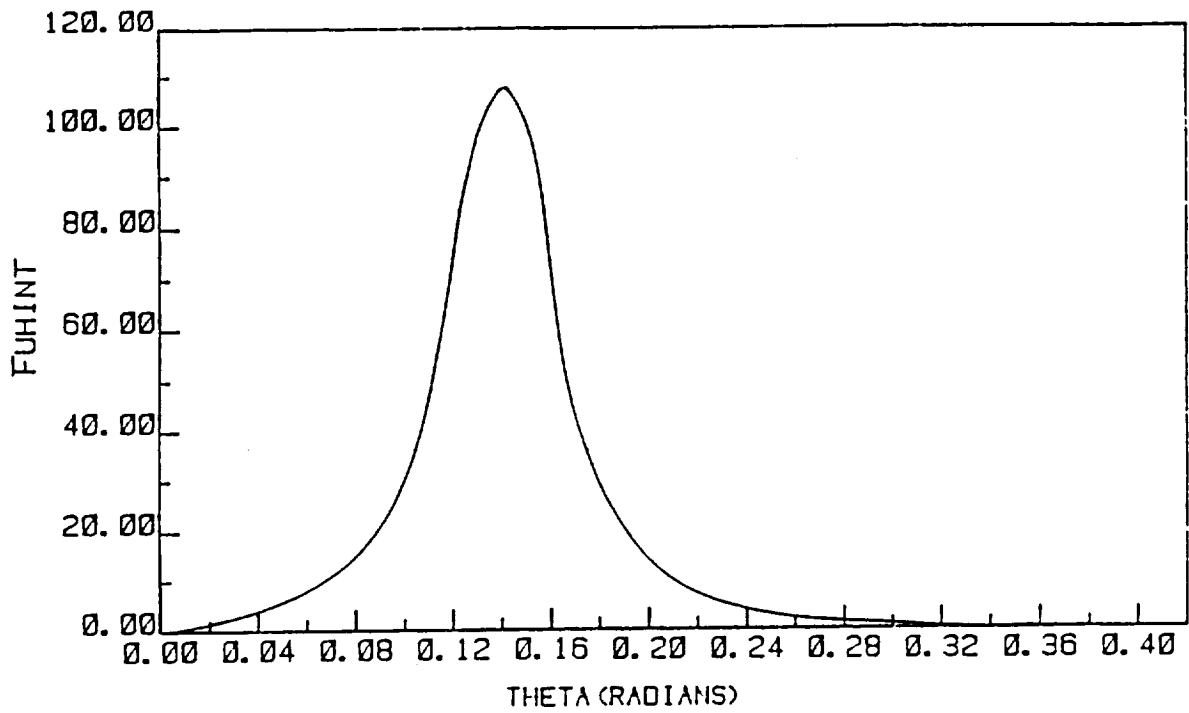


Figure 2.3-3. Variation of FUHINT Versus  $\theta$  from 0 to 0.40 Radian

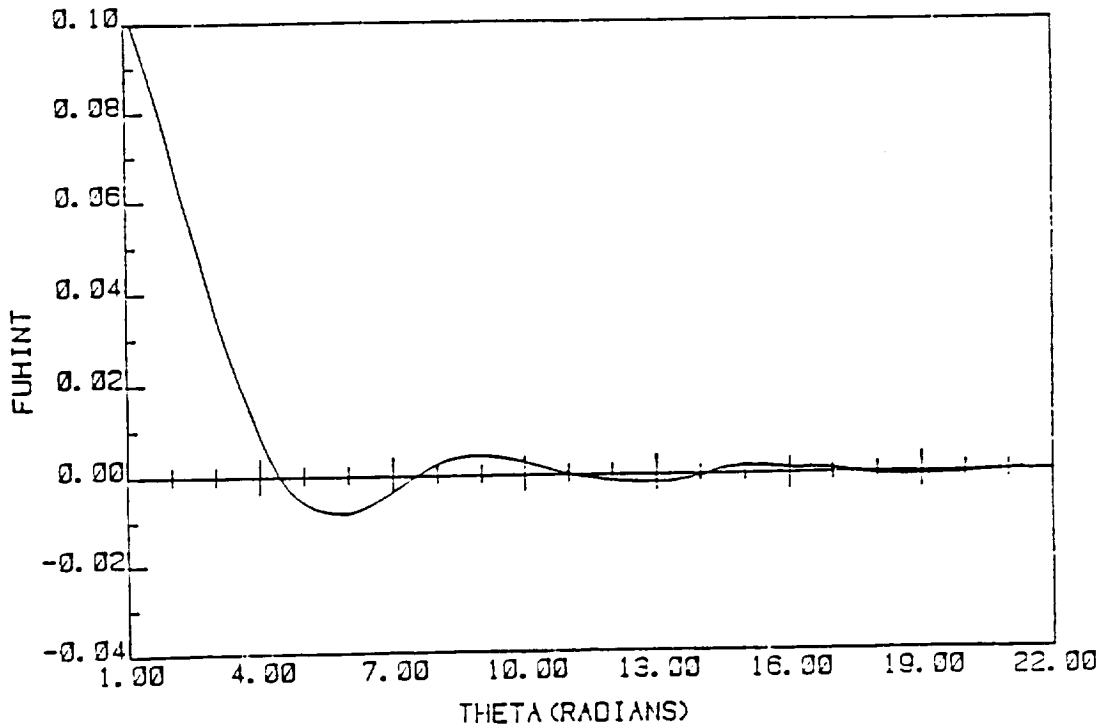


Figure 2.3-4. Variation of FUHINT Versus  $\theta$  from 1 to 20 Radian

Table 2.3-1. Values of FUHINT, FVHINT, and FWHINT at Different Revolutions of Integration

Number of Revolutions	FUHINT	FVHINT	FWHINT
0.5	0.03143250	0.02303523	0.00709475
1.0	-0.00854547	0.00014987	-0.00004620
1.5	0.00359349	0.00082761	0.00025488
2.0	-0.00206796	0.00001806	-0.00000558
2.5	0.00129113	0.00017642	0.00005432
3.0	-0.00090924	0.00000527	-0.00000163
3.5	0.00065769	0.00006389	0.00001967
4.0	-0.00050871	0.00000219	-0.00000068
4.5	0.00039744	0.00002996	0.00000922
5.0	-0.00032452	0.00000111	-0.00000035
5.5	0.00026585	0.00001637	0.00000503
6.0	-0.00022488	0.00000063	-0.00000020
6.5	0.00019024	0.00000991	0.00000304
7.0	-0.00016496	0.00000039	-0.00000012
7.5	0.00014283	0.00000644	0.00000198
8.0	-0.00012615	0.00000026	-0.00000008
8.5	0.00011116	0.00000442	0.00000135
9.0	0.00009959	0.00000018	-0.00000005
9.5	0.00008896	0.00000317	0.00000097
10.0	-0.00008061	0.00000012	-0.00000004

### 2.3.7.2 Truncation Error

The truncation error can be estimated at a large value of  $\theta$  of 10 revolutions. Since  $|\dot{R}|$  is in the denominator, we then approximate  $|\dot{R}|$  by a small value of  $|\dot{R}|$  for large  $\theta = \theta_u$ . For large values of  $\theta$ ,  $\frac{z_j}{V_\infty/\omega R}$  can be neglected for  $\frac{z_j}{V_\infty/\omega R} \ll \theta_u$ . The value of  $2 \rho r_j \cos(\bar{\theta} - \psi)$  is approximated by  $2 \rho r_j$ , which gives less contribution to the value of the denominator. That is

$$\rho^2 + r_j^2 - 2 \rho r_j \cos(\bar{\theta} - \psi) \approx$$

$$\rho^2 + r_j^2 - 2 \rho r_j = (\rho - r_j)^2$$

= 0 corresponding to the lowest contribution of those terms. We then obtain the result:

$$|\dot{R}| > \frac{V_\infty}{\omega R} \theta$$

The numerator of Equation (2.3-8) is

$$Nu = \left( \frac{V_\infty}{\omega R} \right) \left[ \rho \sin \bar{\theta} - r_j \sin \psi - \left( \theta - \frac{z_j}{V_\infty/\omega R} \right) \rho \cos \bar{\theta} \right]$$

The approximation principle is to find a larger value of Nu based upon a large value of  $\theta$ . In this case

$$Nu \leq \frac{V_\infty}{\omega R} \theta$$

The error for  $\frac{u}{V_\infty}$  is then obtained by:

$$\frac{u}{V_\infty} < \frac{\Gamma}{4\pi R V_\infty} \int_{\theta_u}^{\infty} \sum_{n=1}^k \frac{d\theta}{(V_\infty/\omega R)^2 \theta^2} = \frac{-\Gamma}{4\pi R V_\infty (V_\infty/\omega R)^2} K \left( \frac{1}{\theta} \right) \Big|_{\theta_u}^{\infty}$$

For  $K = 8$

$$\frac{u}{V_\infty} = \left( \frac{\Gamma}{4\pi R V_\infty} \right) (8) \left( \frac{\pi}{J} \right)^2 \left( \frac{1}{62.8} \right)$$

For the SR-2 propeller,  $\left( \frac{\Gamma}{4\pi R V_\infty} \right)_{\max} = 0.03$ ,  $J = 3.06$

$$\left( \frac{u}{V_\infty} \right) = 0.03 (8) \left( \frac{\pi}{J} \right)^2 \frac{1}{62.8} = 0.00403$$

This is a very conservative figure due to the approximation of the numerator and the denominator of the integrand.

The truncation error of  $\frac{v}{V_\infty}$  can also be obtained following the same line as above.

$$\frac{v}{V_\infty} = \frac{\Gamma}{4\pi R V_\infty} \int_0^{\infty} \sum_{n=1}^k \frac{\left( \frac{V_\infty}{\omega R} \right) \left[ -\rho \cos \bar{\theta} + r_j \cos \psi - \left( \theta - \frac{z_j}{V_\infty/\omega R} \right) \rho \sin \bar{\theta} \right] d\theta}{|R_1|^3} \quad (2.3-9)$$

thus the truncation error is the same as  $\frac{u}{V_\infty}$ .



The truncation error of  $\frac{w}{V_\infty}$  is

$$\begin{aligned} \frac{w}{V_\infty} &= \frac{\Gamma}{4\pi R V_\infty} \int_{\theta_u}^{\infty} \sum_{n=1}^k \frac{[\rho^2 - \rho r_j \cos(\bar{\theta} - \psi)] d\theta}{|\dot{R}|^3} \\ &< \frac{\Gamma}{4\pi R V_\infty} \sum_{n=1}^k \int_{\theta_u}^{\infty} \frac{d\theta}{(V_\infty / \omega R)^3 \theta^3} \\ &= \left( \frac{\Gamma_i}{4\pi R V_\infty} \right) \left( \frac{\pi}{J} \right)^3 (4) \left( \frac{1}{2} \right) \left[ -\frac{1}{\theta^2} \right] \Big|_{\theta_u}^{\infty} \\ &= \left( \frac{\Gamma_i}{4\pi R V_\infty} \right) \left( \frac{\pi}{J} \right)^3 (4) \left( \frac{1}{62.8} \right)^2 = 0.000033 \end{aligned}$$

The truncation errors estimated above represent the upper bound of the error, based upon  $\theta_u = 10$  revolutions. This fact comes from the approximation principle employed. These values are too conservative to be used in the application, but they give us a general idea about the error of truncation.

### 2.3.7.3 Error of Integration

The error of integration is usually affected by the selection of spacing, the degree of interpolation polynomial, and the location of the base points. The first degree polynomial results in the trapezoidal rule, which is familiar in numerical analysis:

$$T_{N,1} = \int_a^b f(x) dx = \frac{(b-a)}{2} [f(a)+f(b)] - \frac{(b-a)^3}{12} f''(\xi), \quad x_0 < \xi < x_1,$$

where  $-\frac{(b-a)^3}{12} f''(\xi)$  is the integration error, which will decrease by a factor of  $1/2^3$  as the interval of  $h$  is halved. While higher order formulas could give smaller error terms, a low order integration formula is usually

preferred with successively subdividing the interval of integration and using the formula separately on each subinterval. One routine of automatic integration is called CADRE, which is based on successively interval halving and Romberg integration (Ref. 2.3-6), which is described as follows.

Let  $f(x)$  be integrated in an interval  $[a,b]$ . The general formula for trapezoidal rule (Reference 2.3-6) is

$$T_{N,1} = \frac{1}{2N} \left\{ T_{N-1,1} + \frac{(b-a)}{2^{N-1}} \sum_{\substack{i=1 \\ \Delta i=2}}^{2^N} f\left(a + \frac{b-a}{2^N} i\right) \right\} \text{ for } N=1,2, \dots N.$$

A sequence of  $T_{0,1}, T_{1,1}, \dots, T_{N,1}$  is then generated. By applying Richardson's extrapolation formula,

$$T_{N,J} = \frac{4^{J-1} T_{N+1,j-1} - T_{N,j-1}}{4^{J-1} - 1}$$

to each pair of the adjacent elements of the sequence, Bauer, et al. as cited in Reference 2.3-6, showed that each of the sequence  $T_{N,j}$  for  $j = 1, 2, \dots$  converges to the true integral value with an increasing number of  $N$ .

The function CADRE returns an approximate value of the true value of the integral  $\int_a^b f(x)dx$ , which is denoted by INT in the function program. The requirement that must be satisfied is that the absolute value of  $(INT - CADRE)$  should be less than the maximum value of absolute error AERR or the relative error RERR absolute (INT).

In this scheme, the program integral is calculated as the sum of integrals over suitably small subintervals, and the absolute and relative errors are found. If the error satisfies the requirement, the value of integration is accepted, otherwise the subinterval is halved.

In the Chang-Sullivan code, the function CADRE is used for performing the integration along the helical vortex lines. Table 2.3-2 represents the results of the output by using CADRE for the NASA SR-2 propeller for different numbers of revolutions. It is clear from the table that the thrust coefficient, the power coefficient, and the efficiency are all accurate to the hundred thousandth digit after five revolutions.

Table 2.3-2. Values of  $C_T$ ,  $C_p$ , and  $\eta$  at Different Revolutions of Integration SR-2,  $J = 3.06$ ,  $\beta = 60^\circ$

Number of Revolutions	Thrust Coefficient $C_T$	Power Coefficient $C_p$	Efficiency $\eta$
1	0.065469	0.221508	0.904411
2	0.065430	0.221401	0.904321
3	0.065423	0.221380	0.904305
4	0.065421	0.221372	0.904300
5	0.065419	0.221369	0.904297
6	0.065419	0.221367	0.904296
7	0.065418	0.221366	0.904296
8	0.065418	0.221365	0.904295
9	0.065418	0.221365	0.904294
10	0.065418	0.221364	0.904294

### 2.3.8 Dictionary of Input Variables

A list of the input labels is shown below. Labels with no units specified represent nondimensional quantities.

<u>Input Labels</u>	<u>Description</u>
ADV	Advance ratio, $V/nD$ , where $V$ is the advancing velocity, $n$ the rotating speed, and $D$ the propeller diameter.
BETA3Q	Design blade angle at 0.75R of the blade (degree).
BETAST	Blade angle at 0.75 R for the current run (degree).
BETAW	Cant angle of the proplet. Proplets are excluded by setting $BETAW = 0.0$ and $MPROP = M$ , where $M$ is the number of the horseshoe vortex lattice elements (degree).

BLADE	Number of propeller blades.
CE	Chord Length at each blade station (inch).
DECLC	Design lift coefficient at each blade station.
DELHT	Distance from propeller rotation axis to blade spanwise reference location (inch).
HT	Spanwise location for blade section characteristics (inch).
HOBE	Thickness ratio of the airfoil section at each blade station.
ID2	Control parameter for the input of inflow velocity of propeller. If ID2 = 1, the inflow velocity is uniform, and ID2 $\neq$ 1, nonuniform.
M	Number of vortex lattice elements.
MPROP	Number of blade element on which the proplet is installed.
N	Number of blade stations where the blade geometry is input.
OR	The distance from the airfoil mean line at 0.75 chord position to the chord line.
RBLADE	Propeller radius.
RHUB	Radius of the propeller hub.
SA	Blade angle (degree).
ALEA	Blade leading edge location (inch).
SLO	Slope of the mean line at 0.75 chord relative to the chord line.
TITLE	Title input.

XM Flight Mach number.

VN Propeller inflow velocity at element endpoints  
(nondimensionalized by flight velocity).

Input Cards and Formats

Card No.	Labels	Formats
1	TITLE	(8x,12A6)
1A	RPM	(I10)
2	M, RHUB, RBLADE, BLADE, ADV, BETA3Q, BETAST, BETAW	(I10,7F10.5)
3	MPROP, ICAMB, ID2, XM	(3I10, F10.5)
4	DELHT, N	(F10.4, I10)
5	HT(I), ALEA(I), CE(I), SA(I), HOBE(I), DECLE(I), I = 1, N	(6F10.4)
5+N	SLO, OR	(2F11.5)
6+N	VN(I) I = 1, M	(8F10.4)

2.3.9 Program Capabilities; User Options

The capabilities of the Chang-Sullivan methodology are that it calculates propeller aerodynamic performance for straight and swept blades, including proplets. In addition, it includes the effects of compressibility in the drag coefficient, cascade effects (for thin blades), and includes nacelle effects by treating the nacelle as an infinite cylinder. There is, however, no provision for supersonic tip effects or correction for finite blade-thickness effects. User options are the inclusion or exclusion of proplets and the selection of uniform or nonuniform inflow velocity at the propeller plane.

### 2.3.10 Program Validation

The program has been validated by comparison with experimental data. Verification calculations carried out by TRW for the SR-3 propeller at advance Mach numbers of 0.8, 0.45, and 0.2 are shown in Figures 2.3-5, 2.3-6, and 2.3-7, respectively, in which power coefficient and propeller efficiency are shown as functions of advance ratio. Note the excellent agreement for the Mach 0.8 and 0.45 cases, and fair agreement for the Mach 0.2 case. The fall-off in accuracy at the lower advance Mach number is most probably due to flow separation effects in calculations at the higher blade section angles of attack at this Mach number. Extensive comparisons of calculations with data, and of proplet effects, are also given in Reference 2.3-1 for the SR-1, SR-2, SR-3, NACA 109622, and Cessna 172 propellers. Comparisons of power coefficients and efficiency for two-bladed propellers show good agreement. For propellers having more than two blades, the underprediction of induced effects results in an overprediction of ideal efficiency, however.

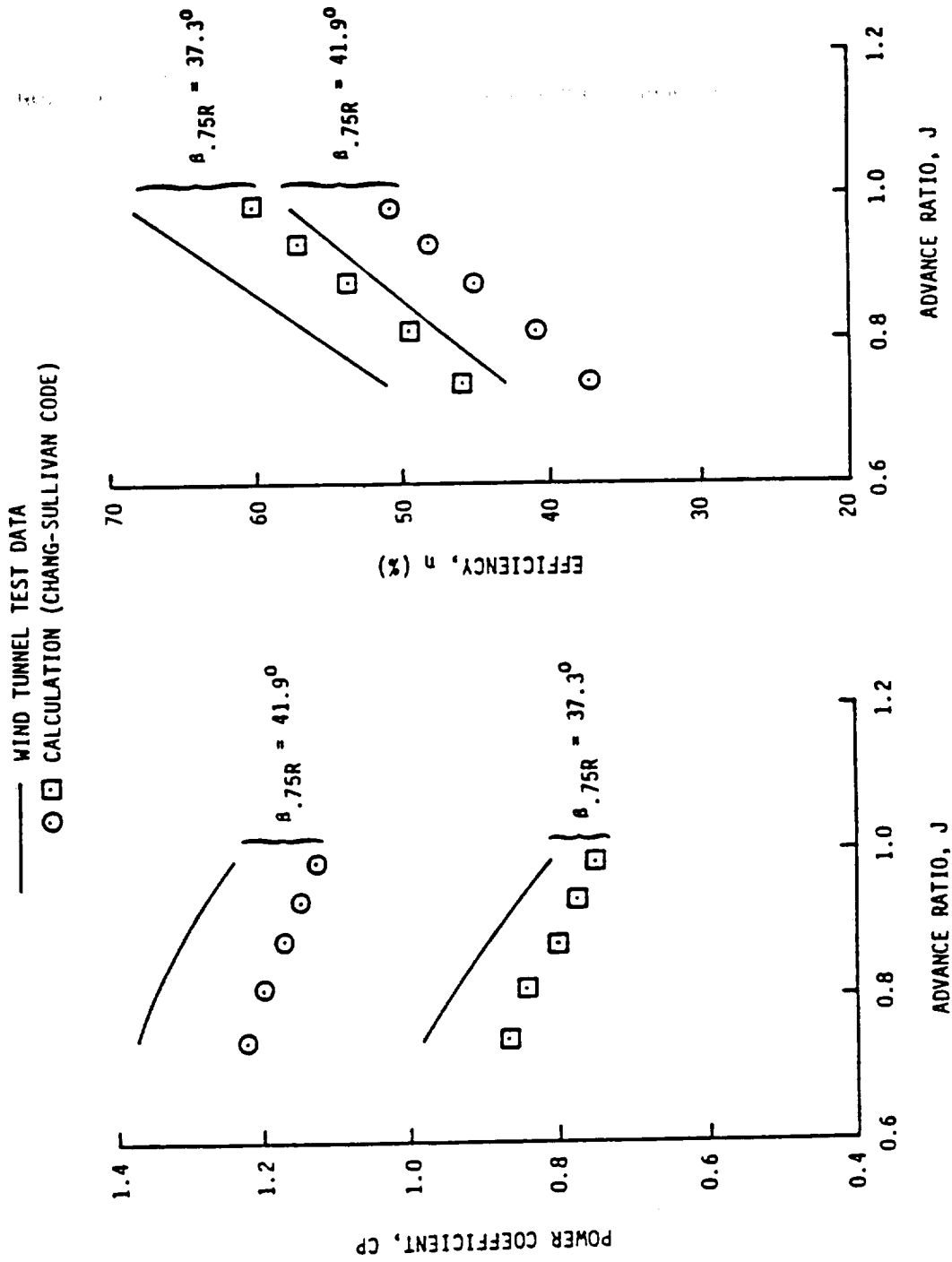


Figure 2.3-5. Aerodynamic Performance of SR-3 Propeller,  $M_0 = 0.2$

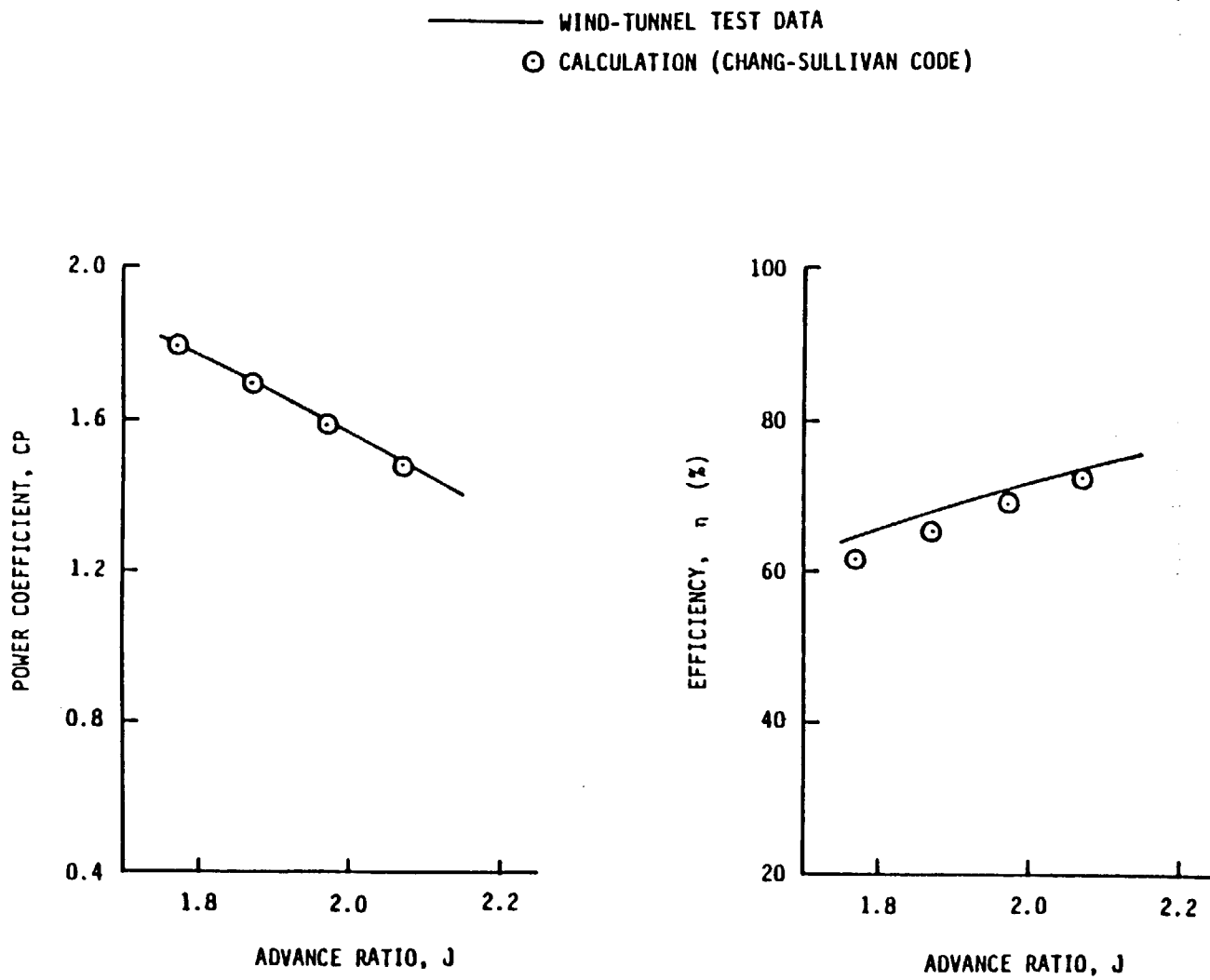


Figure 2.3-6. Aerodynamic Performance of SR-3 Propeller  
 $M_0 = 0.45$ ,  $\beta_{.75R} = 53.5^\circ$



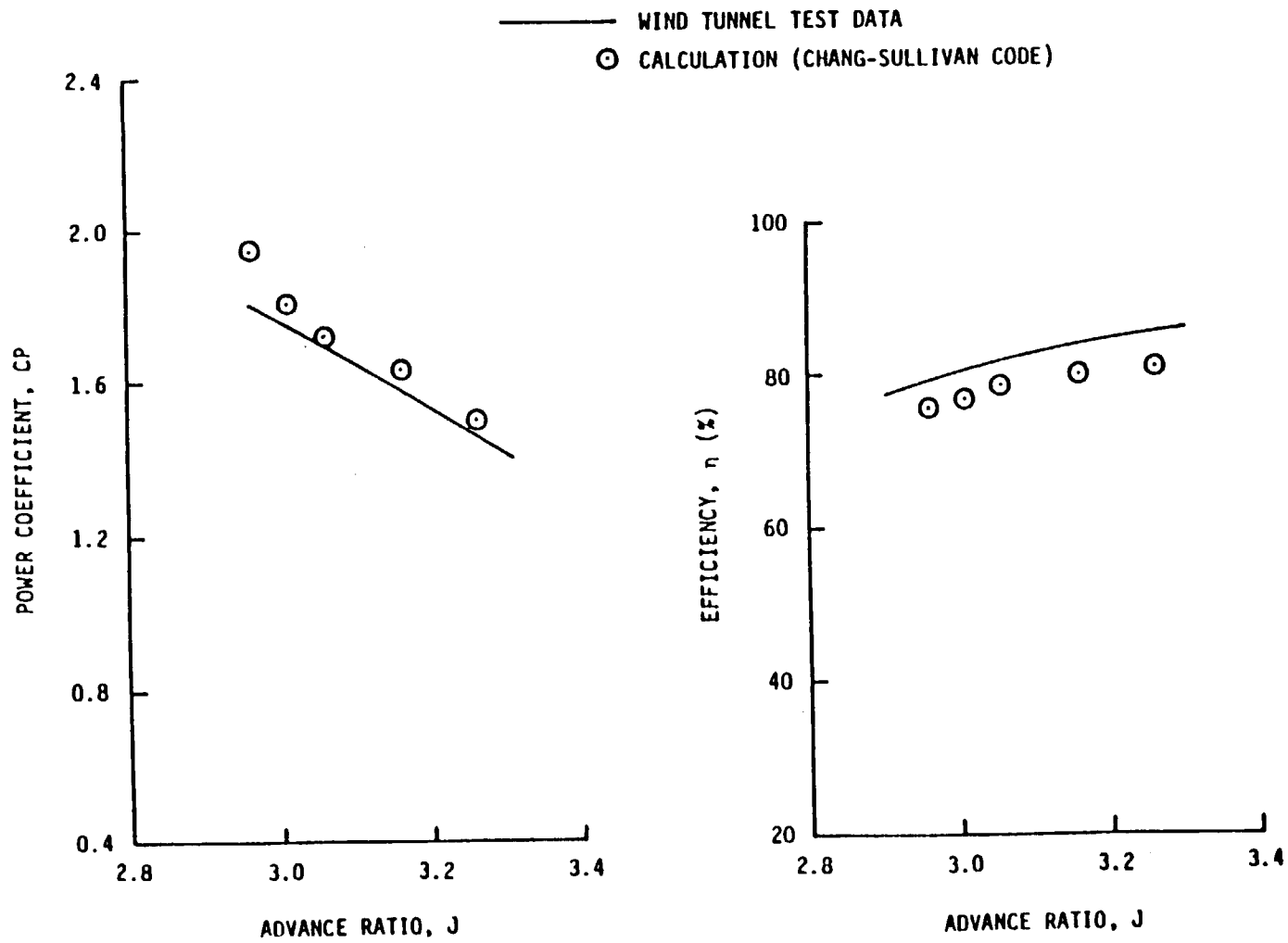


Figure 2.3-7. Aerodynamic Performance of SR-3 Propeller  
 $M_0 = 0.80$ ,  $\beta.75R = 60.95^\circ$

## 2.4 AIRFOIL LOADING MODULE

### 2.4.1 Airfoil Analysis Method: TRANSEP

A numerical technique for analyzing transonic airfoils is presented. The method employs the basic features of Jameson's iterative solution for the full potential equation, except that cartesian coordinates are used rather than a grid which fits the airfoil, such as the conformal circle-plane or "sheared parabolic" coordinates that have been used previously. Comparison with previous results shows that it is not necessary to match the computational grid to the airfoil surface and that accurate results can be obtained with a cartesian grid for lift in supercritical airfoils.

The first practical numerical technique for solving two-dimensional steady transonic flows was developed by Murman and Cole (Ref. 2.4-1) for the small perturbation equations in cartesian coordinates. The essential features of this technique were the utilization of retarded (upwind) differences in regions of supersonic flow, central differences in subsonic zones, and the application of a far-field boundary condition at a finite distance from the airfoil. The surface flow-tangency boundary condition was applied on the horizontal axis, consistent with the small disturbance approximation, and the resultant finite difference equations were solved iteratively by numerical relaxation. Shock waves occurred naturally in the course of solution as steep compressions smeared over several mesh points, and accurate results were obtained for slender airfoils.

This technique was subsequently extended to the complete potential flow equations and the exact boundary conditions by Garabedian and Korn (Ref. 2.4-2), Steger and Lomax (Ref. 2.4-3), and Jameson (Ref. 2.4-4) and was successfully applied to thick, blunt-nosed, and aft-cambered airfoils. In these investigations, the edge of the computational coordinate system was closely matched and aligned with the airfoil surface in order to accurately represent the surface flow-tangency boundary condition. For example, Steger and Lomax used a curvilinear wraparound system, while Garabedian and Korn and also Jameson conformally mapped the exterior of the airfoil onto the interior of a unit circle with infinity corresponding to the circle origin. Jameson also used a parabolic type system (Ref. 2.4-4) in another version designed to handle supersonic freestreams. These

transformations had the advantage that near the body they aligned the coordinates with the flow, which is particularly advantageous in the leading edge region where body slopes and vertical velocities can be large. Unfortunately, such transformations introduce into the differential equation many, sometimes complicated, transformation derivatives.

Now in order for any backward difference scheme at supersonic points to work properly, the coordinate along which the backward differences are taken must be closely aligned with the flow direction. This alignment is very difficult to achieve with mapped coordinates when the freestream Mach number is high or the supersonic zone is large. On the other hand, improper alignment at supersonic points can lead to situations where the velocity component in each coordinate direction is subsonic, that is

$$U^2, V^2 < a^2 < U^2 + V^2 \quad (2.4-1)$$

This situation not only can lead to numerical instability but also can impart to the finite difference equations an incorrect zone of dependence. To remedy this problem South and Jameson (Refs. 2.4-4 and 2.4-5) have introduced a rotated finite difference scheme which simulates a local rotation to coordinates along and normal to the velocity vector; and, in so doing, they have created a scheme that not only has the correct zone of dependence but also does not require coordinate alignment with the flow.

As a consequence of this development, questions arose. Can the full potential flow equations be solved for a transonic case in a simple cartesian or stretched cartesian system by using the concept of rotated differences? If so, what is the accuracy and what are the properties of such an approach? In order to answer these questions, the present study was initiated. Its primary objectives have been to:

- (1) Develop an analysis program using cartesian coordinates (airfoil given,  $C_p$  and flow field unknown)
- (2) Develop a design program in cartesian coordinates ( $C_p$  given, airfoil and flow field unknown)
- (3) Study the results obtained by such an approach and determine the accuracy and idiosyncrasies associated with a cartesian system.

This section describes the development of, and results obtained with, the TRANSEP analysis program. Reference 2.4-6 discusses the design program.

### Symbols

$A_1, A_2, A_3$	Coordinate stretching constants
$a$	Isentropic speed of sound
$a, b$	Coordinate stretching constants
$C_p$	Pressure coefficient, $(p-p_\infty)/(1/2 \rho_\infty U_\infty^2)$
$C_L$	Lift coefficient
$C_{MLE}$	Coefficient of moment about the leading edge
$f, g$	Coordinate stretching functions
$M$	Mach number
$N$	Coordinate normal to the streamwise direction
$p$	Pressure
$q$	Velocity
$S$	Streamwise coordinate
$t$	Time
$U, V$	Velocity component in the x-, y- direction respectively
$x, y$	Cartesian coordinates
$\alpha$	Angle of attack or coefficient in Equation (2.4-34)
$\beta^2$	$1-M_\infty^2$ or coefficient in Equation (2.4-34)
$\gamma$	Ratio of specific heats or coefficient in Equation (2.4-34)
$\Gamma$	Circulation
$\epsilon$	Damping coefficient
$\theta$	Polar Coordinate
$\xi, \eta$	Computational coordinates
$\rho$	Density
$\Phi$	Potential function, Equation (2.4-3)
$\phi$	Perturbation potential, Equation (2.4-4)
Subscripts:	
$\infty$	Freestream condition
$b, \text{body}$	Body
TE	Trailing edge
$i, j$	Grid location

$\xi, \eta, x, y$  Differentiation, i.e.,  $f_x = \frac{\partial f}{\partial x}$

Superscripts:

+ Value computed by current relaxation sweep

### 2.4.1.1 Problem Formulation

The exact equation for the potential function for two dimensional compressible flow can be written in cartesian coordinates as

$$(a^2 - \phi_x^2)\phi_{xx} - 2\phi_x\phi_y\phi_{xy} + (a^2 - \phi_y^2)\phi_{yy} = 0 \quad (2.4-2)$$

where

$$\phi_x = \frac{\partial \phi}{\partial x}, \quad \phi_{xx} = \frac{\partial^2 \phi}{\partial x^2}, \quad \phi_{xy} = \frac{\partial^2 \phi}{\partial x \partial y} \quad \text{etc.}$$

If a perturbation potential,  $\phi$ , is introduced such that

$$\phi = xq_\infty \cos \alpha + yq_\infty \sin \alpha + q_\infty \phi \quad (2.4-3)$$

where the velocity components are given by

$$U = \phi_x = q_\infty (\cos \alpha + \phi_x) \quad (2.4-4a)$$

$$V = \phi_y = q_\infty (\sin \alpha + \phi_y) \quad (2.4-4b)$$

then the governing equation for the perturbation potential becomes

$$(a^2 - U^2)\phi_{xx} - 2UV\phi_{xy} + (a^2 - V^2)\phi_{yy} = 0 \quad (2.4-5)$$

with

$$a^2 = a_\infty^2 - \left[ \frac{\gamma-1}{2} \right] \left[ U^2 + V^2 - q_\infty^2 \right] \quad (2.4-6)$$

The pressure coefficient at any point is given by

$$C_p = \frac{p - P_\infty}{1/2 \rho_\infty U_\infty^2} = \frac{2}{\gamma M_\infty^2} \left\{ \left[ 1 + \frac{\gamma-1}{2} M_\infty^2 \left( 1 - \frac{U^2 + V^2}{q_\infty^2} \right) \right]^{\frac{\gamma}{\gamma-1}} - 1 \right\} \quad (2.4-7)$$

The appropriate boundary conditions are, at the airfoil surface

$$\left( \frac{dy}{dx} \right)_{\text{body}} = \left( \frac{V}{U} \right)_{\text{body}} \quad (2.4-8)$$

and at infinity (Refs. 2.4-2, 2.4-7 and 2.4-8)

$$\phi = \frac{-\Gamma}{2\pi} \text{Tan}^{-1} (\beta \text{Tan}(\theta - \alpha)) \quad (2.4-9)$$

Where  $\theta$  is the polar angle and  $\Gamma$  is the circulation determined by the change in potential across the Kutta-Joukowski cut at the trailing edge of the airfoil, i.e.,

$$\Gamma = (\phi_{y=0^+} - \phi_{y=0^-})_{\text{Trailing Edge}} \quad (2.4-10)$$

#### 2.4.1.2 Numerical Analysis

##### Coordinate Stretching

In the present problem an infinity boundary condition must be applied at the edge of the computational grid. It is convenient to transform or stretch the original  $x,y$  plane to some  $\xi,\eta$  plane so that the edges of the  $\xi,\eta$  grid correspond to infinity. In this way, Equation (2.4-9) can be applied directly. If this is not done, a far-field condition (Ref. 2.4-8) can be used at the grid edges instead.

After investigating several possibilities, the coordinate stretching represented in Figure 2.4-1 was selected. Here  $x,y$  is the physical plane and  $\xi,\eta$  represents the computational plane, and each is subdivided into three regions. The stretching is symmetrical about the origin and it is given by

$$x = x_4 + A_2 \text{Tan} \left[ \frac{\pi}{2} (\xi - \xi_4) \right] + A_3 \text{Tan} \left[ \frac{\pi}{2} (\xi - \xi_4)^3 \right] \quad (2.4-11)$$

in region III and by

$$x = \xi(a + b \xi^2) \quad (2.4-12)$$

in region II. The constants  $a$  and  $b$  are determined by the requirements

$$x = x_4 \text{ at } \xi = \xi_4 \quad (2.4-13a)$$

and

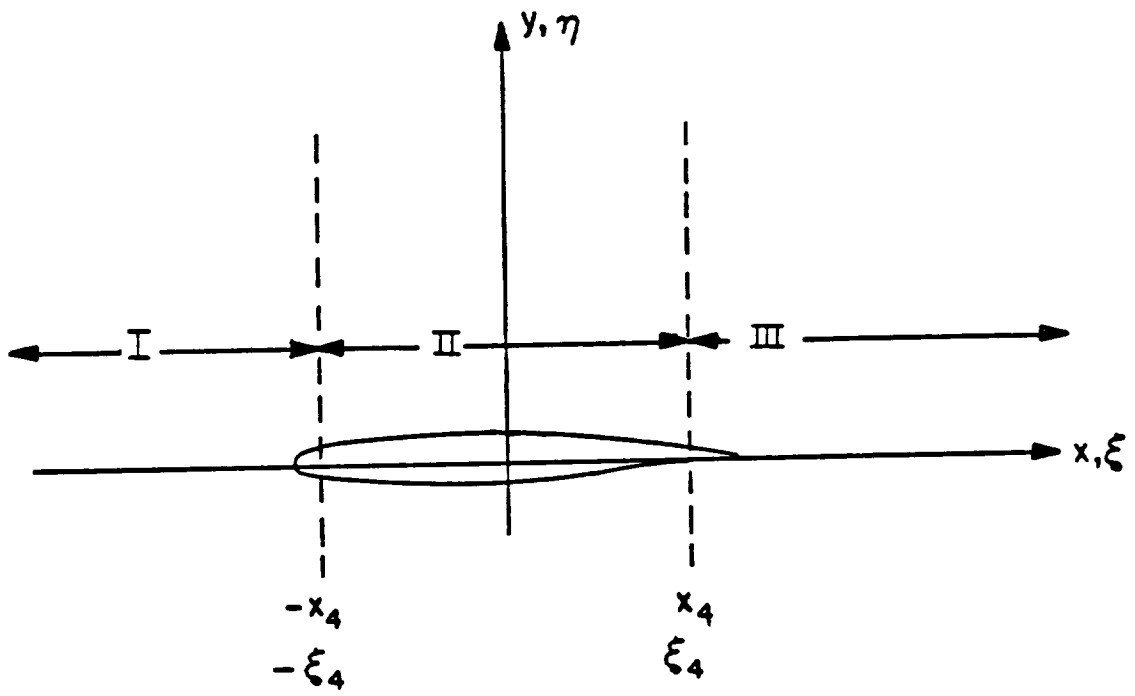


Figure 2.4-1. Flow Field Subdivision for Coordinate Stretching

$$\frac{dx}{d\xi} = \frac{\pi A_2}{2} \text{ at } \xi = \xi_4 \quad (2.4-13b)$$

The constant  $A_2$  controls the grid spacing in the vicinity of  $x_4$ , usually near the leading and trailing edges; while  $A_3$  determines the physical location of the grid line adjacent to the grid edge.

In the  $y$ -direction the stretching relationship is given by

$$y = A_1 \tan\left(\frac{\pi}{2} \eta\right) \quad (2.4-14)$$

where  $A_1$  controls the grid size near the airfoil.

Notice that the stretchings map the infinite physical  $x, y$  plane

$$-\infty \leq x \leq \infty$$

$$-\infty \leq y \leq \infty$$

into the finite computational plane

$$-(1 + \xi_4) \leq \xi \leq 1 + \xi_4$$

$$-1 \leq \eta \leq 1 \quad (2.4-15)$$

where  $\xi_4$  determines the amount of the computational plane confined to the vicinity of the airfoil. A typical grid system is shown in Figure 2.4-2.

When selecting a stretching, care must be taken to ensure that the stretching does not force a physically unrealistic or abnormal behavior on the solution. For example, analysis of a compressible doublet indicates that  $\phi_y$  and  $\phi_x$  should decay as

$$\begin{aligned} \phi_y &\sim y^{-3} \text{ as } y \rightarrow \infty \\ \phi_x &\sim x^{-3} \text{ as } x \rightarrow \infty \end{aligned}$$



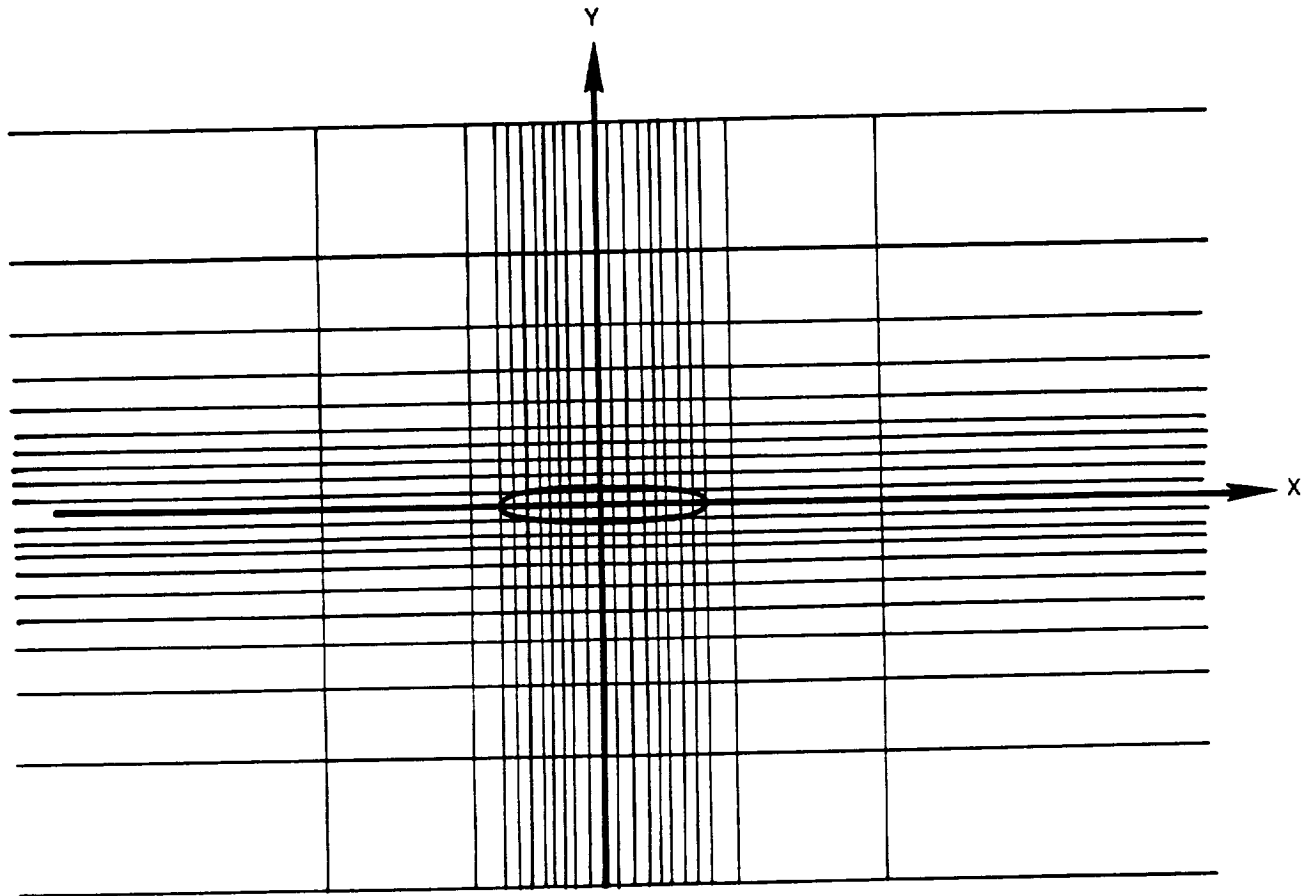


Figure 2.3-2. Typical Grid System (25 x 25)

$$\begin{aligned}\phi_x &\sim x^{-2} \text{ as } y \rightarrow \infty \\ &\sim y^{-2} \text{ as } y \rightarrow \infty\end{aligned}\tag{2.4-16}$$

In the present  $\xi, \eta$  system, this type of decay would require that

$$\begin{aligned}\phi_\eta &\sim y^{-1} \text{ as } y \rightarrow \infty \\ \phi_\xi &\sim \text{constant as } x \rightarrow \infty\end{aligned}\tag{2.4-17}$$

and this type of behavior is quite permissible. However, with some other stretchings such as  $\eta = \tanh(y)$ , the decay indicated by Equation (2.4-16) would require that

$$\phi_\eta \sim \infty \text{ as } y \rightarrow \infty, \eta \rightarrow 1\tag{2.4-18}$$

In practice, it usually happens that the derivatives in the computational plane remain finite, even with the exponential stretchings; but this precludes the correct asymptotic decay, and the effect may produce stable, convergent, wrong solutions.

Now as a result of the introduction of coordinate stretching, the governing equations should be written in terms of the independent computational variables  $\xi$  and  $\eta$ . By defining

$$f = \frac{d\xi}{dx}, \quad g = \frac{d\eta}{dy}\tag{2.4-19}$$

Equations (2.4-4) and (2.4-5) become

$$(a^2 - U^2)f(f\phi_\xi)_\xi - 2UVfg\phi_{\xi\eta} + (a^2 - V^2)g(g\phi_\eta)_\eta = 0\tag{2.4-20}$$

$$U = q_\infty(\cos\alpha + f\phi_\xi)$$

$$V = q_\infty(\sin\alpha + \phi_\eta)\tag{2.4-21}$$

### Finite Difference Scheme

As indicated earlier, at supersonic points in order to avoid difficulties associated with nonalignment of the coordinates and the flow field a rotated finite difference scheme is used in the present problem. In this approach (Refs. 2.4-4 and 2.4-5) the principal part of the governing differential equation can be written in coordinates parallel and perpendicular to the local velocity vector, S and N, respectively, as

$$\left(1 - \frac{q^2}{a^2}\right) \phi_{SS} + \phi_{NN} = 0 \quad (2.4-22)$$

where

$$\begin{aligned} \phi_{SS} &= \frac{1}{q^2} \left[ U^2 f(f\phi_\xi)_\xi + 2UVfg\phi_{\xi\eta} + V^2 g(g\phi_\eta)_\eta \right] \\ \phi_{NN} &= \frac{1}{q^2} \left[ V^2 f(f\phi_\xi)_\xi + 2UVfg\phi_{\xi\eta} + U^2 g(g\phi_\eta)_\eta \right] \end{aligned} \quad (2.4-23)$$

Notice that substitution of Equation (2.4-23) into (2.4-22) yields the governing equation, Equation (2.4-20), and thus, as pointed out by South (Ref. 2.4-5), Equation (2.4-22) is simply a rearrangement of terms which exhibits the basic features of a local rotation to the streamline direction.

The basic concept (Refs. 2.4-4 and 2.4-5) of the rotated scheme is to use, at supersonic points, first-order, upwind differencing in both the  $\xi$  and  $\eta$  directions for all contributions to  $\phi_{SS}$  and central differencing for all contributions to  $\phi_{NN}$ . In this manner the correct zone of dependence at supersonic points is built into the finite difference scheme. At subsonic points, the normal procedure of using central differences for all derivatives directly in Equation (2.4-20) is used. Consequently, the scheme is second-order accurate at subsonic points and first-order accurate at supersonic points. The resulting finite difference equations are solved iteratively by using column relaxation sweeping from upstream to

downstream. In such a procedure the values obtained during an iterative sweep can be thought of as new values,  $\phi_{ij}^+$ ; while those obtained during the previous sweep can be considered old values,  $\phi_{ij}$ . In this manner, the change from one iteration to the next can be viewed as that occurring during some artificial time step,  $\Delta t$ ; artificial time derivatives such as  $\phi_t$ ,  $\phi_{\xi t}$  and  $\phi_{\eta t}$  can be considered. Of course, as the relaxation process converges, these terms become negligible.

Various investigators have used different finite difference formulas to represent the derivatives in Equation (2.4-23). For example, Jameson (Ref. 2.4-4) and South and Jameson (Ref. 2.4-5) in their finite difference formulation used a mixture of old and new values. In this way, terms such as  $\phi_{St}$  and  $\phi_{Nt}$  were implicitly added to the problem and used to control numerical stability. Since it was determined by Jameson (Ref. 2.4-4) that additional  $\phi_{St}$  needed to be explicitly added to the problem in order to ensure stability, it was decided in the present case to use all old values in the difference expressions for  $\phi_{SS}$  and to add  $\phi_{St}$  explicitly. For  $\phi_{NN}$ , appropriate combinations of old and new values were used, causing  $\phi_{Nt}$  to occur implicitly. It should be emphasized again that  $\phi_{Nt}$  and  $\phi_{St}$  go to zero as the solution converges, and hence do not affect the final result.

Thus the finite difference expressions used in the present formulation are:

For contributions to  $\phi_{NN}$  when  $q^2 > a^2$

$$(f\phi_{\xi})_{\xi} = \frac{1}{\Delta\xi^2} \left\{ f_{i+1/2}(\phi_{i+1,j} - \phi_{ij}) - f_{i-1/2}(\phi_{ij}^+ - \phi_{i-1,j}^+) \right\} \quad (2.4-24a)$$

$$\phi_{\xi\eta} = \frac{1}{4\Delta\xi\Delta\eta} \left\{ \phi_{i-1,j}^+ - \phi_{i-1,j+1}^+ - \phi_{i+1,j-1} + \phi_{i+1,j+1} \right\} \quad (2.4-24b)$$

$$(g\phi_{\eta})_{\eta} = \frac{1}{\Delta\eta^2} \left\{ g_{j+1/2}(\phi_{i,j+1}^+ - \phi_{ij}^+) - g_{j-1/2}(\phi_{ij}^+ - \phi_{i,j-1}^+) \right\} \quad (2.4-24c)$$

For contributions to  $\phi_{SS}$  when  $q^2 > a^2$  and  $V > 0$

$$(f\phi_\xi)_\xi = \frac{1}{\Delta\xi^2} \{f_{i-1/2}(\phi_{ij} - \phi_{i-1,j}) - f_{i-3/2}(\phi_{i-1,j} - \phi_{i-2,j})\} \quad (2.4-25a)$$

$$\phi_{\xi\eta} = \frac{1}{\Delta\xi\Delta\eta} \{\phi_{ij} - \phi_{i-1,j} - \phi_{i,j-1} + \phi_{i-1,j-1}\} \quad (2.4-25b)$$

$$(g\phi_\eta)_\eta = \frac{1}{\Delta\eta^2} \{g_{j-1/2}(\phi_{ij} - \phi_{i,j-1}) - g_{j-3/2}(\phi_{i,j-1} - \phi_{i,j-2})\} \quad (2.4-25c)$$

where to create stability there is added to the basic equation

$$\begin{aligned} \frac{-\epsilon\Delta t}{\Delta\xi} f\phi_{St} &= \frac{-\epsilon f\Delta t}{\Delta\xi} \left[ \frac{Uf}{q} \phi_{\xi t} + \frac{Vg}{q} \phi_{\eta t} \right] \\ &= \frac{-\epsilon f}{\Delta\xi} \left[ \frac{Uf_{i-1/2}}{\Delta\xi q} (\phi_{ij}^+ - \phi_{ij} - \phi_{i,j-1}^+ + \phi_{i,j-1}) \right. \\ &\quad \left. + \frac{V}{q} \frac{g_{j-1/2}}{\Delta\eta} (\phi_{ij}^+ - \phi_{ij} - \phi_{i-1,j}^+ + \phi_{i-1,j}) \right] \end{aligned} \quad (2.4-25d)$$

For contributions to  $\phi_{SS}$  when  $q^2 > a^2$  and  $V < 0$

$$(f\phi_\xi)_\xi = \frac{1}{\Delta\xi^2} \{f_{i-1/2}(\phi_{ij} - \phi_{i-1,j}) - f_{i-3/2}(\phi_{i-1,j} - \phi_{i-2,j})\} \quad (2.4-26a)$$

$$\phi_{\xi\eta} = \frac{1}{\Delta\xi\Delta\eta} \{-\phi_{ij} + \phi_{i-1,j} + \phi_{i,j+1} - \phi_{i-1,j+1}\} \quad (2.4-26b)$$

$$(g\phi_\eta)_\eta = \frac{1}{\Delta\eta^2} \{g_{j+1/2}(\phi_{ij} - \phi_{i,j+1}) - g_{j+3/2}(\phi_{i,j+1} - \phi_{i,j+2})\} \quad (2.4-26c)$$

and to create stability there is added

$$\begin{aligned} \frac{-\epsilon f \Delta t}{\Delta \xi} \phi_{St} = & \frac{-\epsilon f}{\Delta \xi} \left[ \frac{U}{q} \frac{f_{i-1/2}}{\Delta \xi} (\phi_{ij}^+ - \phi_{ij} - \phi_{i-1,j}^+ + \phi_{i-1,j}) \right. \\ & \left. + \frac{V}{q} \frac{g_{j+1/2}}{\Delta \eta} (-\phi_{ij}^+ + \phi_{ij} + \phi_{i,j+1}^+ - \phi_{i,j+1}) \right] \end{aligned} \quad (2.4-26d)$$

At points where  $q^2 < a^2$  the expressions are

$$\begin{aligned} (f\phi_\xi)_\xi = & \frac{1}{\Delta \xi^2} \{ f_{i+1/2} \phi_{i+1,j} - (f_{i+1/2} + f_{i-1/2}) \cdot \\ & \left[ \frac{\phi_{ij}^+}{w} + \left(1 - \frac{1}{w}\right) \phi_{ij} \right] + f_{i-1/2} \phi_{i-1,j}^+ \} \end{aligned} \quad (2.4-27a)$$

$$\phi_{\xi\eta} = \frac{1}{4\Delta \xi \Delta \eta} \{ \phi_{i-1,j-1}^+ - \phi_{i-1,j+1}^+ - \phi_{i+1,j-1} - \phi_{i+1,j+1} \} \quad (2.4-27b)$$

$$(g\phi_\eta)_\eta = \frac{1}{\Delta \eta^2} \{ g_{j+1/2} (\phi_{i,j+1}^+ - \phi_{ij}^+) - g_{j-1/2} (\phi_{ij}^+ - \phi_{i,j-1}^+) \} \quad (2.4-27c)$$

where the relaxation factor,  $w$ , has been incorporated into the difference formulas.

In all cases, the velocities  $U$  and  $V$  are represented by central differences using old values. When these expressions, Equations (2.4-24) through (2.4-27), are substituted appropriately into Equation (2.4-22) and (2.4-23) or Equation (2.4-20), the result is a tridiagonal system of equations that can be solved for the values of  $\phi_{ij}^+$  on column  $i$ .

#### Treatment of Boundary Conditions

There are many ways to approximate the flow tangency condition, Equation (2.4-8), at the airfoil boundary. One of the simplest, which is used here, is to generate dummy values of  $\phi$  at mesh points inside the boundary such that the usual difference equations can be solved at points just outside the boundary. The problem is to develop a scheme for providing and updating these dummy values by using the surface flow

tangency condition and neighboring values of  $\phi$  in the mesh, with adequate accuracy and without creating instability.

To accomplish this, we first note that, in the computational coordinates, Equation (2.4-8) becomes:

$$\left(\frac{dy}{dx}\right) = \frac{v_b}{u_b} = \frac{\sin\alpha + g_b\phi_{\eta b}}{\cos\alpha + f_b\phi_{\xi b}} \quad (2.4-28)$$

Next a Taylor series about the dummy point  $(i,j-1)$  is written (see Figure 2.4-3)

$$\phi_{\eta b} = \phi_{\eta_{i,j-1}} + (\eta_b - \eta_{j-1})\phi_{\eta\eta_{i,j-1}} + \dots \quad (2.4-29a)$$

$$\phi_{\xi b} = \phi_{\xi_{i,j-1}} + (\eta_b - \eta_{j-1})\phi_{\xi\eta_{i,j-1}} + \dots \quad (2.4-29b)$$

When these are written in finite difference form using second-order expressions for  $\phi_{\xi}$  and  $\phi_{\eta}$  and at least first-order ones for  $\phi_{\eta\eta}$  and  $\phi_{\xi\eta}$  they become (for the upper surface case)

$$\begin{aligned} \phi_{\eta b} &= \frac{-3\phi_{i,j-1} + 4\phi_{ij} - \phi_{i,j+1}}{2\Delta\eta} \\ &+ (\eta_b - \eta_{j-1}) \left[ \frac{\phi_{i,j-1} - 2\phi_{ij} + \phi_{i,j+1}}{\Delta\eta^2} \right] \end{aligned} \quad (2.4-30a)$$

$$\begin{aligned} \phi_{\xi b} &= \frac{\phi_{i+1,j-1} - \phi_{i-1,j-1}}{2\Delta\xi} \\ &+ (\eta_b - \eta_{j-1}) \left[ \frac{\phi_{i+1,j} - \phi_{i+1,j-1} - \phi_{i-1,j} + \phi_{i-1,j-1}}{2\Delta\xi\Delta\eta} \right] \end{aligned} \quad (2.4-30b)$$

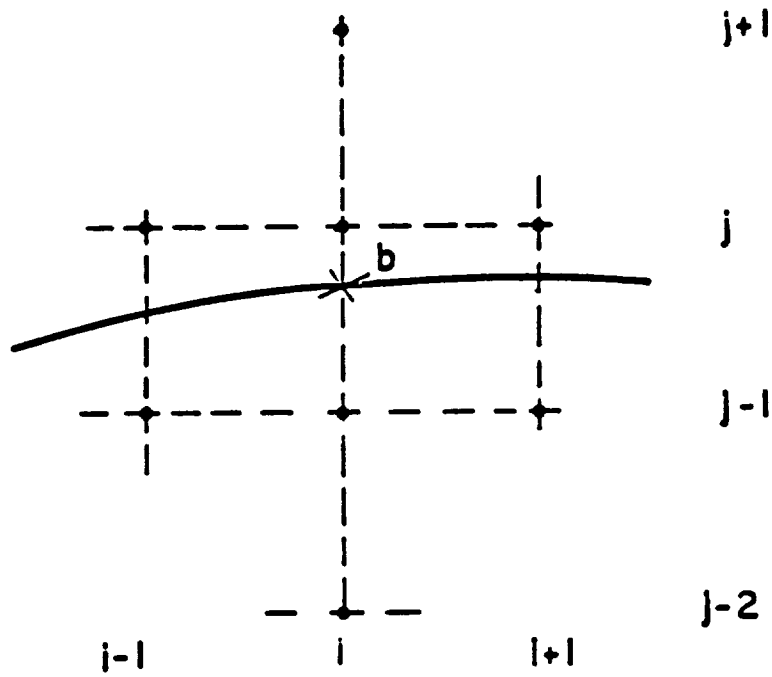


Figure 2.4-3. Relationship Between Airfoil and Grid (Upper Surface)



These expressions can then be substituted into Equation (2.4-28) and the result solved for a sufficiently accurate  $\phi_{i,j-1}$  that is in terms of the neighboring potentials, body slope, and body position.

For those situations where the flow at point (i,j) is supersonic and  $V_{ij} > 0$ , examination of Equation (2.4-25) shows that the difference scheme will require a  $\phi_{i,j-2}$  values as well as  $\phi_{i,j-1}$ . Thus, in all cases a value of  $\phi_{i,j-2}$  is determined by extrapolation as

$$\phi_{i,j-2} = -\phi_{ij} + 2\phi_{i,j-1} \quad (2.4-31)$$

The above procedure for determining the values of the dummy mesh points inside the boundary is performed twice for the relaxation procedure of column i. First, Equation (2.4-28) is used with  $\phi$  values obtained by the previous relaxation sweep in order to obtain old values for the dummy points  $\phi_{i,j-1}$  and  $\phi_{i,j-2}$ . Then, after column i has been relaxed, it is used again with as many current values of  $\phi$  as possible to obtain new values  $\phi_{ij-1}^+$  and  $\phi_{ij-2}^+$ . In this manner, the dummy mesh points will have both old and new values just like regular mesh points, and they can be used directly in the finite difference formulas without special treatment.

A similar procedure is used to satisfy the boundary conditions on the lower surfaces of the air foil.

At the edge of the grid the infinity boundary condition given by Equation (2.4-9) must be used. In the stretched  $\xi, \eta$  system the polar angle at the edge of the grid will have the values shown in Figure 2.2-4. Thus, the boundary values will be:

$$\text{On line AB, } \theta = 0, \phi = \frac{-\Gamma}{2\pi} (-\tan^{-1} [\beta \tan \alpha]) \quad (2.4-32a)$$

$$\text{On line BC, } \theta = \frac{\pi}{2}, \phi = \frac{-\Gamma}{2\pi} \tan^{-1} (\beta \cot \alpha) \quad (2.4-32b)$$

$$\text{On line CD, } \theta = \pi, \phi = \frac{-\Gamma}{2\pi} [\pi - \tan^{-1} (\beta \tan \alpha)] \quad (2.4-32c)$$

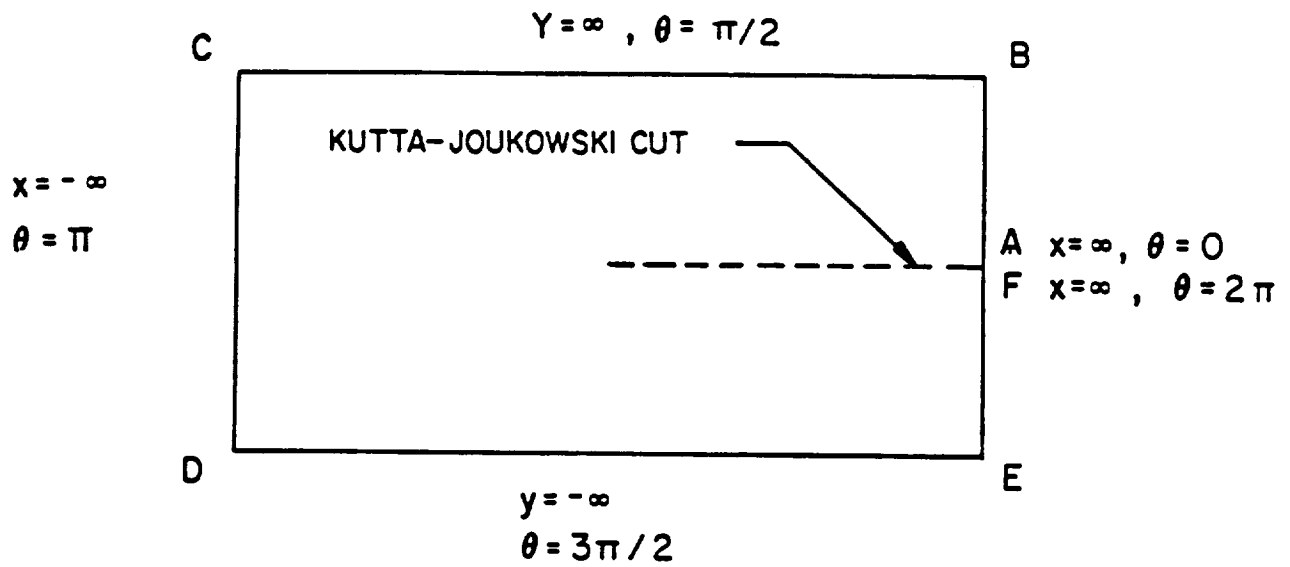


Figure 2.4-4. Computational Grid - Infinity Points

$$\text{On line DE, } \theta = \frac{3\pi}{2}, \phi = \frac{-\Gamma}{2\pi} [\pi - \text{Tan}^{-1} (\beta \cot \alpha)] \quad (2.4-32d)$$

$$\text{On line EF, } \theta = 2\pi, \phi = \frac{-\Gamma}{2\pi} [2\pi - \text{Tan}^{-1} (\beta \tan \alpha)] \quad (2.4-32e)$$

Boundary values must also be assigned to the singular corner points B, C, D, and E since they will be used in the finite difference formula for  $\phi_{\xi\eta}$ . The choice of these values is somewhat arbitrary since they depend upon how the corner points are approached. Fortunately, numerical studies indicate that the solution is insensitive to the values selected; and, thus, the following have been used:

$$\text{Point B, } \phi = \frac{-\Gamma}{2\pi} \text{Tan}^{-1} \left[ \frac{A_1}{A_2} \frac{\Delta\xi}{\Delta\eta} \right] \quad (2.4-33a)$$

$$\text{Point C, } \phi = \frac{-\Gamma}{2\pi} \left\{ \pi - \text{Tan}^{-1} \left[ \frac{A_1}{A_2} \frac{\Delta\xi}{\Delta\eta} \right] \right\} \quad (2.4-33b)$$

$$\text{Point D, } \phi = \frac{-\Gamma}{2\pi} \left\{ \pi + \text{Tan}^{-1} \left[ \frac{A_1}{A_2} \frac{\Delta\xi}{\Delta\eta} \right] \right\} \quad (2.4-33c)$$

$$\text{Point E, } \phi = \frac{-\Gamma}{2\pi} \left\{ 2\pi - \text{Tan}^{-1} \left[ \frac{A_1}{A_2} \frac{\Delta\xi}{\Delta\eta} \right] \right\} \quad (2.4-33d)$$

These assume that the corner points are approached along a diagonal (in the  $\xi$ - $\eta$  plane) connecting the interior point just inside the corner and the corner points

Using the above finite difference formulas and boundary conditions, the governing differential equation can be solved iteratively by column relaxation. For each iteration, the sweep pattern is from upstream to downstream in the region in front of the airfoil, then in the region above the airfoil, then below the airfoil, and finally in the portion downstream of the trailing edge. The latter region contains the Kutta-Joukowski cut,

and the equations are appropriately differenced to give the proper jump in the potential.

### 2.4.1.3 Numerical Stability

#### Supersonic Points

The incorporation into the finite difference formulas of both old,  $\phi$ , and new,  $\phi^+$ , values and the explicit inclusion of  $\phi_{St}$  at supersonic points introduces time-like derivatives into the equation being solved. Thus, during the iterative process the actual equation at supersonic points is of the form

$$\gamma\phi_t + \left[ \frac{q^2}{a^2} - 1 \right] \phi_{SS} + 2\alpha\phi_{St} - \phi_{NN} + 2\beta\phi_{Nt} = 0 \quad (2.4-34)$$

Jameson (Ref. 2.4-4) has shown that for numerical stability the coefficients must be such that

$$\gamma = 0 \quad (2.4-35a)$$

$$\alpha^2 > \beta^2 \left[ \frac{q^2}{a^2} - 1 \right] \quad (2.4-35b)$$

Since the  $\phi_{St}$  is included explicitly [see Equations (2.4-25d) and (2.4-26d)], that  $\alpha$  is known, and the  $\beta$  can be determined from the  $\phi_{NN}$  term.

By adding and subtracting old values of  $\phi$  to get  $\phi_{NN}$  in terms of old values, the time-like derivatives can be isolated. Thus

$$\phi_{NN} - 2\beta\phi_{Nt} = \phi_{NN}$$

$$\begin{aligned}
& + \frac{v^2 f}{q^2} \left[ \frac{-f_{i-1/2} (\phi_{ij}^+ - \phi_{ij})}{\Delta\xi^2} + f_{i-1/2} \frac{(\phi_{i-1,j}^+ - \phi_{i-1,j})}{\Delta\xi^2} \right] \\
& - \frac{2UVfg}{q^2} \left[ \frac{\phi_{i-1,j-1}^+ - \phi_{i-1,j-1} - \phi_{i-1,j+1}^+ + \phi_{i-1,j+1}}{4\Delta\xi\Delta\eta} \right] \\
& + \frac{U^2g}{q^2} \left[ g_{i+1/2} \frac{(\phi_{i,j+1}^+ - \phi_{i,j+1} - \phi_{ij}^+ + \phi_{ij})}{\Delta\eta^2} \right. \\
& \left. - g_{j-1/2} \frac{(\phi_{ij}^+ - \phi_{ij} - \phi_{i,j-1}^+ - \phi_{i,j-1})}{\Delta\eta^2} \right] \tag{2.4-36}
\end{aligned}$$

$$= \phi_{NN} - \frac{v^2 f}{q^2} \frac{\Delta t}{\Delta\xi} (f\phi_{\xi t})_{i-1/2,j} + \frac{UVfg}{q^2} \frac{\Delta t}{\Delta\xi} (\phi_{\eta t})_{i-1,j}$$

$$+ \frac{Ug^2}{q^2} \frac{\Delta t}{\Delta\eta} [(g\phi_{\eta t})_{i,j+1/2} - (g\phi_{\eta t})_{i,j-1/2}] \tag{2.4-36}$$

which in the limit of vanishing step size becomes

$$\phi_{NN} - 2\beta\phi_{Nt} = \phi_{NN} + \frac{Vf}{q} \frac{\Delta t}{\Delta\xi} \left[ \frac{-V}{q} f\phi_{\xi t} + \frac{Ug}{q} \phi_{\eta t} \right] \tag{2.4-37}$$

where the term in brackets is  $\phi_{Nt}$ . Hence, from a time-like viewpoint the actual differential equation being solved is

$$\left( 1 - \frac{q^2}{a^2} \right) \phi_{SS} - \frac{\epsilon\Delta t}{\phi\xi} f\phi_{St} + \phi_{NN} + \frac{V}{q} \frac{f}{\Delta\xi} \Delta t \phi_{Nt} = 0 \tag{2.4-38}$$

and so

$$\gamma = 0, \quad \alpha = \frac{\epsilon \Delta t f}{2 \Delta \xi}, \quad \beta = \frac{-V f \Delta t}{q \Delta \xi} \quad (2.4-39)$$

Thus, from Equation (2.4-35), a necessary condition for stability is

$$\epsilon^2 > \frac{V^2}{q^2} \left[ \frac{q^2}{a^2} - 1 \right] \quad (2.4-40)$$

Notice that this requirement means that, for a fixed value for  $\epsilon$ , numerical instability is most likely to occur where the local Mach number is large, say immediately upstream of a shock wave. Numerical experiments have been conducted to verify Equation (2.4-40), and in most cases instability, if present, does originate from the high Mach number region in front of a shock wave. However, the minimum value of  $\epsilon$  predicted by (2.4-40) is usually much smaller than the value actually required in practice. This latter phenomena has also been observed by other investigators using time-like damping, and usually the actual value is only slightly less than  $\epsilon_{\max}$  where

$$\epsilon_{\max}^2 \sim M_{\max}^2 - 1 \quad (2.4-41)$$

### Subsonic Damping

Since at subsonic points Equation (2.4-20) is used while at supersonic points Equation (2.4-22) is solved, there is the possibility that during the initial iterative stages that the time-dependent terms of the two finite difference equation forms do not match near the sonic line. This mismatch possibility can be prevented by adding, to the subsonic finite difference equations, a term

$$\frac{-\epsilon \Delta t f}{\Delta \xi} \phi_{St} = \frac{-\epsilon \Delta t f}{\Delta \xi} \left[ \frac{U f}{q} \phi_{St} + \frac{V}{q} g \phi_{\eta t} \right] \quad (2.4-42)$$

which when  $q$  equals  $a$  causes the supersonic and subsonic difference equations to match exactly.

The question naturally arises, however, of what effect Equation (2.4-42) has upon convergence and stability away from sonic points. By substituting Equations (2.4-27) into Equation (2.4-20) and isolating the time-like terms, it can be determined that the actual equation being solved at subsonic points is

$$\begin{aligned}
 & \left(1 - \frac{U^2}{a^2}\right) f (f\phi_\xi)_\xi - \frac{2UV}{a^2} fg\phi_{\xi\eta} + \left(1 - \frac{V^2}{a^2}\right) g (g\phi_\eta)_\eta \\
 & + \left(1 - \frac{U^2}{a^2}\right) \frac{f^2\Delta t}{\Delta\xi^2} \left(\frac{w-2}{w}\right) \phi_t - \left(1 - \frac{U^2}{a^2}\right) \frac{f^2}{\Delta\xi} \Delta t\phi_{\xi t} \\
 & + \frac{UVfg}{a^2} \frac{\Delta t}{\Delta\xi} \phi_{\eta t} - \epsilon \frac{\Delta t f}{\Delta\xi} \left[\frac{Uf}{q} \phi_{\xi t} + \frac{V}{q} g\phi_{\eta t}\right] = 0
 \end{aligned} \tag{2.4-43}$$

Far away from the supersonic zone and the airfoil,  $V$  will normally be very small and the effective equation will be

$$\begin{aligned}
 & \left(1 - \frac{U^2}{a^2}\right) f (f\phi_\xi)_\xi + g (g\phi_\eta)_\eta + \left(1 - \frac{U^2}{a^2}\right) \frac{f^2}{\Delta\xi^2} \Delta t \left(\frac{w-2}{w}\right) \phi_t \\
 & - \left[\left(1 - \frac{U^2}{a^2}\right) \frac{f^2\Delta t}{\Delta\xi} + \frac{\epsilon\Delta t f}{\Delta\xi} \frac{Uf}{q}\right] \phi_{\xi t} = 0
 \end{aligned} \tag{2.4-44}$$

In order to see the effect of the  $\epsilon\phi_{\xi t}$  term on the solution behavior of this equation, assume

$$\begin{aligned}
 f & \approx g \\
 \Delta t & \Delta\xi \\
 U & q
 \end{aligned} \tag{2.4-45}$$

and let

$$A = 1 - \frac{U^2}{a^2} \approx \text{constant} \quad (2.4-46)$$

Then Equation (2.4-44) can be modeled as

$$A\phi_{\xi\xi} + \phi_{\eta\eta} = \frac{A}{\Delta\xi} \left( \frac{2-w}{w} \right) \phi_t + A\phi_{\xi t} + \epsilon\phi_{\xi t} \quad (2.4-47)$$

The convergence behavior of this equation can be studied following the approach of Garabedian (Ref. 2.4-4). First, transform from  $(\xi, \eta, t)$  to  $(\xi, \eta, s)$  by

$$s = t + \left( \frac{A + \epsilon}{2A} \right) \xi \quad (2.4-48)$$

and assume solution of the form  $\phi = U(\xi, \eta)W(s)$ . Thus, in general, the solution to Equation (2.4-47) is

$$\phi = U_0(\xi, \eta) + \sum_{m=1}^{\infty} W_m U_m \quad (2.4-49)$$

where  $W_m$  and  $U_m$  are eigenfunctions and

$$W_m = C_m e^{p_{m1}s} + D_m e^{p_{m2}s} \quad (2.4-50)$$

and

$$p_{m1,2} = \frac{-\beta + [\beta^2 - 4\alpha k m^2]^{1/2}}{2\alpha} \quad (2.4-51)$$

and  $C_m$  and  $D_m$  are the Fourier coefficients.



Obviously, the rate of decay of the time terms is determined by the real part of  $P_{11}$ , and the optimum convergence rate will occur when the relaxation factor,  $w$ , is such that the radical in Equation (2.4-51) is zero. When it is zero or purely imaginary, the rate of convergence will be determined by

$$p = \frac{-\beta}{2\alpha} = \frac{-2A^2 \left( \frac{2-w}{w} \right)}{\Delta\xi (A + \epsilon)^2} \quad (2.4-52)$$

For a fixed relaxation factor, as the damping coefficient,  $\epsilon$ , increases, the radical in Equation (2.4-51) eventually goes negative; and the error,  $E$ , due to  $N$  relaxation iterations will be of the order

$$E = O(e^{-pN\Delta t}) \quad (2.4-53)$$

Since  $p$  decreases as  $(A + \epsilon)^{-2}$ , an increase in  $\epsilon$  from zero to one can cause an order of magnitude increase in the number of iterations required for convergence.

From this analysis it can be concluded that the addition of  $\epsilon\phi_{St}$  at subsonic points significantly decreases the rate of convergence and large values of  $\epsilon$  should be avoided.

In addition, South has shown by a linear stability analysis that for the model equation

$$\phi_{xx} + \phi_{yy} - \epsilon U\phi_{xt} - \epsilon V\phi_{yt} = 0 \quad (2.4-54)$$

that the inclusion of explicit time-like damping in addition to that implicitly obtained by the finite difference forms of  $\phi_{xx}$  and  $\phi_{yy}$  can, in a stretched coordinate system, be destabilizing and may cause numerical instability, particularly in regions of hard stretching where  $V$  is large.

Based upon these two analyses, both of which have been verified numerically, it is believed the specific addition of  $\epsilon\phi_{St}$  at subsonic points should be avoided and that the mismatch in time-like terms near

sonic points should be prevented by changing the damping coefficient at supersonic points to  $[M^2 - 1]^{1/2}\epsilon$ .

#### 2.4.1.4 Numerical Studies

In actually carrying out the numerical solution, the parameters  $A_1$ ,  $A_2$ , and  $A_3$  should be selected so that the grid spacing,  $\Delta x$  and  $\Delta y$ , is small near the airfoil, and  $x_4$  should be near the trailing edge. The question is how small and how close? See Figure 2.4-5.

To answer these questions a series of numerical experiments was conducted and it was determined that accurate results for lift and surface pressures could be obtained when

$$\frac{\Delta x}{c} \approx \frac{\Delta y}{c} \approx \frac{2r_{LE}}{c} \quad (2.4-55)$$

in the vicinity of the leading edge. (In the present computer formulation the chord  $c$  is always unity. The airfoil leading edge is at  $x = -0.5$  and the trailing edge at  $x = +0.5$ .) No significant changes in lift or surface pressures were observed for smaller step sizes as long as  $\Delta x$  was about the same magnitude as  $\Delta y$  near the leading edge. However, when  $\Delta y$  was three times  $\Delta x$  near the leading edge good accuracy was not achieved until

$$\frac{\Delta x}{c} \sim \frac{r_{LE}}{c}, \Delta y \sim 3\Delta x \quad (2.4-56)$$

and considerable sensitivity to the value of  $\Delta x$  was observed. In addition, the pressure coefficient near the nose exhibited slight oscillatory behavior.

Considerable sensitivity was also observed for the value chosen for  $x_4$ , which essentially determines the first and last grid points on the airfoil. For a grid spacing given by Equation (2.4-55),  $x_4$  values of 0.50 and 0.49 yielded pressure coefficient results that differed by as much as 10 percent, with the results associated with the 0.5 value being worse. In addition, sensitivity to grid spacing was observed in the 0.5 case but was minimal in the 0.49 case. Further investigation showed that, if the

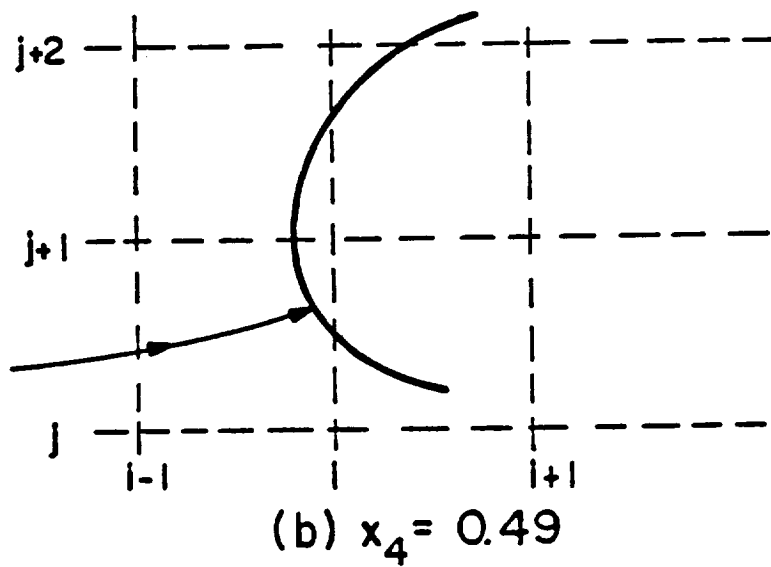
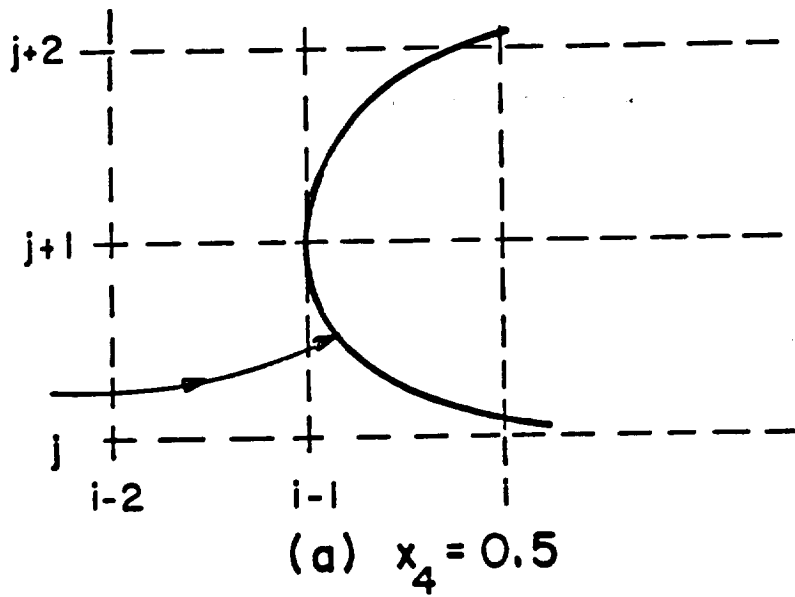


Figure 2.4-5. Grid Pattern Versus  $x_4$

spacing near the leading edge were reduced to  $r/c$  either by using more points or by coordinate stretching, good results insensitive to the  $x_4$  choice could be obtained even for  $x_4$  values near 0.5. The cause of this behavior is not completely understood.

Based upon these results and those previously discussed concerning numerical stability, it is believed that accurate solutions can be obtained when

$$x_4 \approx 0.49 \leftrightarrow 0.495$$

$$\frac{\Delta x}{c} \approx \frac{\Delta y}{c} \approx \frac{2r_{LE}}{c} \quad \text{near the leading edge} \quad (2.4-57)$$

In addition, considering the destabilizing effects of hard stretching, it is recommended that whenever possible  $\Delta x$  be kept relatively constant between the leading and trailing edges of the airfoil.

#### 2.4.1.5 Viscous Boundary Layer and Wave Drag Correction

Experimental evidence indicates that viscous boundary layer effects are very important in transonic flow. For example, an aft-cambered airfoil inviscidly designed to have a lift coefficient of 0.6 may actually develop 25 to 50 percent less lift. This loss in lift is due not only to the existence of a boundary layer displacement surface but also to such factors as wake curvature and vertical pressure gradients in the trailing edge region. To prevent such discrepancies, the effect of the viscous boundary layer should be included in both the analysis and design portion of any numerical method.

In the present program, the approach is to assume that the inviscid streamlines follow a displacement surface having ordinates and slopes different from the actual airfoil. The effect of the fact that the streamlines do not follow a displacement surface in the vicinity of the trailing edge and that they are influenced by wake curvature is assumed to be either secondary or capable of being handled empirically. In the design case, the approach is to treat the airfoil determined by the inverse method

as the displacement surface and to subtract from it the displacement thickness determined by a boundary layer computation. The result is considered to be the actual airfoil ordinates. For the analysis case, the approach is to calculate a boundary layer displacement surface (i.e., airfoil ordinate plus  $\delta^*$ ) using underrelaxation. The inviscid flow field is then solved, and the displacement surface is updated every ten iterative cycles.

Obviously, the boundary layer scheme must be reliable, reasonably accurate, and computationally very efficient. After extensive investigation, the Nash-Macdonald method (Ref. 2.4-10) together with certain smoothing operations, was selected for incorporation into the present program. In addition, the displacement thickness at the trailing edge grid point was always determined by linear extrapolation from the previous two upstream grid point values. As a result, the basic approach used in the present program is similar to that of Reference 2.4-11.

To update the displacement surface, the momentum integral equation

$$\frac{d\theta}{ds} + (H + 2-M^2) \frac{\theta}{q} \frac{dq}{ds} = \tau \quad (2.4-58)$$

is solved for the momentum thickness  $\theta$  using the formulas of Nash and Macdonald for skin friction,  $\tau$ , and the shape factor  $H = \delta^*/\theta$ . This computation is performed on the same grid spacing as the corresponding inviscid solution. The resultant displacement thickness is then smoothed everywhere and extrapolated to obtain the thickness at the trailing edge point. The smoothing is performed twice on grids having IMAX less than 55 and four times on grids with IMAX greater than 55.

This smoothing and extrapolation process appears to have two consequences. First, it reduces the rapid variations in the solution that sometimes occur in regions with high pressure gradients. Second, based on comparisons with experiments, the Nash-Macdonald method with smoothing and extrapolation seems to yield a trailing edge behavior that is correct with respect to the effect of the boundary layer on pressure distribution and lift. Admittedly, this behavior is fortuitous and some sensitivity to grid

spacing has been detected. However, it should serve as a reasonable engineering model until a more complete, rational, trailing-edge theory is available. At that time such a theory could be easily incorporated into the present program.

If a case with extensive upper surface separation is encountered, the user may need to incorporate the optional trailing edge correction feature of the program. In this correction the boundary layer is solved using a modified pressure distribution that is linear from a point XLSEP, corresponding to separation, to the trailing edge. The base pressure, which determines the pressure gradient in this region, is determined semiempirically. For aft-cambered airfoils, it is automatically selected to be the same as the maximum value of the pressure coefficient encountered on the lower surface of the airfoil. For conventional airfoils, it should be selected by the user based upon experience. (A typical value is 0.6.) For aft-cambered airfoils the resultant modified boundary layer computation is normally applied only to the upper surface and is only used to determine the ordinate and slope of the displacement surface at XLSEP. The slope is then assumed to be constant from XLSEP to the trailing edge and the resultant displacement surface shape determined. Based upon comparisons with experiment, this approach yields reasonably good results and eliminates oscillations in the pressure distribution which can occur due to very small changes in the displacement surface slopes. Thus the method is a combination of the approaches used in References 2.4-11 and 2.4-12.

For conventional airfoils, the modified boundary layer computation is used to determine when the slope of the displacement surface becomes zero. From that point to the trailing edge the slope is then held constant. This approach is based on the concept that the streamlines from both the upper and lower surfaces should enter the wake almost parallel. Thus, for conventional airfoils, if the trailing edge correction is used, it should be applied to both surfaces. Since in most cases this correction is not needed on conventional airfoils, its extensive use is not recommended until it has been verified by experiment.

In both the normal and corrected uses, separation is assumed to occur when  $(-\theta/q \, dq/ds)$  is greater than 0.004.

Now one of the difficulties associated with using a cartesian grid is that such a grid does not place a large number of computational points near the leading and trailing edges. Thus, the wave drag coefficient, which is determined by integration of the pressure distribution, has an inherent error associated with grid size, grid spacing, and the magnitude of the lift coefficient. Extensive comparisons with experimental data have indicated, however, that accurate estimates of the wave drag can be obtained by applying a suitable correction factor, CDCORR. This correction factor, which is different for each airfoil and computational grid, can be determined as a function lift from a series of calculations at different angles of attack at subcritical speeds, where the wave drag should be zero.

For each subcritical calculation, using a CDCORR of zero, determine the axial and normal coefficients using

$$\begin{aligned} CN &= CL \cos \alpha + CDWAVE \sin \alpha \\ CA &= -CL \sin \alpha + CDWAVE \cos \alpha \end{aligned}$$

where CL is the lift coefficient determined from integration of the  $C_p$  distribution. Then the true CDCORR corresponding to CL can be computed from

$$CDCORR = CA + CN \sin \alpha / \cos \alpha$$

This value can then be used in supercritical runs having the same CL.

In some cases, it may be more convenient to compute the supercritical flows using a CDCORR of zero and to apply the correction later. In those cases, the following procedure can be used to determine the drag. First, compute the axial and normal coefficients using

$$\begin{aligned} CN &= CL \cos \alpha + CDWAVE \sin \alpha \\ CA &= -CL \sin \alpha + CDWAVE \cos \alpha \end{aligned} \tag{2.4-59}$$

Then correct the axial coefficient for the appropriate lift and grid by

$$C_A = C_A - CDCORR$$

and recompute CDWAVE by

$$CDWAVE = C_N \sin \alpha + C_A \cos \alpha$$

The total drag coefficient is then given by

$$C_D = CDWAVE + CDF \quad (2.4-60)$$

In all cases the drag due to skin friction and to changes resulting from the displacement surface shape is computed using the Squire-Young formula. While this formula is not exactly theoretically correct, it has been found to yield very accurate predictions.

#### 2.1.4.6 Massive Separation Model

In the design and analysis of high performance airfoils, aerodynamicists would like not only to compute cruise behavior, but also to predict airfoil pressure distributions at high lift, high angle-of-attack conditions. Since such situations are characterized by large regions of separated flow and are dominated by strong viscous interaction effects, inviscid methods are not applicable. Furthermore, subsonic-transonic analysis and design computer codes (Refs. 2.4-13 and 2.4-14) typically only include the effects of weak viscous interaction and fail to give acceptable answers whenever the length of the separated zone exceeds a few percent of the airfoil chord.

However Barnwell (Ref. 2.4-15) recently demonstrated that the direct-inverse technique could be successfully applied to the low-speed high-lift case. By specifying the separation point, he was able to obtain excellent agreement with experimental data by solving the linear equation with direct nonlinear boundary conditions (airfoil ordinates specified) upstream of separation and inverse boundary conditions (pressure specified) downstream of separation. Thus, the question arose - could similar results be



obtained using the full potential equation coupled with viscous interaction and letting the separation point and pressure level be determined as part of the solution?

This section describes the flow model that can be used to determine the flow field about a low-speed single-element airfoil at high angle-of-attack, high lift conditions with massive boundary layer separation (Ref. 2.4-16).

As indicated previously, the present approach is based upon the direct-inverse method as developed in the TRANDES program and its ability to use either the displacement surface (airfoil ordinate plus displacement thickness) or pressure as the airfoil boundary condition. For the low speed high angle-of-attack case, the airfoil lower surface only experiences weak viscous interaction. The lower surface boundary layer, however, frequently has a long laminar run before transitioning to fully turbulent flow. Thus, the model needs to include an initial laminar boundary layer in its viscous interaction section. On the upper surface the boundary layer is also initially laminar, but it quickly becomes turbulent in character followed by boundary layer separation and a separated zone which can extend over as much as three-fourths of the airfoil surface. Fortunately, however, the separated region at low freestream velocities is characterized by an approximately constant pressure level. Consequently, the low speed massive separation problem has been modeled as shown in Figure 2.4-6.

To obtain the inviscid portion of the flow field, the exact perturbation potential equation is solved iteratively using a rotated finite difference scheme and column relaxation. To include viscous effects, the basic approach is to calculate a boundary layer displacement thickness for weak interaction regions and to use it to correct the location of the displacement surface (i.e., airfoil ordinate plus displacement thickness,  $\delta^*$ ). For the strongly interacting separated zone, the pressure is specified and the location of the displacement surface is computed by integrating the surface-tangency conditions, with the initial conditions specified by the displacement surface slope and ordinate at the

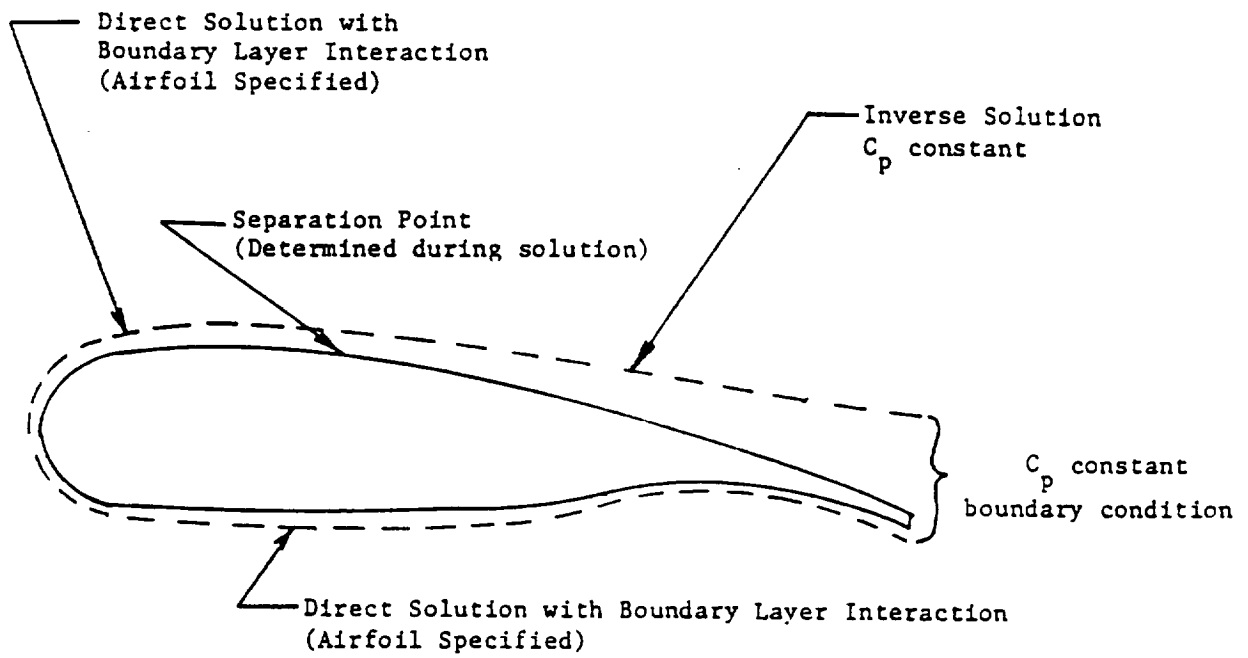


Figure 2.4-6. Problem Formulation

interface between the two regions. At present, the location and slopes of the displacement surface are updated every ten relaxation cycles.

On the lower surface of the airfoil, the flow field is determined using direct boundary conditions (airfoil specified) including the effects of weak viscous interaction. On the upper surface, the flow field is also computed directly with viscous interaction up to the separation point, which is determined as part of the solution. Downstream of separation, inverse boundary conditions are used, and the pressure is assumed to be constant in the separated zone. Studies with this model have shown that the separated-zone pressure, which has to be computed as part of the solution, must be determined by conditions at both the separation point and at the trailing edge and not just on conditions in the vicinity of separation.

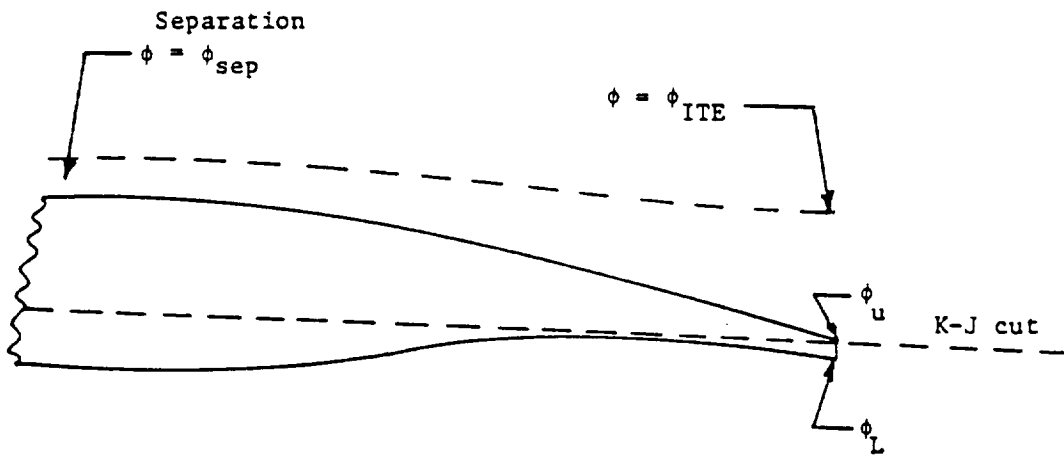
In the present formulation, the pressure coefficient for the constant pressure separated zone is computed by the equation

$$C_{p,sep} = \frac{-2(\phi_{ITE} - \phi_{sep})}{X_{ITE} - X_{sep}} \quad (2.4-61)$$

as illustrated on Figure 2.4-7. Here  $\phi_{ITE}$  and  $\phi_{sep}$  are the perturbation potentials at the trailing edge and the turbulent boundary layer separation point, respectively. Note that Equation (2.4-61) is a small-perturbation approximation for  $C_{p,sep}$ ; and while this form probably introduces some error into the overall problem, its usage has been found to be simple, accurate, and adequate.

In principle, the separated region and wake probably should be accurately modeled with respect to physical phenomena and details. This would typically introduce a large number of computational points to model the wake region. In the present model, however, the wake region contains very few computational points due to the coordinate stretching. Thus, it is treated very simply in that it is assumed to be inviscid with a constant pressure-trailing edge formed by the upper and lower displacement surfaces. As will be discussed later, this simplistic approach yields results which agree well with experimental data. In fact, numerical experiments with the

$$C_{p_{sep}} = -2 \frac{(\phi_{ITE} - \phi_{sep})}{X_{ITE} - X_{sep}}$$



$$\Gamma = \phi_u - \phi_L$$

Figure 2.4-7.  $C_{p_{sep}}$  and  $\Gamma$  Formulation

present model indicate that the results for the pressure distribution and aerodynamic coefficients are primarily dependent upon obtaining accurate predictions for the location of the separation point and the magnitude of the separated pressure. Apparently, the details of the wake region are of secondary importance. Finally, as shown in Figure 2.4-7, the airfoil circulation,  $\Gamma$ , is modeled as the difference in the perturbation potentials at the airfoil trailing edge, i.e.,  $\Gamma = \phi_U - \phi_L$ , and not as  $(\phi_{ITE} - \phi_L)$ .

In the present code the turbulent boundary layer is computed using the Nash-Macdonald method in the same manner as in the original TRANDES program (Ref. 2.4-15). For the laminar portion, the boundary layer is computed using a compressible Thwaites method which is a slightly modified version of a NASA Langley code originally developed by Grumman Aerospace Corporation. These integral methods are efficient and reliable and yield excellent predictions for displacement thickness values. Internally, the transition point is determined from a Granville type correlation based upon the difference between the local momentum-thickness Reynolds number and the value at the laminar instability point combined with the pressure gradient history. Sometimes, particularly on the upper surface at high angles of attack, laminar separation is predicted upstream of the transition point. In these cases, the local momentum-thickness Reynolds number is compared to an empirical correlation in order to determine if the laminar bubble is long or short. If the estimate indicates that the bubble is long, the calculation proceeds, but a warning is printed which indicates that the results are probably in error. If the bubble is short, its length is assumed to be one horizontal  $\Delta x$  grid width (about 3 percent of chord) and the turbulent-flow computation is initiated at the next downstream grid point. As will be demonstrated, this very simple model and approach usually yield adequate results.

The calculation procedure used in TRANSEP is the same iterative successive-column-relaxation scheme used in the basic TRANDES program except that the separation point and separated pressure level are permitted to vary. A convergence history for a typical case is shown in Figure 2.4-8. Initially, some oscillation occurs on each grid, but, as can be seen, the values quickly converge. Normally 400 iterative cycles are

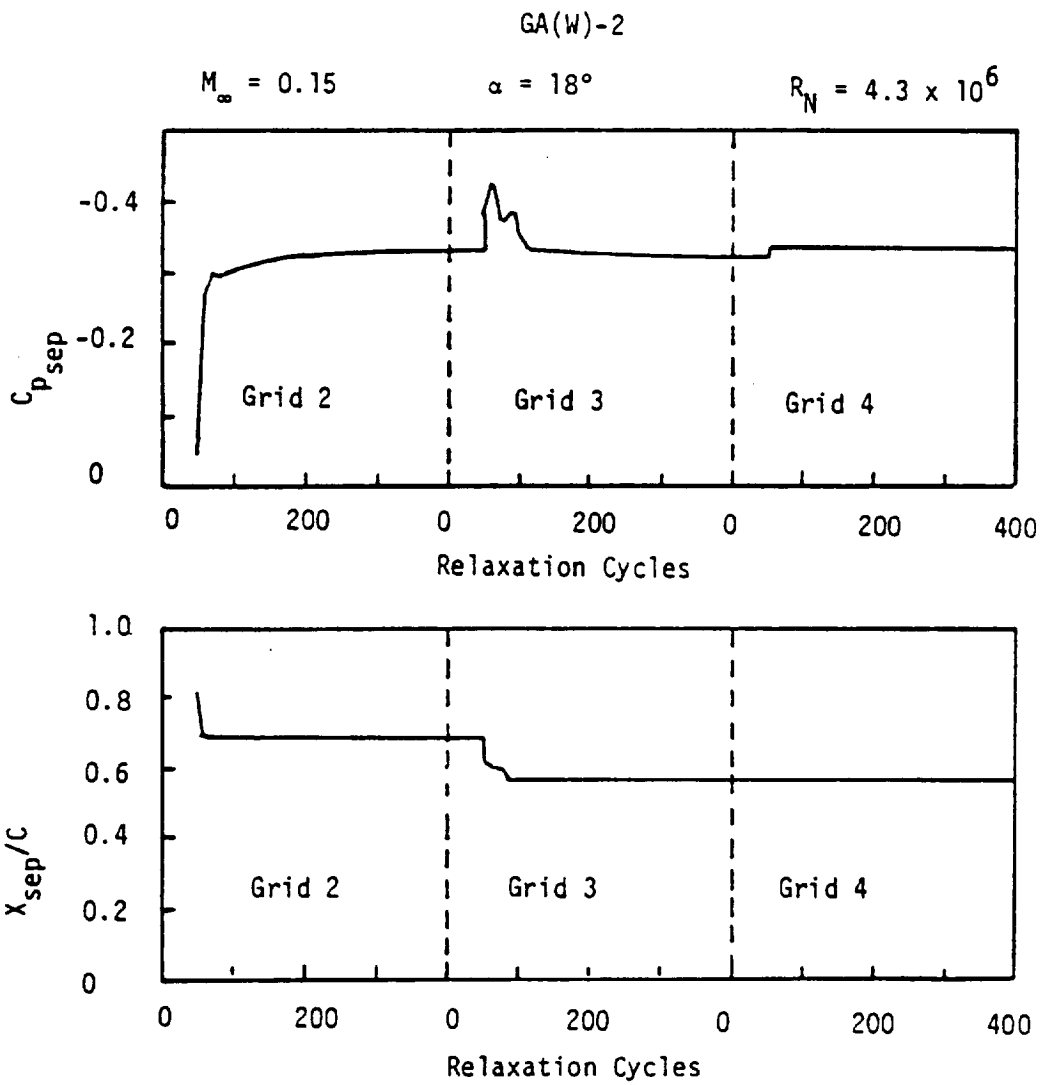


Figure 2.4-8. Separation Point and Pressure Behavior During Relaxation Process

performed on both the medium and fine grids. Also, the location of the separation point is held fixed on the fine grid (typically 97 x 49 with 130 points on the airfoil) at the x/c location computed on the medium grid (49 x 25 with 66 points on the airfoil).

Extensive investigations with the present model (Ref. 2.4-16) have shown that the theoretical results are primarily dependent upon the accurate prediction of the upper surface separation-point location. In the Nash-Macdonald integral method, separation is assumed to occur when a parameter, SEP, defined as

$$SEP = - \frac{\theta}{q} \frac{dq}{ds} \quad (2.4-62)$$

exceeds a specific value, SP. Here  $\theta$  is the local momentum thickness, and  $q$  is the local boundary layer edge velocity. Obviously, agreement between the theoretical and experimental values requires an appropriate selection of the value for SP in order to obtain the correct separation point.

Theoretically, in the Nash-Macdonald method, separation occurs whenever the displacement-momentum thickness ratio,  $\delta^*/\theta$ , exceeds three. At low speeds and low angles of attack, this value usually corresponds to an SP of about 0.0055. In Reference 2.4-15, however, it was found that at transonic speeds, an SP value of 0.0040 was more appropriate. Interestingly, the present study indicates that accurate results can only be obtained at high angles of attack by making SP a function of angle of attack,  $\alpha$ , Reynolds number, RN, and the low angle of attack lift curve slope,  $C_{L\alpha_0}$ . The present best estimate for this correlation is:

For  $\alpha \leq 15.3^\circ$

$$\text{Use: } SP = -7.14352 \times 10^{-5} \alpha + (0.0142857 C_{L\alpha_0} + 0.004714337)$$

$$\text{If: } SP > 0.0055, \text{ set } SP = 0.0055$$

For  $\alpha > 15.3^\circ$

$$\text{Use: } SP = 3.6213784 \times 10^{-3} + 0.0142875 C_{L\alpha_0} + (-8.4074 \times 10^{-11} RN + 2.1707 \times 10^{-4}) (\alpha - 15.3)$$

where the angle of attack,  $\alpha$ , is always in degrees.

This formula should be viewed strictly as an empirical attempt to force the correction separation point behavior into the Nash-Macdonald approach as it is formulated in a finite difference grid. Thus, the user should anticipate further empirical modification of it if his experience so dictates. This alteration should not be viewed as a criticism of the Nash-Macdonald method. Nevertheless, it should lead to reasonable results for the flow about low-speed airfoils up to  $C_{Lmax}$ , and sometimes even beyond  $C_{Lmax}$ .

#### 2.4.1.7 Input Variable Description

The input cards are summarized below:

<u>Read Order</u>	<u>Variables</u>	<u>Format</u>
1	NTITLE	20A4
2	NAMelist/FINP/M,W,X1,X2,ALP,EPS,EPSS,X4,S4,CONV,A1,A2,A3,RN,XIBDLY,CIR,CDCORR,RDEL,RDELFN,SP,XSEP,XLSEP,XPC	Namelist
3	NAMelist/IINP/IMAX,JMAX,IKASE,INV,MITER,NHALF,ITACT,ISKP2,ISKP3,ISKP4,ITERP,IREAD,LP,ITEUPC,ITELWC	Namelist
4	P(I,J) I=1,IMAX; J=1,JMAX (Only if IREAD=1)	5E15.7
5	PB(I) I=1,IMAX (Only if IREAD=1)	5E15.7
6*	X1,X2	2F10.5
7	NI	I5
8	XI(I),YI(I), I=1,NI	8F10.4
9	DERIX,DERIY,DERFX,DERFY	8F10.4
10	NIB	I5

---

\*Read only in the design mode when INV=1.



11	XIB(I),YIB(I), I=1, NIB	8F10.4
12	DERIXB,DERIYB,DERFXB,DERFYB	8F10.4
13*	X1,X2	2F10.5
14*	CPU(I), I=I1, ITE	8F10.3
15*	CPL(I),I=I1, ITE	8F10.3
16*	X1,X2	2F10.5
17*	CPU(I),I=I1,ITE	8F10.3
18*	CPL(I),I=I1, ITE	8F10.3
19*	X1,X2	2F10.5
20*	CPU(I), I=I1, ITE	8F10.3
21*	CPL(I), I=I1, ITE	8F10.3

NOTE: In the design mode Steps 13-15 are for the coarse grid, 16-18 for the medium and 19-21 for the fine grid (if used).

The definitions of these input variables are as follows:

NTITLE - Description of case. Up to 80 alphanumeric characters. Appears on printed output, at the beginning of the results for each grid.

M - Freestream Mach number (real variable). Default 0.5.

W - Relaxation factor for subsonic points. Should be in the range  $0 < W < 2.0$ . Default 1.7.

X1 - X location where direct calculation stops. In analysis mode it should be set to 0.5 (i.e., trailing edge). In design mode it is usually set to slightly less than the third point from the leading edge or larger. Default 0.5.

X2 - End of the inverse region. For analysis case set to a large number. In inverse design case set to 0.5 (i.e., trailing edge). Default 10000.0.

ALP - Angle of attack in degrees. Default 0.0.

---

\*Read only in the design mode when INV=1.

- EPS - Subsonic damping factor to match difference equations at sonic line if needed. EPS has no effect on accuracy of solution, only on stability and convergence rate. Normally it is not needed. Default 0.0.
- EPSS - Supersonic damping factor for iterative stability. Note that EPSS has no effect on the accuracy of the converged solution, only on the stability and convergence rate. EPSS should typically be about  $M_{\max}^2 - 1$ , where  $M_{\max}$  is the maximum local Mach number. Default 0.4.
- X4 - The positive X location where the coordinate stretching changes. It should be near the airfoil trailing edge. Default 0.49.
- S4 - The positive  $\zeta$  value in the computational plane where the stretching changes. Default 2.0.
- CONV - Convergence criteria control value. Iterations stop when the maximum change in the perturbation potential (between relaxation cycles) is less than CONV. Default 1.E-05.
- A1 - Stretching constant for the Y direction. It can be used to control  $\Delta Y$  and  $\Delta \eta$  near the horizontal axis. It is usually best to have  $\Delta \zeta = \Delta \eta$  near the leading edge of the airfoil. Default 0.246.
- A2 - First stretching constant for the X-direction. It is equivalent to  $\frac{2}{\pi} \left( \frac{dx}{d\zeta} \right)$  at  $\zeta = \zeta_4$ . The value of  $A_2$  determines the horizontal step size near the leading and trailing edges, i.e.,

$$\Delta X_{x=x_4} = \frac{\pi A_2}{2} \Delta \zeta = \frac{\pi A_2}{2} \frac{(2(1+S_4))}{(IMAX-1)}$$

Default 0.15.

- A3 - Second stretching constant for the x-direction. It determines the physical location of the vertical grid line adjacent to grid side edge. Default 3.87.

- RN - Freestream Reynolds number based on chord length. Used only when viscous interaction included. Default 20.E+06.
- CIR - Circulation about airfoil. If an initial solution is input, it must be the corresponding value of circulation. ( $CIR = C_L/2.0$ ). Default 0.0.
- CDCORR - Correction to the computed wave drag coefficient for the finest grid used. Because of the lack of a large number of points in the leading and trailing edge regions, the wave drag coefficient has an error associated with grid size, spacing, and lift coefficient. The magnitude of CDCORR as a function of lift can be determined from a series of calculations at different angles of attack at subcritical speeds, where the wave drag should be zero. Note that the correction should be determined for each airfoil and grid combination. Default 0.0.
- RDEL - Relaxation parameter for the boundary layer displacement thickness. It is used only when viscous interaction is included and  $IMAX \leq 55$ . Default 0.25.
- RDELFN - Fine grid relaxation parameter for the boundary layer displacement thickness. It is used only when viscous interaction is included and  $IMAX > 55$ . Default 0.125.
- SP - Maximum value allowed for the Nash-Macdonald separation parameter when  $x > XSEP$ . Used only in the viscous interaction case. Default 0.004.
- XSEP - X location after which the Nash-Macdonald separation parameter can assume its calculated value. Used only in the viscous interaction case. Default 0.44.
- XPC - Location after which the lower surface displacement thickness is required to continue decreasing once it has started to decrease. Upstream of XPC the displacement thickness is required to be monotonically increasing. For most aft-cambered airfoils it should be 0.1 and in conventional airfoils it should be 0.5. Default 0.1.

- IMAX - Number of vertical grid lines in the horizontal direction.  $I = 1$  is upstream infinity and  $I = \text{IMAX}$  is downstream infinity. For each grid refinement IMAX is increased such that  $\text{IMAX}_{\text{new}} = 2(\text{IMX}_{\text{old}}) - 1$ . The limit on IMAX is 99. Default for use on first grid is 13.
- JMAX - Number of horizontal grid lines in the vertical direction.  $J = 1$  corresponds to infinity below the airfoil and  $J = \text{JMAX}$  is infinity above the airfoil. The same formula and limit that apply to IMAX also apply to JMAX. Default 7.
- IKASE - An integer number describing the case. It is limited to a maximum of six digits. Default 100.
- INV - Parameter determining program mode. It should be zero for analysis cases and one for inverse design cases. Default 0.
- MITER - Maximum number of iterations (complete relaxation cycles) allowed on first grid. MITER is halved for each grid refinement. However, on the fourth grid, MITER is reset to 400. Default 2.
- NHALF - Number of grid refinements to be done. Default 2.
- ITACT - Viscous interaction control parameter. It should be set to zero for analysis cases without interaction and for design cases. It should be one for analysis cases with interaction. Default 0.
- ISKP2 - Airfoil update control parameter for grid two. It should be 0 if on grid two an update is desired every 10 iterations. It should be 1 if an update is not desired until the grid two solution is completed. Only used in the inverse design mode. Default 0.
- ISKP3 - Same as ISKP2 but for grid 3 (medium grid).
- ISKP4 - Same as ISKP2 but for grid 4 (fine grid).
- ITERP - Interpolation parameter. If in the design mode the input  $C_p$  distribution for the grid 4 is to be read in, ITERP should be 0. If it is desired to linearly interpolate the  $C_p$  distribution of grid 3, it should be 1. Default 0.

- IREAD - Starting solution control parameter. If IREAD is 0, the initial perturbation solution is assumed to everywhere be zero. If it is 1, an initial solution is read in from data cards. Default 0.
- LP - Relaxation cycle interval at which boundary layer, surface ordinates, etc., details are printed. Useful for diagnostics. Default 1000. (No printout.)
- ITEUPC - Upper surface trailing edge correction control parameter. If trailing edge correction desired, ITEUPC should be 1. If not it should be zero. Only used in the viscous interaction case. Normally the correction is not needed. Default 0.
- ITELWC - Lower surface trailing edge correction control parameter. If correction desired, ITELWC should be 1. If not it should be 0. Only used in the viscous interaction case, and normally the correction is not needed. Default 0.
- P(I,J) - Nondimensional perturbation potential,  $\phi_{ij}$ , at point I,J.
- PB(I) - Nondimensional perturbation potential at point I on the y=0 grid line.
- X1, X2 - Same definition as above. However, in the inverse design case they must be read in prior to the solution of each grid. On the first grid (Step 6 in above listing) should use X1=0.5, X2=10000.0. On remaining grids (Steps 13, 16, and 19), X1 should be the location where the direct calculation stops and X2 should be 0.5.
- NI - The number of coordinate pairs used to describe the upper surface of the airfoil. Presently limited to 99.
- XI(I) - Input coordinates in the horizontal direction for the airfoil upper surface. The leading edge corresponds to XI=0.0 and the trailing edge is XI=1.0.
- YI(I) - Input coordinates in the vertical direction for the airfoil upper surface.

- DERIX -  $DX/DS$  of the airfoil upper surface at the leading edge ( $XI=0.0$ ).  
It usually is 0.0.
- DERIY -  $DY/DS$  of the airfoil upper surface at the leading edge ( $XI=0.0$ ).  
It usually is 1.0.
- DERFX -  $D^3X/DS^3$  of the airfoil upper surface at the trailing edge  
( $XI=1.0$ ). It is usually sufficiently accurate to use 0.0.
- DERFY -  $D^3Y/DS^3$  of the airfoil upper surface at the trailing edge  
( $XI=1.0$ ). It is usually sufficiently accurate to use 0.0.
- NIB - The number of coordinate pairs used to describe the lower surface  
of the airfoil. Presently limited to 99.
- XIB(I) - Input coordinates in the horizontal direction for the airfoil  
lower surface. The leading edge corresponds to  $XIB=0.0$  and the  
trailing edge is  $XIB=1.0$ .
- YIB(I) - Input coordinates in the vertical direction for the airfoil lower  
surface. Since positive is up, the values of YIB are usually  
negative.
- DERIXB -  $DS/DS$  of the airfoil lower surface at the leading edge. It is  
usually 0.0.
- DERIYB -  $DY/DS$  of the airfoil lower surface at the leading edge. It is  
usually -1.0.
- DERFXB -  $D^3X/DS^3$  of the airfoil lower surface at the trailing edge. It is  
usually sufficiently accurate to use 0.0.
- DERFYB -  $D^3Y/DS^3$  of the airfoil lower surface at the trailing edge. It is  
usually sufficiently accurate to use 0.0.
- CPU(I) - Upper surface inverse region  $C_p$  values for design case. I1, which  
is computed internally, is the first grid point after X1. The  
distribution must be read in for each grid solved inversely (Steps  
14, 17, and 20).
- CPL(I) - Lower surface inverse  $C_p$  values for design case. They must be  
read in for each grid solved inversely (Steps 15, 18, and 21).

XIBDLY - The x-location at which upper surface transition is assumed to occur. The turbulent boundary layer calculation starts at the next grid point. The relationship to percent chord is:

$$XIBDLY = (\%chord - 50.0)/100.0$$

Used only if ILAM = 0. Ignored if ILAM = 1.

Default -0.44.

XLSEP - When IMASS = 0, location at which the trailing edge correction procedure begins. Between XLSEP and the trailing edge the pressure distribution and the displacement surface is modified. Used in this fashion only if ITEUPC and/or ITELWC equal 1. When IMASS = 1, set to a number forward of expected separation point. Default 0.50.

XLBDLY - The x-location at which lower surface transition is assumed to occur. Same relationship to chord is XIBDLY. Used only if ILAM = 0. Ignored if ILAM = 1. Default - 0.44.

RLAX - Relaxation parameter for separated pressure level. Sometimes needed to enhance convergence. It is used only when ITACT = 1. IMASS = 1. Default 1.0.

RADUS - Leading edge nose radius,  $r_{LE}/c$ . Used only if ILAM = 1. Default 0.035.

CLALP - Low angle of attack lift curve slope. Used only if IMASS = 1. Default 0.10966.

ILAM - Boundary layer parameter. If zero, boundary layer is all turbulent. If one, boundary layer is laminar-turbulent with natural transition. Default 0.

IPRT1 - Print parameter. If one, perturbation potential values printed at the completion of each grid. Default 0.

IPRT2 - Print parameter. If one, x and y velocities at each grid point printed at the completion of each grid. Values printed as part of Mach chart. Default 0.

IMASS - Massive separation parameter. It should be one for massive separation cases and zero for all others. To utilize, ITACT must be 1. Default 0.

#### 2.4.1.8 Output Description

The program output for each grid in the massive separation mode is:

- (1) Heading
- (2) Coordinate system. It is printed as I,X(I) followed by J,Y(J).
- (3) Mach number, angle of attack and location where direct calculation stops
- (4) Case number
- (5) Case type callouts, i.e.,
  - Inviscid analysis case
  - With viscous interaction
  - Massive separation
- (6) Input data in namelists FINP and IINP
- (7) Airfoil ordinates in direct region
  - X - horizontal ordinate, where -0.5 is leading edge and 0.5 is the trailing edge
  - YU - Upper surface ordinate
  - YL - Lower surface ordinate
  - Upper Slope - Upper surface slope
  - Lower Slope - Lower surface slope
- (8) Iteration history at ten-cycle intervals
  - CIR - Circulation
  - DPM - Maximum  $\phi$  correction (absolute value) in the last relaxation cycle, with the corresponding (I,J) grid location
  - NSSP - Number of supersonic points



DELTAY or DELSTAR -  $\delta^*$  at the trailing edge for massive separation, should go to a constant if solution is converging. Only changes after first 50 cycles on each grid.

Separation point and separated  $C_p$  are printed.

(9) Laminar boundary layer values (every LP cycles)

X - Horizontal ordinate

CF - Skin Friction Coefficient (the  $0.1 \times 10^{11}$  value is an arbitrary initial value and should be ignored)

D-STAR -  $\delta^*/c$ , nondimensional boundary layer displacement thickness

D-THETA -  $\theta/c$ , nondimensional momentum thickness

H -  $\delta^*/\theta$ , shape factor

RE-THETA - Reynolds number based on  $\theta$

RE-STAR - Reynolds number based on  $\delta^*$

TM -  $\theta^2 \frac{du}{dx}/\nu$  Pressure gradient parameter

(10) Laminar flow messages

(a) SEPARATION OCCURRED AT X = \_\_; gives x location where laminar separation occurred

(b) SHORT BUBBLE FORMED? TRANSITION TO TURBULENT FLOW ASSUMED, X = \_\_

(c) LONG BUBBLE? LAMINAR STALL MAY OCCUR, X = \_\_ BOUNDARY LAYER CALCULATION WILL BE CONTINUED AS TURBULENT BUT ACCURACY OF RESULTS IS QUESTIONABLE

(d) BOUNDARY LAYER CALCULATION COMPLETED? NEITHER SEPARATION NOR TRANSITION WAS DETECTED

(11) Turbulent boundary layer values

X - Horizontal ordinates

M - Local Mach Number

DELS -  $\delta^*$ , displacement thickness

THETA -  $\theta$ , momentum thickness

SEP - parameter used to determine if separation occurs

H - shape factor

PI -  $\frac{\delta^*}{\tau_w} \frac{dp}{dx}$ , pressure gradient parameter

TAU -  $\tau_w/\rho u^2$ , nondimensional wall shear force

CF, D-STAR, D-THETA, H, RE-THETA, RE-STAR, TM (see 9)

(12) Lower surface flow messages

(a) BOUNDARY LAYER CALCULATION COMPLETED?

(b) NEITHER SEPARATION NOR TRANSITION WAS DETECTED

(13) Upper surface flow message gives separation point and  $C_p$  for separated region

(14) 1st group - x - horizontal ordinates

YU, YL, location of the surface ordinates in the computational plane

2nd group - x - horizontal ordinates

YU, YL, slopes of upper and lower surfaces in physical plane

repeat (8) through (14) until convergence

(15) Final boundary layer results

YUORIG - Original airfoil upper surface ordinates

DU - Upper surface displacement thickness

SLU - Slope of upper displacement surface

YLORIG - Original airfoil lower surface ordinates

DL - Lower surface displacement thickness

SLL - Slope of lower displacement surface

(16) Pressure distribution on airfoil

(17) Displacement surface ordinates and slopes in physical plane

(18) Mach chart, numbers printed are Mach number multiplied by 100. Velocities (U,V) may also be printed.

(19) Wave drag coefficient

## (20) Plot of results

U - Upper surface  $C_p$

L - Lower surface  $C_p$

T - Upper displacement surface

B - Lower displacement surface

CLCIR - Lift coefficient from circulation

CL - Lift coefficient from integration of  $C_p$

$C_D$  - CDWAVE + CDF

CMLE - Moment coefficient about leading edge

CDF - Drag coefficient by modified Squire-Young

CMC4 - Moment coefficient about quarter chord

NOTE: For low speed massive separation, ignore CDWAVE and CD.

Use CDF only.

### 2.4.2 Airfoil Data Banks

The current version of GAPAS contains two airfoil data banks. The first is for the Clark-Y series and the second, the NACA 16 series. The details of the data banks are described below.

#### 2.4.2.1 Clark-Y Series

The Clark-Y airfoil data bank calculates the lift coefficient, drag coefficient, lift-to-drag ratio, and the moment coefficient about the leading edge or about the quarter chord. Some limitations exist in the program, and when exceeded, flags are built into the program to indicate what value exceeded the limitation and where in the program it occurred, under "Limitations" in the output of the program.

#### Input

The input list includes the following parameters: Mach number(s) (HM); angle(s)-of-attack (ALPHA) referenced from the longest chord in degrees; altitude (ALTUDE) in feet; chord length (CHORD) in feet; thickness-to-chord ratio (T) - equal to the known value of the thickness-to-chord ratio or equal to 0 if the value is unknown; number of Mach numbers (NUMACH); number of angles-of-attack (NUMALP); lift coefficient factor (NCL) - if not equal to 0, the lift coefficient is calculated; drag coefficient factor (NCD) - if not equal to 0, the drag coefficient is

calculated; airfoil section coordinates factor (L) - if not equal to 0, the Clark-Y airfoil section coordinates and its moments of inertia are given; design lift coefficient or equal to 0 if the value is unknown; moment coefficient parameter (M) - if equal to 0 a moment coefficient is not calculated, if equal to 1 the moment coefficient about the leading edge is calculated, and if equal to 2 the moment coefficient about the quarter chord is calculated.

#### Main Program Description

When the number of data are changed for parameters HM (Mach number) and ALPHA (angle of attack), the dimension statements DATA NUMACH and DATA NUMALP must also be changed accordingly.

#### Subroutines

- (1) ATMCON calculates temperature, pressure, speed of sound, density, and viscosity for given altitude.
- (2) CYCOOR gives the dimensions and section properties for a Clark-Y airfoil section for any given thickness and reference chord. It also calculates the moment of inertia for the major and minor axes. The input units will correspond with the output units, however the coordinates are nondimensional with respect to the reference chord. The upper and lower x-coordinates are identical.
- (3) CLTMAX contains the equation for design lift coefficient as a function of maximum thickness-chord ratio, and the equation for maximum thickness-chord ratio as a function of design lift coefficient. These equations are generated from plotting the calculated CLD versus (T/C)MAX using the data from NACA TR 628 (Ref. 2.4-17). The design lift coefficient and thickness-to-chord ratio have a parabolic relationship and are fitted to the curve with the equation  $NCLD = (-0.16879 + \text{SQRT}(0.02849 + 0.496405 * NTMAX)) / 0.248205$ . If the thickness-to-chord ratio (NTMAX) needs to be calculated, the following two equations are used:

(a)  $NTMAX=0.1241*NCLD**2+0.16879*NCLD$  if NCLD is positive,

or

(b)  $NTMAX=0.1241*NCLD**2-0.16879*NCLD$  if NCLD is negative.

The section slope of the low speed lift curve for each thickness is also calculated and stored in FUNCTION AOIOF. With the value for CLD and AO, the angle of attack at zero lift is calculated.

- (4) CLCD contains numerous functions which interpolate the contained data tables through Lagrangian method, and determines the lift and drag coefficients according to corresponding Mach number, maximum thickness-chord ratio and angle-of-attack. The data base for this subroutine is from Haines and Monaghan (Ref. 2.4-18) and MacDougall (Ref. 2.4-19). The equations and data tables contained in this document form the basis for all further discussion concerning this subroutine.

The lift coefficient is calculated by multiplying the lift curve slope by the angle-of-attack relative to the zero lift line. If the thickness-to-chord is less than 16 percent, the critical Mach number is determined. If the Mach number is less than the critical Mach number, the lift coefficient calculated includes the Prandtl-Glauert correction. If the Mach number is greater than the critical Mach number, the lift coefficient is calculated by the following equation:  $CL=CLO/SQRT(1-HML*HML)+CLS(H)$  where CLO is the incompressible lift coefficient, HML is the critical Mach number, and the function CLS(H) calculates the difference of the lift coefficient at a Mach number above the critical from its critical value.

For the calculation of the drag coefficient, data tables are set for thickness ratios of 0.09, 0.10, 0.11, 0.12, 0.13, 0.14, 0.15, 0.16, 0.17, 0.19, 0.21, 0.23, 0.25, 0.30, and 0.35 at angles-of-attack of 3, 4, 5, 6, 7, 8, 9, 10, and 11 degrees. A Lagrangian interpolation routine interpolates between the tables to obtain the desired drag coefficient.

- (5) Function CMFF contains the moment coefficients about the leading edge and about the quarter chord. The variation of the moment coefficient with the lift coefficient is determined for several thickness-to-chord ratios through data obtained from the Smetana, et.al. airfoil analysis code. A linear relationship is found and fitted to a simple line equation. The variation of the slope and the intercept with the thickness-to-chord are determined, and quadratic equations are fitted to the curves. The quadratic equations are substituted into the line equation to obtain the final equation for the moment coefficient. The equation for the moment coefficient about the leading edge is,  $CMLE=T^{**2}*(-0.3065*CL+0.875)-T*(0.1514*CL+0.705)-0.23993*CL+6.25/100000$ . The equation for the moment coefficient about the quarter chord is,  $CMQC=(-0.44*CL+1.15)*T^{**2}-(0.0993*CL+0.785)+0.0036*CL+0.00638$ . The compressibility effects are included by dividing the moment coefficient by the Prandtl-Glauert correction.

#### 2.4.2.2 NACA 16 Series

The NACA 16-Series airfoil data bank calculates the lift coefficient, drag coefficient, lift-to-drag ratio, and the moment coefficient about the leading edge or about the quarter chord. Some limitations exist in the program, and when exceeded, flags are built into the program to indicate what value exceeded the limitation and where in the program it occurred, under "Limitations" in the output of the program. A Mach number of 0.0 is not allowable. Also, the program has difficulty handling negative angles-of-attack of large magnitude at high Mach numbers for thick airfoils.

#### Input

The input list includes the following parameters: the drag coefficient compressibility check factor (ICD) - if equal to 0, the incompressible drag coefficient is given, and if equal to 1, the compressible drag coefficient is given; the lift coefficient compressibility factor (ICL) - if equal to 0, the incompressible lift coefficient is given, and if equal to 1, the compressible lift coefficient is given; a check parameter which controls the format of the output (IFORM) - if equal to 0, all of the increments in the drag are given, and if equal

to 1, a tabular output of the lift coefficient (CL), the drag coefficient (CD), the lift-to-drag ratio (CL/CD), and the moment coefficient about the leading edge (CM(LED)) or the moment coefficient about the quarter chord (CM1/4)) is given for each angle-of-attack (ALPHA), one table is given for each Mach number; number of angles-of-attack specified (NUMALP); number of Mach number; number of angles-of-attack specified (NUMALP); number of Mach numbers (NUMACH); design lift coefficient (CLD) - equal to the known value of the design lift coefficient or equal to 0 if the value is unknown; the airfoil section area (AREA) - equal to the known value of the section area or equal to 0 if the value is unknown; thickness-to-chord ratio (TC); chord length (CHORD) in feet; altitude (ALTUDE) in feet; Mach number(s) (M); angle(s)-of-attack (ANGLE) referenced from the longest chord in degrees; angle of zero lift (ALPHA0); radial location (X); the moment coefficient parameter (MP) - if equal to 0, a moment coefficient is not given, if equal to 1, the moment coefficient about the leading edge is given, and if equal to 2, the moment coefficient about the quarter chord is given; and the airfoil coordinate parameter (L) - if not equal to 0, the NACA 16-Series airfoil section coordinates and its moments of inertia are given.

#### Main Program

When the number of data are changed for parameters M (Mach number) and ANGLE (angle of attack), the dimensions for M and ANGLE, DATA NUMACH, and DATA NUMALP must also be changed accordingly.

For the calculation of the moment coefficient about the quarter chord, data tables are set up for design lift coefficients of 0.0, 0.1, 0.3, and 0.5 at Mach numbers of 0.3, 0.45, 0.6, 0.7, and 0.75 with thickness ratios of 0.02, 0.06, 0.09, 0.12, 0.15, 0.21, and 0.30. A Lagrangian interpolation routine interpolates between the tables to obtain the desired moment coefficient. The data tables are taken from Lindsey, Stevenson, and Daley (Ref. 2.4-20). The moment coefficient about the quarter chord is transformed to the leading edge with the following equation:

$$CMLE=CMQC-0.25*CL.$$

function of thickness. Given a thickness and a Mach number, the program calculates the drag-divergence Mach number by linear interpolation, and tests to see if the specified Mach number is above or below the drag-divergence Mach number. It then calculates the basic drag coefficient accordingly. For subsonic section velocities, the subroutine performs a linear interpolation if the local Mach number is below the drag-divergence Mach number, or uses a polynomial curve fit for local Mach numbers above the drag-divergence Mach number. For supersonic section velocities, the subroutine uses a polynomial curve fit on the supersonic segment of the curve to arrive at a value for the basic drag coefficient. Note that if the thickness is less than 0.03 or if the basic drag coefficient is negative, then the basic coefficient is set equal to 0.0.

Data for the drag coefficient due to lift calculation is based on Cooper (Ref. 2.4-21). First,  $\text{DELTA} = (\text{CLXCLI})$ . Then, the subroutine calculates the weighted Mach number,  $\text{XMXT} = \text{XM} / (0.9 - \text{XTC})$ . Then it calculates DELCD using the formula:  $\text{DELCD} = (\text{XMXT}^2 - \text{XMXT} - 0.1765) * (-0.4397 * \text{DELTA}^2 + 0.1173 * \text{DELTA} - 0.0938)$ . Next, TESTM is calculated. TESTM is the value of XMXT on the line labeled LIM A in Figure 9A of Reference 2.4-23 which corresponds to the value of DELCD just calculated:  $\text{TESTM} = (\text{DELCD} - 0.03) * (0.9 - 0.875) / (0.03 - 0.08) + 0.9$ . If the value of XMXT is greater than TESTM, the equation above does not hold, and the value of DELCD is arrived at by using a sixth degree polynomial curve fit to find the slope of the plot. Then,  $\text{DELCD} = \text{SLOPE} * (\text{XMXT} - 1.011111) + 0.0911111$ . The subroutine then tests if the Mach number is supersonic. If so,  $\text{DELCD} = \text{SQRT}(\text{XM}^2 - 1) / 4$  and  $\text{DELTA} = \text{CL}$ . Once DELTA and CL have been calculated in the proper manner, the drag coefficient due to lift is given by:  $\text{CDLIFT} = \text{DELCD} * \text{DELTA}^2$ .

The coefficient of drag due to friction is a function of Reynolds number and percent laminar flow, and is also based on the work of Cooper. This subroutine determines the percent laminar flow over the airfoil (PLAM) by a second degree polynomial curve fit:  $\text{PLAM} = (33 + 1/3) * \text{XR}^2 + (8 + 1/3) * \text{XR} + 17$ , after which, the Reynolds' number for the airfoil is calculated by  $\text{XRN} = \text{DEN} * \text{XM} * \text{WA} * \text{CHORD} / \text{VIS}$ . When the Reynolds number and the percent laminar flow are determined, the drag coefficient due to friction is calculated



from:  $CDFRIC=2.65/\sqrt{XRN}+(100-PLAM)/PLAM*(0.944/DLOG10(XRN)**2.65/\sqrt{XRN})$ .

The drag due to camber represents the increase in the minimum basic drag coefficient of cambered NACA 16-Series airfoils over that of the uncambered airfoils of the same series. The drag coefficient due to camber (CDCAMB) is completed in the form of a plot of CDCAMB vs. XMXT as a function of the design lift coefficient. The subroutine performs a series of linear interpolations to find the proper drag coefficient due to camber at a specified XMXT for a specified design lift coefficient.

## 2.5 ACOUSTICS MODULE

The primary acoustics analysis procedure and computer code selected for inclusion in GAPAS is that developed at NASA-Langley by Martin and Farassat. The computer code is called SPN (Subsonic Propeller Noise); an advanced version to include transonic and supersonic flow effects, called TPN (Transonic Propeller Noise) has also been developed by NASA-Langley and incorporated into the current version of GAPAS.

### 2.5.1 Problem Definition

This module computes the periodic acoustic pressure signature and spectrum of a propeller rotating at subsonic tip speed (SPN) or transonic/supersonic tip speeds (TPN). The computation is based on a solution of the Ffowcs Williams-Hawkings (FW-H) equation without the quadrupole source term (Refs. 2.5-1 through 2.5-3). The blade surface pressure can also be periodic in time. The relative position of the observer with respect to the center of rotation of the noise source is always assumed to be fixed (e.g., the observer is always assumed to be moving with the aircraft); only in such a frame is the acoustic pressure periodic. The main acoustic formulation uses the source distribution on the actual blade surface. Several approximations to the full solution are included as options.

### 2.5.2 Subsonic Propeller Noise (SPN)

A definition of symbols is provided below in order to allow the reader to more easily follow the analysis methodology.

#### Symbols

A	Area of a blade section, $m^2$ ( $ft^2$ )
c	Ambient speed of sound, $m/s$ ( $ft/s$ )
$d_x$	Radial distance of observer from $x_3$ -axis, $m$ ( $ft$ )
$d_y$	Radial distance of source from $x_3$ -axis, $m$ ( $ft$ )
f	Function defining the blade surface

$f_k$	Frequency, Hz
F	Symbol substituted for integrands in Equation (2.5-41)
G	Symbol substituted for integrands in Equation (2.5-45)
H	Azimuthal force on a single blade, N (lbf)
g	Equation $g = 0$ describes relationship between source and observer time
k	Harmonic number
$\vec{r}$	Local force per unit area of the blade acting on the fluid, Pa (lbf/ft <sup>2</sup> )
$\vec{F}$	Net force of the blade acting on the fluid, N (lbf)
$\vec{F}'$	Net force per unit radial distance, N/m (lbf/m)
m	Exponent; $2^m$ is the number of time points used by FFT routine
M	Source Mach number
$\hat{n}$	Blade surface unit normal vector
$N_b$	Number of blades
$N_s$	Maximum harmonic number in calculations
$N_t$	Number of times
p	Surface pressure ( $p = p_a - p_0$ ), Pa (lbf/ft <sup>2</sup> )

$p'$	Acoustic pressure, Pa (lbf/ft <sup>2</sup> )
$p_a$	Absolute surface pressure, Pa (lbf/ft <sup>2</sup> )
$p_o$	Ambient pressure, Pa (lbf/ft <sup>2</sup> )
$p'_L$	Acoustic pressure produced by loading, Pa (lbf/ft <sup>2</sup> )
$p'_p$	Total acoustic pressure produced by propeller, Pa (lbf/ft <sup>2</sup> )
$p'_T$	Acoustic pressure produced by thickness, Pa (lbf/ft <sup>2</sup> )
$\Delta p$	Difference between lower and upper surface pressure, Pa (lbf/ft <sup>2</sup> )
$p_{ref}$	Reference pressure for definition of dB, $2 \times 10^{-5}$ N/m <sup>2</sup>
$\vec{r}$	Vector from source point at emission time to observer, m (ft) ( $\vec{r} =  \vec{x} - \vec{y}(\tau^*) $ )
$\hat{\vec{r}}$	Unit vector in the direction of $\vec{r}$
$R_E$	Effective radial distance to be used in point source approximation, m (ft)
$SPL_k$	Acoustic spectrum, dB
$t$	Time at which noise signal is received by observer (s)
$\hat{\vec{t}}$	Blade surface unit tangent vector
$T$	Net thrust from a single blade, N (lbf)
$v$	Source velocity, m/s (ft/s)

$V_F$	Forward velocity of the aircraft, m/s (ft/s)
$\vec{x}$	Observer position in the ground-fixed frame, m (ft)
$\vec{x}$	Observer position in the aircraft-fixed frame, m (ft)
$\vec{y}$	Source position in the ground-fixed frame, m (ft)
$\vec{y}$	Source position in the aircraft-fixed frame, m (ft)

Greek

$\delta$	Dirac delta function
$\vec{\eta}$	Source position in the blade-fixed frame, m (ft)
$\theta_0$	Change in root pitch angle, rad
$\theta$	Angle between the radiation vector and the surface normal vector, rad
$\theta'$	Angle between the radiation vector and the surface tangent vector, rad
$\xi_1, \xi_2$	Elliptic surface coordinates
$\rho_0$	Ambient density, kg/m <sup>3</sup> (slug/ft <sup>3</sup> )
$\sigma$	Local tangential stress, N/m <sup>2</sup> (lbf/ft <sup>2</sup> )
$\tau$	Time at which noise signal is emitted at source position, s (s)
$\tau^*$	Solution to the retarded time equation, $g = 0$ , for a given observer time, s (s)
$t$	See Equation (2.5-27)
$\phi$	Solution to retarded time equation as solved by Newton's method ( $\phi = \Omega(\tau - t)$ ), s (s)

ϕ Angle between the  $x_1$  and  $\eta_1$  axes, rad

ψ Blade volume,  $m^3$  ( $ft^3$ )

### Superscripts

→ Vector

— Mean

$\hat{\rightarrow}$  Unit vector

• Time derivative

\* Complex conjugate; specific solution to retarded time equation

### Subscripts

i Component along the  $i^{\text{th}}$  coordinate axis  
(note:  $\hat{r}_i$  implies summation convention); also inner radius

m Property of the mean blade surface

n Component in the direction of the surface normal

o Ambient condition; also outer radius

r Component in the direction of the radiation vector

#### 2.5.2.1 Acoustic Formulation

The governing acoustic equation is the wave equation

$$\frac{1}{c^2} \frac{\partial p'}{\partial t^2} - \nabla^2 p' = \frac{\partial}{\partial t} [\rho_0 v_n |\nabla f| \delta(f)] - \frac{\partial}{\partial n} [\hat{r}_i |\nabla f| \delta(f)] \quad (2.5-1)$$

This is the FW-H equation without the quadrupole source term. The blade surface is described by  $\hat{f}(x,t) = 0$ . The acoustic pressure is denoted by  $p'$ ,  $v_n$  is the local normal velocity of the blade and  $\hat{l}_i$  is the local force per unit area of the blade acting on the fluid. The density and the speed of sound in the undisturbed medium are denoted as  $\rho_0$  and  $c$ , respectively. The symbol  $\delta(f)$  stands for the Dirac delta function. The two terms on the right of Equation (2.5-1) are known as the thickness and loading noise sources, respectively. Note that Equation (2.5-1) is written in a Cartesian frame fixed to the undisturbed medium, which will be called the "ground-fixed frame." We will also use two other frames of reference which will be more suitable for describing the solution and computation of the noise.

The FW-H equation is valid in the entire three-dimensional unbounded space. The Green's function of the wave equation for the unbounded space is  $\delta(g)/4\pi r$  where  $g = \tau - t + r/c$  and  $r = |\vec{x} - \vec{y}|$ . Here  $\tau$  and  $t$  are the source and the observer times and  $\vec{y}$  and  $\vec{x}$  are the source and observer positions, respectively. Using this Green's function, the formal solution of Equation (2.5-1) is

$$4\pi p'(x,t) = \frac{\partial}{\partial t} \int \frac{1}{r} \rho_0 v_n |\nabla f| \delta(f) \delta(g) d\vec{y} d\tau - \frac{\partial}{\partial x_i} \int \frac{1}{r} l_i |\nabla f| \delta(f) \delta(g) d\vec{y} d\tau \quad (2.5-2)$$

Note that the integrals above are four dimensional as written below:

$$\int \dots d\vec{y} d\tau = \int_{-\infty}^t \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \dots d\vec{y} d\tau \quad (2.5-3)$$

Before integrating Equation (2.5-2), we use the following identity established by differentiation:

$$-\frac{\partial}{\partial x_i} \left[ \frac{\delta(g)}{r} \right] = \frac{1}{c} \frac{\partial}{\partial t} \left[ \frac{\hat{r}_i \delta(g)}{r} \right] + \frac{r_i \delta(g)}{r^2}, \quad (2.5-4)$$

where  $\hat{r}_1 = (x_1 - y_1)/r$  is the unit vector in radiation direction. Since only the function  $\delta(g)/r$  is dependent on  $\vec{x}$ , the second term in Equation (2.5-2) may be rewritten as

$$-\frac{\partial}{\partial x} \int \frac{1}{r} l_1 |\nabla f| \delta(f) \delta(g) d\vec{y} d\tau = \frac{1}{c} \frac{\partial}{\partial t} \int \frac{1}{r} l_r |\nabla f| \delta(f) \delta(g) d\vec{y} d\tau + \int \frac{1}{r^2} l_r |\nabla f| \delta(f) \delta(g) d\vec{y} d\tau \quad (2.5-5)$$

where  $l_r = l_1 \hat{r}_1$ . Using Equation (2.5-5), in Equation (2.5-2) gives

$$4\pi p'(\vec{x}, t) = \frac{1}{c} \frac{\partial}{\partial t} \int \frac{1}{r} (\rho_0 c v_n + l_r) |\nabla f| \delta(f) \delta(g) d\vec{y} d\tau + \int \frac{1}{r^2} l_r |\nabla f| \delta(f) \delta(g) d\vec{y} d\tau \quad (2.5-6)$$

Now a Cartesian frame, denoted as the  $\vec{\eta}$ -frame, fixed to the blade is introduced. This frame will be referred to as the "blade-fixed frame." It is assumed that the blade is rigid so that its surface description in the  $\vec{\eta}$ -frame is time independent. To integrate Equation (2.5-6), the following steps are employed:

- (1) Use the transformation of variable  $\vec{y} \rightarrow \vec{\eta}$ ,
- (2) Use the transformation  $\tau \rightarrow g$  to integrate with respect to the variable,  $g$ ,
- (3) Integrate the remaining delta functions.

Step (1): Since the Jacobian of the transformation is 1, it follows that the volume elements in the two spaces are equal. Also note that in the  $\vec{\eta}$ -frame the equation of the blade surface depends on  $\vec{\eta}$  only. This is obtained as follows

$$f[\vec{y}(\vec{\eta}, \tau), \tau] \equiv \tilde{f}(\vec{\eta}) = 0 \quad (2.5-7)$$



That is, the equation of the blade surface in the  $\vec{\eta}$ -frame is obtained by substituting for  $\vec{y}$  in terms of  $\vec{\eta}$  and  $\tau$ . Note that since the blade-fixed frame is in motion, the transformation  $\vec{y} \rightarrow \vec{\eta}$  is time dependent and  $\vec{y} = \vec{y}(\vec{\eta}, \tau)$ . To reduce confusion, the tilde on  $\vec{f}(\vec{\eta})$  will be dropped in the following results. The function  $g$  is defined as follows:

$$g = \tau - t + |\vec{x} - \vec{y}(\vec{\eta}, \tau)|/c \quad (2.5-8)$$

Step (2): Now fix  $\vec{\eta}$  and use the transformation  $\tau \rightarrow g$ . The Jacobian of the transformation is

$$J = \frac{1}{|\partial g / \partial \tau|_{\vec{\eta}}} = \frac{1}{|1 - M_r|} \quad (2.5-9)$$

where  $M_r = v_i \hat{r}_i / c$ . Here  $\vec{v}$  is the velocity of the point with position vector  $\vec{\eta}$  in the blade-fixed frame relative to the ground-fixed frame. In Step (3), this velocity will be defined more carefully. The result of Steps (1) and (2), above, is

$$\begin{aligned} 4\pi p'(\vec{x}, t) = & \frac{1}{c} \frac{\partial}{\partial t} \int \left[ \frac{\rho_0 c v_n + l_r}{r |1 - M_r|} \right]_{\tau^*} |\Delta f| \delta(f) d\vec{\eta} \\ & + \int \left[ \frac{l_r}{r^2 |1 - M_r|} \right]_{\tau^*} |\nabla f| \delta(f) d\vec{\eta} \end{aligned} \quad (2.5-10)$$

Here the integrals are evaluated in the blade-fixed frame over the entire space. The source time,  $\tau^*$ , is the emission time obtained by finding the root of the equation

$$g = \tau^* - t + |\vec{x} - \vec{y}(\vec{\eta}, \tau^*)|/c = 0 \quad (2.5-11)$$

For subsonic propellers considered here, this equation has only one root.

Step (3): In this step, the integrations over the delta functions in Equation (2.5-10) are performed by using the relation

$$d\vec{\eta} = \frac{df dS}{|\nabla f|} \quad (2.5-12)$$

where at this stage  $dS$  is the element of surface area of the surface  $f =$  constant. This relation is derived in Reference 2.5-2. Substituting Equation (2.5-12) into Equation (2.5-10) and integrating with respect to  $f$  results in the following equation:

$$4\pi p'(\vec{x}, t) = \frac{1}{c} \frac{\partial}{\partial t} \int_{f=0} \left[ \frac{\rho_0 c v_n + l_r}{r |1 - M_r|} \right]_{\gamma} dS + \int_{f=0} \left[ \frac{l_r}{r^2 |1 - M_r|} \right]_{\tau^*} dS \quad (2.5-13)$$

Here,  $M_r = v_j \hat{r}_j / c$ , where  $v$  is the local velocity of points on the blade surface itself. Since  $M_r < 1$  for all points on a subsonic blade, the absolute value about  $1 - M_r$  will be dropped in the following formulas.

Equation (2.5-13) has been used for high speed propeller noise calculations by Farassat (Ref. 2.5-4). For subsonic propellers, since  $M_r < 1$ , the time derivative is taken inside the integral since the following terms are functions of time in the first integral of Equation (2.5-13):

$$l_j; \frac{1}{r(1-M_r)}; \frac{\hat{r}_j}{r(1-M_r)}.$$

The chain rule of differentiation is then used to differentiate the integrand of the first integral in Equation (2.5-13):

$$\frac{\partial}{\partial t} = \frac{\partial \tau^*}{\partial t} \frac{\partial}{\partial \tau^*} \quad (2.5-14)$$

From Equation (2.5-11), by differentiating with respect to  $t$ , the following is obtained:

$$\frac{\partial \tau^*}{\partial t} = \frac{1}{1-M_r} \quad (2.5-15)$$

Therefore, from Equation (2.5-14)

$$\frac{\partial}{\partial t} = \frac{1}{1-M_r} \frac{\partial}{\partial \tau^*} \quad (2.5-16)$$

The following relations are thus established:

$$\frac{\partial r}{\partial \tau^*} = -v_r \quad (2.5-17)$$

$$\frac{\partial \hat{r}_i}{\partial \tau^*} = \frac{\hat{r}_i v_r - v_i}{r} \quad (2.5-18)$$

$$\frac{\partial M_r}{\partial \tau^*} = \frac{1}{cr} \left[ r_i \frac{\partial v_i}{\partial \tau} + v_r^2 - v^2 \right] \quad (2.5-19)$$

Using Equation (2.5-16) through (2.5-19), the time derivative in Equation (2.5-13) is carried out explicitly to give:

$$\begin{aligned}
 4\pi p_L'(\vec{x}, t) &= \frac{1}{c} \int_{f=0} \left[ \frac{\dot{l}_i \hat{r}_i}{r(1-M_r)^2} \right]_{\tau^*} dS \\
 &+ \int_{f=0} \left[ \frac{l_r - l_i M_i}{r^2(1-M_r)^2} \right]_{\tau^*} dS \\
 &+ \frac{1}{c} \int_{f=0} \left[ \frac{l_r (r \dot{M}_i \hat{r}_i + c M_r - c M^2)}{r^2 (1-M_r)^3} \right]_{\tau^*} dS \quad (2.5-20a)
 \end{aligned}$$

$$4\pi p_T'(\vec{x}, t) = \int_{f=0} \left[ \frac{\rho_0 v_n (r \dot{M}_i \hat{r}_i + c M_r - c M^2)}{r^2 (1-M_r)^3} \right]_{\tau^*} dS \quad (2.5-20b)$$

and

$$p'(\vec{x}, t) = p_L'(\vec{x}, t) + p_T'(\vec{x}, t) \quad (2.5-20c)$$

where  $p_L'$  and  $p_T'$  denote the loading and thickness noise, respectively. Equations (2.5-20a) through (2.5-20c) are used when the sources on the full-blade surface are specified. They are related to an equation derived by Woan and Gregorek (Refs. 2.5-4 and 2.4-5).

The symbols  $M_i$  and  $\dot{M}_i$  are defined as follows:

$$M_i = \frac{v_i}{c} \quad (2.5-21)$$

$$\dot{M}_i = \frac{\dot{v}_i}{c} \quad (2.5-22)$$

Note that  $v_i$  and  $\dot{v}_i$  are the local velocity and acceleration of the blade with respect to the ground-fixed frame. It is assumed here that the propeller blades are rigid. It is important to note that  $\dot{l}_i$  is the rate of change of the vector  $l_i$  as seen from the ground-fixed frame. This means that even if  $l_i$  is steady in the blade-fixed frame,  $\dot{l}_i$  is not zero.

The force intensity vector  $l_i$  can be written as

$$l_i = p \hat{n}_i + \sigma \hat{t}_i \quad (2.5-23)$$

where  $p$  is the surface pressure defined as  $p_a - p_0$ . Here  $p_a$  is the absolute surface pressure and  $p_0$  is the ambient pressure in the undisturbed medium. The local tangential stress is denoted by  $\sigma$ . Also the unit tangent vector pointing toward the section leading edge is denoted by  $\hat{t}_i$ . From Equation (2.5-23), the rate of change of  $l_i$  is found to be

$$\dot{l}_i = \dot{p} \hat{n}_i + p \dot{\hat{n}}_i + \dot{\sigma} \hat{t}_i + \sigma \dot{\hat{t}}_i \quad (2.5-24)$$

It can be shown that

$$\dot{\hat{n}}_i = (\dot{\hat{n}} \times \hat{n})_i \quad (2.5-25a)$$

and

$$\dot{\hat{t}}_i = (\dot{\hat{t}} \times \hat{t})_i \quad (2.5-25b)$$

From Equations (2.5-24) and (2.5-25), the following results are obtained:

$$\dot{\hat{r}}_i = \dot{p} \cos \theta + \dot{\sigma} \cos \theta' + p \hat{r}_1 + \sigma \hat{r}_2 \quad (2.5-26a)$$

$$l_i \hat{r}_i = p \cos \theta + \sigma \cos \theta' \quad (2.5-26b)$$

and

$$l_i M_i = p M_n + \sigma M_t \quad (2.5-26c)$$

Note that  $\theta$  is the angle between  $\hat{n}$  and  $\hat{r}$ ,  $\theta'$  is the angle between  $\hat{t}_i$  and  $\hat{r}_i$ , and  $\hat{r}_1$  and  $\hat{r}_2$  are defined as follows:

$$\hat{r}_1 = \hat{r} \cdot (\hat{n} \times \hat{n}) \quad (2.5-27a)$$

$$\hat{r}_2 = \hat{r} \cdot (\hat{n} \times \hat{t}) \quad (2.5-27b)$$

Also  $M_t$  is defined as  $M_i \hat{t}_i / c$ . Equation (2.5-26) is substituted into Equation (2.5-20) and the results discretized for numerical calculation.

A useful approximation to Equation (2.5-20) will be written next. Assuming that the propeller blades are thin, the acoustic sources can be moved to the mean surface of the blades. The mean surface is defined to be that surface formed by generating the section mean curves along the leading edge curve. Let the surface be denoted by the equation  $f_m(\vec{x}, t) = 0$ . The FW-H equation becomes

$$\begin{aligned} \frac{1}{c^2} \frac{\partial p'}{\partial t^2} - \nabla^2 p' &= 2 \frac{\partial}{\partial t} \left[ \rho_0 \bar{v}_n |\nabla f_m| \delta(f_m) \right] \\ &+ \frac{\partial}{\partial x_i} \left[ \Delta p n_{im} |\nabla f_m| \delta(f_m) \right] \end{aligned} \quad (2.5-28)$$

where  $\Delta p = p_{\text{lower}} - p_{\text{upper}}$  and  $n_{im}$  is the unit normal to  $f_m = 0$  pointing toward the upper side of the blade. Here  $\bar{v}_n$  is the local normal velocity of the blade on one side of the mean surface due to thickness distribution alone (i.e., the camber effect is ignored). Note that  $\bar{v}_n$  is calculated by evaluating  $v_n$  at each upper surface point and  $v_n$  at the corresponding point on the lower surface and then averaging. Furthermore, skin friction is neglected.

Following a procedure similar to that leading to Equation (2.5-20), we obtain an expression for  $p'(\vec{x}, t)$ :

$$4\pi p'_L(\vec{x}, t) = -\frac{1}{c} \int_{f_m=0} \left[ \frac{\Delta \dot{p} \cos \theta + \Delta p \Gamma_1}{r(1-M_r)^2} \right]_{\mathcal{T}^*} dS$$

$$- \int_{f_m=0} \left[ \frac{\Delta p (\cos \theta - M_n)}{r^2 (1-M_r)^2} \right]_{\mathcal{T}^*} dS$$

$$- \frac{1}{c} \int_{f_m=0} \left[ \frac{\Delta p \cos \theta (r \dot{M}_i \hat{r}_i + cM_r - cM^2)}{r^2 (1-M_r)^3} \right]_{\mathcal{T}^*} dS \quad (2.5-29a)$$

$$4\pi p'_T(\vec{x}, t) = \int_{f_m=0} \left[ \frac{2\rho_0 \bar{v}_n (r \dot{M}_i \hat{r}_i + cM_r - cM^2)}{r^2 (1-M_r)^3} \right]_{\mathcal{T}^*} dS \quad (2.5-29b)$$

and

$$p'(x, t) = p'_L(x, t) + p'_T(x, t) \quad (2.5-29c)$$

In Equation (2.5-29a),  $\theta$  is the angle between the radiation vector  $\hat{r}_i$  and  $n_{im}$ .

Two compact-source approximations for the subsonic propeller noise calculation will also be used. In the first approximation, each blade is

replaced by a single point source involving both the thickness and loading effects. The loading noise formula of Lawson and the thickness noise formula of Succi are used (Refs. 2.5-6 and 2.5-7). Let  $L_i$  be the net force by each blade on the fluid and  $\Psi$  be the net volume of each blade. Then the loading and thickness noise are calculated from:

$$4\pi p_L'(\vec{x}, t) = \left\{ \frac{1}{cr(1-M_r)^2} \left[ \hat{r}_i \dot{L}_i + \frac{\hat{r}_i L_i}{(1-M_r)} \left[ \dot{M}_r + \frac{c}{r} (1-M^2) \right] - \frac{c}{r} M_i L_i \right] \right\}_{\tau = \tau^*}$$

$$\equiv F_L(\vec{x}, t, R_E) \quad (2.5-30)$$

and

$$4\pi p_T'(\vec{x}, t) = \left\{ \frac{\rho_0 \Psi}{r(1-M_r)^4} \left[ (1-M_r) \left( \dot{M}_r - 3 \frac{c}{r} \dot{M} \cdot \dot{M} \right) + 3 \left( \frac{c}{r} M_r \right)^2 + 3 \left( \dot{M}_r - \frac{c}{r} M^2 \right)^2 + \frac{c}{r} \left( \dot{M}_r - \frac{c}{r} M^2 \right) \left( 1+4M_r + M_r^2 \right) \right] \right\}_{\tau = \tau^*}$$

$$\equiv F_T(\vec{x}, t, R_E) . \quad (2.5-31)$$

Here the sources are assumed to be located at some effective radial distance  $R_E$  (usually about 0.8 radius of the propeller). The effect of each blade is computed separately and summed to get the total noise.

A second compact-source formulation is obtained by assuming that each blade can be approximated by a line source (chordwise compactness assumption). In Equations (2.5-30) and (2.5-31), let  $L_1'(\xi_1)$  and  $A(\xi_1)$  be substituted for  $L_i$  and  $\Psi$ , respectively, and represent the force per unit distance and blade volume per unit distance. Here  $\xi_1$  is the radial



distance in the blade-fixed frame. The noise due to loading and thickness of the propeller is then given by

$$4\pi p'(x, t) = \int_0^{R_0} \left[ F_L'(\vec{x}, t, \xi_1) + F_T'(\vec{x}, t, \xi_1) \right] d\xi_1 \quad (2.5-32a)$$

where the integrands  $F_L'$  and  $F_T'$  are defined by

$$F_L'(\vec{x}, t, \xi_1) = \frac{1}{cr(1-M_r)^2} \left\{ \hat{r}_i L_i' + \frac{\hat{r}_i L_i'}{(1-M_r)} \left[ \dot{M}_r + \frac{c}{r} (1-M^2) \right] - \frac{c}{r} M_i L_i' \right\} \quad (2.5-32b)$$

and

$$F_T'(\vec{x}, t, \xi_1) = \frac{\rho_0 A(\xi_1)}{r(1-M_r)^4} \left\{ (1-M_r) \left[ \dot{M}_r - 3 \frac{c}{r} \dot{M} \cdot \dot{M} \right] + 3 \left( \frac{c}{r} M_r \right)^2 + 3 \left( \dot{M}_r - \frac{c}{r} M^2 \right)^2 + \frac{c}{r} \left( \dot{M}_r - \frac{c}{r} M^2 \right) \left[ 1 + 4M_r + M_r^2 \right] \right\} \quad (2.5-32c)$$

The symbols  $R_i$  and  $R_0$  in Equation (2.5-32a) represent the inner and the outer radii of the blade. The effect of several blades is taken into account by a method requiring a single blade calculation as discussed below.

#### 2.5.2.2 Frames of Reference

In this section the coordinate frames used in the acoustic calculations will be discussed. Basically there are two reference frames: the ground-fixed  $\vec{x}$ -frame and the blade-fixed  $\vec{\eta}$ -frame. The  $\vec{x}$ -frame which remains fixed to the undisturbed medium is set up as follows:

- The origin of this frame is at the propeller center at time  $t=0$ ,

- The propeller disk is in the  $x_1x_2$ -plane and the  $x_3$  axis coincides with the propeller axis, with positive direction in the flight direction.
- The  $x_1$ -axis is assumed to be orthogonal to  $x_3$ , upward and parallel to the plane of symmetry of the aircraft.
- The  $x_2$ -axis is defined in such a way that the  $\vec{x}$ -frame is right-handed (Figure 2.5-1).

The observer position is always specified in this frame as will be explained shortly.

The blade-fixed frame is set up as follows:

- The origin coincides with the origin of the  $\vec{x}$ -frame at  $t=0$ .
- The  $\eta_3$ -axis coincides with the  $x_3$ -axis.
- The  $\eta_2$ -axis lies on the pitch change axis of the blade.
- The  $\eta_1\eta_2$ -plane is normal to the propeller shaft.
- The  $\eta_1$ -axis is defined to make the  $\vec{\eta}$ -frame right-handed (Figure 2.5-2).

Note that the blade mean surface does not lie in the  $\eta_1\eta_2$ -plane but makes an angle equal to the geometric pitch angle at each radial position. Note also that the  $\vec{\eta}$ -frame rotates with angular velocity,  $\Omega$ .

In the acoustic formulas presented in the previous section, many operations involving dot products or cross products of two vectors are required. When the products involve blade normals or surface pressures, it is convenient to transform the components of these vectors to the  $\vec{\eta}$ -frame and then perform the operation. However, note that the source position  $\vec{y}$  always refers to a source point in the  $\vec{x}$ -frame with a known position  $\vec{\eta}$  in the blade-fixed frame. The quantities  $\vec{r} = \vec{x} - \vec{y}$  and  $\hat{r} = \vec{r}/r$  are calculated in the  $\vec{x}$ -frame, and then the vector components are calculated in the  $\vec{\eta}$ -frame. Note that for any variable which has a velocity term, such as  $M_r = M_i \hat{r}_i$  and  $v_n = \vec{v} \cdot \hat{n}$ , the velocity is with respect to the  $\vec{x}$ -frame. In other words

$$v_i = \frac{\partial y_i}{\partial \tau} \quad (2.5-33)$$

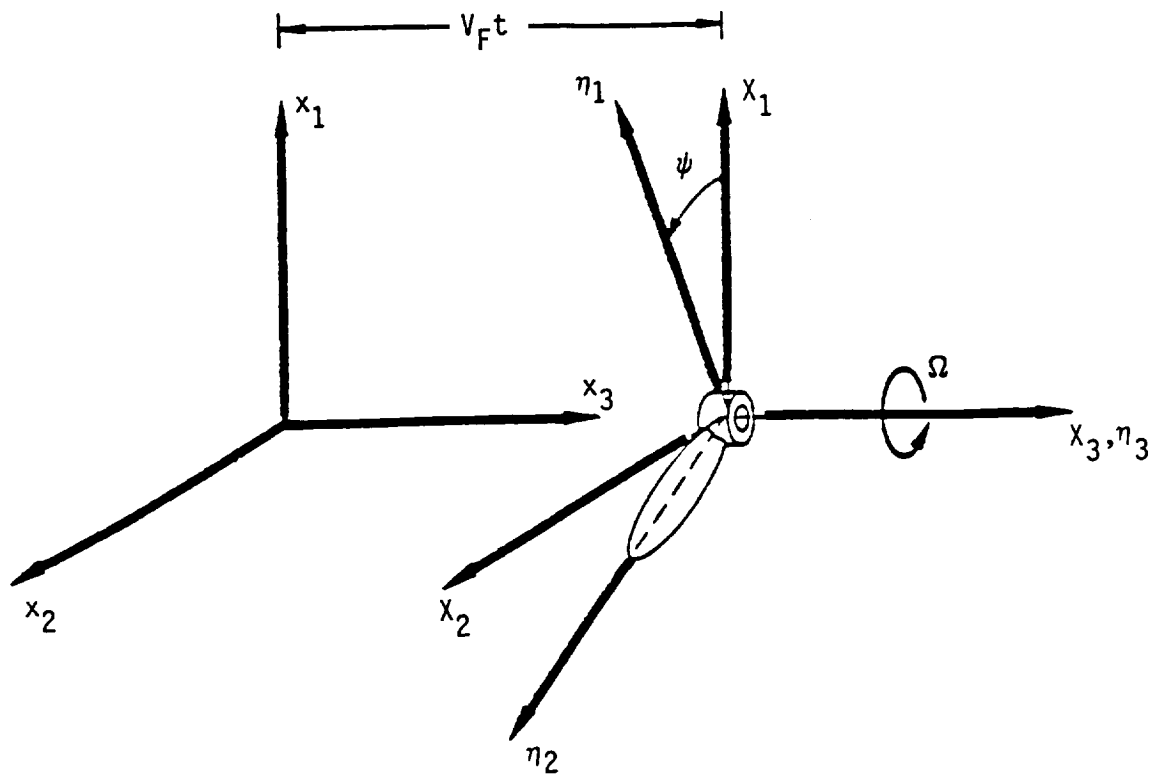


Figure 2.5-1. Transformation of Coordinates

The components of this vector must then be calculated in the  $\vec{\eta}$ -frame to get  $M_r$  and  $v_n$ .

Another frame of reference which is used conceptually is the aircraft-fixed nonrotating  $\vec{X}$ -frame. This frame coincides with the ground-fixed  $\vec{x}$ -frame at  $t=0$ . The acoustic pressure  $p'(\vec{X},t)$  for an observer in the aircraft-fixed frame with the position vector  $\vec{X}$  is found as follows. Since both  $\vec{X}$  and  $t$  are known, the relative position of the ground-fixed frame and the aircraft-fixed-frame is known is calculated. From this, the observer position  $\vec{x}$  in the ground-fixed frame. Using this  $\vec{x}$  and  $t$ ,  $p'(\vec{x},t)$ , which is the same as the acoustic pressure at position  $\vec{X}$  in aircraft-fixed frame and at time  $t$  is calculated. Thus, the  $\vec{X}$ -frame itself is never used in acoustic calculations, even though the pressure signal that results is  $p'(\vec{X},t)$ .

The transformation from one frame to the other at time  $t$  can be written in matrix notation:

$$\begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{bmatrix} \quad (2.5-34a)$$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ V_F t \end{bmatrix} \quad (2.5-34b)$$

where  $(\eta_1, \eta_2, \eta_3)$  is a point in the rotating  $\vec{\eta}$ -frame,  $(X_1, X_2, X_3)$  is the corresponding point in the moving  $X$  frame,  $(x_1, x_2, x_3)$  is the corresponding point in the ground-fixed frame, and  $\psi(=\Omega t)$  is the angle between the  $X_1$  and  $\eta_1$  axes.

A final coordinate transformation is necessary to account for changes in the root pitch setting. This rotation through a small angle,  $\theta_0$ , is

applied to the blade surface coordinates which are calculated by the geometry generator module.

$$\vec{\eta}(\xi_1, \xi_2) = \begin{bmatrix} \cos\theta_0 & 0 & \sin\theta_0 \\ 0 & 1 & 0 \\ -\sin\theta_0 & 0 & \cos\theta_0 \end{bmatrix} \begin{bmatrix} \eta_1'(\xi_1, \xi_2) \\ \xi_1 \\ \eta_2'(\xi_1, \xi_2) \end{bmatrix} \quad (2.5-35)$$

Whenever the  $\vec{\eta}$ -frame or pitch axis coordinate system is mentioned herein, it is assumed that the above corrective transformation has occurred. Often the Geometry Generator Module and propeller noise module will be executed without change to the collective pitch setting. Thus,  $\theta_0=0$  and the last transformation is unnecessary.

### 2.5.2.3 Computational Strategy

A pressure time history of a multibladed propeller is calculated by summing the acoustic pressures of each blade as a function of time. The signals are periodic because the observer position is a fixed distance from the hub at all times. Thus, the signal produced by one revolution of a single blade can be used to determine the time history and spectrum of the propeller. The general computational strategy is outlined below. Details concerning the different methods for calculating  $p'(X,t)$  are contained in later sections.

The steps involved in calculating the acoustic time signature for a single blade are as follows. Assume that the initial pitch change axis of the blade is aligned with the  $\vec{\lambda}_2$ -axis. That is, assume  $t = \psi = 0$  and the  $\vec{x}$ -frame,  $\vec{\lambda}$ -frame and  $\vec{\eta}$ -frame coincide. Divide the time required for one full revolution of the blade into even time increments such that

$$t_i = \frac{(i-1) 2\pi}{N_t \Omega} \quad i = 1, 2, \dots, N$$

Likewise, represent the blade by a set of source points,  $\vec{\eta}(j)$ , where the number and location of the source points depends on which method for

calculating noise is selected. Then, for each time,  $t_j$ , and each source point,  $\vec{\eta}(j)$ , determine the source,  $\vec{y}$ , and observer,  $\vec{x}$ , locations in the ground-fixed frame. Using these, solve the retarded time equation and calculate distances and velocities at emission time. Next, calculate vector components in the rotating  $\vec{\eta}$ -frame and calculate the influence of this source point on the total noise. Finally, accumulate the information for each time and source point into a complete time history.

The steps involved in producing a time history and spectrum for the propeller are as follows:

- Tabulate the time history for one blade so that intermediate values can be obtained by interpolation.
- Decide how many evenly spaced time points are required by the FFT procedure. If  $N_S$  is the highest harmonic number of interest and  $N_t$  the number of time points, find an integer  $m$  such that  $2^m$  is at least as large as  $N_t \geq 2 N_S$ ;
- Calculate the total acoustic pressure-time history by adding the contribution from each blade;
- Finally, perform an FFT to get the pressure spectrum.

The above process can be stated mathematically as follows. The time required for one revolution of the blade is divided equally into  $2^m$  points such that

$$t_k = \frac{(k-1) 2\pi}{(2^m) \Omega} \quad k = 1, 2, \dots, 2^m$$

$$m = \text{INT} (\log_2 N_t) + 1 \quad (2.5-36)$$

where the INT function turns a real number into the largest integer not exceeding the real number.

The acoustic signal generated by a propeller is just the sum of the pressures produced by each blade. The pressure produced at time  $t$  by blade  $b$  which had an initial angle  $\psi_b = \psi_0 + 2\pi (b - 1)/N_b$  relative to the  $\vec{\chi}_2$

axis, can be found by interpolating into the single blade pressure table for  $p'(\vec{x}, t + \psi_b/\Omega)$ . Thus, the total acoustic signal is

$$p_T(\vec{X}, t_k) = \sum_{b=1}^{N_b} p'(\vec{X}, t_k + \psi_b/\Omega), \quad (k = 1, 2, \dots, 2^m) \quad (2.5-37)$$

where  $\vec{X}$  is the observer position in the aircraft fixed frame,  $N_b$  is the number of blades, and  $\psi_b$  is the initial position of each blade. The spectrum associated with this time history is calculated by using a standard FFT routine. Section 2.5.2.9 contains further details.

#### 2.5.2.4 Retarded Time Equation

This section describes a method for solving the retarded time equation,  $g = \tau - t + r/c = 0$ . As noted above, the root  $\tau = \tau^*$  of this equation must be found for each source point used to represent the blade and for each observer time,  $t$ . Since the roots of the equation are calculated repeatedly, it is imperative to solve the equation in an efficient way. Details of the solution technique are reserved for Section 2.5.2.8.

The problem can be stated in terms of arbitrary receiver location  $\vec{x}$ , and source location,  $\vec{y}$  defined in the ground-fixed coordinate system at time  $t$ . As described in Reference 2.5-4, the equation can be written

$$\begin{aligned} r^2 = x_3^2 + d_x^2 + d_y^2 - 2d_x d_y \cos(\Omega\tau + \psi_y - \psi_x) \\ + V_F^2 \tau^2 - 2x V_F \tau = c^2(\tau - t)^2 \end{aligned} \quad (2.5-38)$$

where  $d_x$  and  $d_y$  are the radial distances between the  $x_3$  axis and the observer and source points, and  $\psi_x$ ,  $\psi_y$  are the observer and source angles of rotation at  $t = 0$ . For instance, if the source point is located on the pitch change axis then  $d_y = \eta_2$  and  $\psi_y = 0$  (Figure 2.5-2).

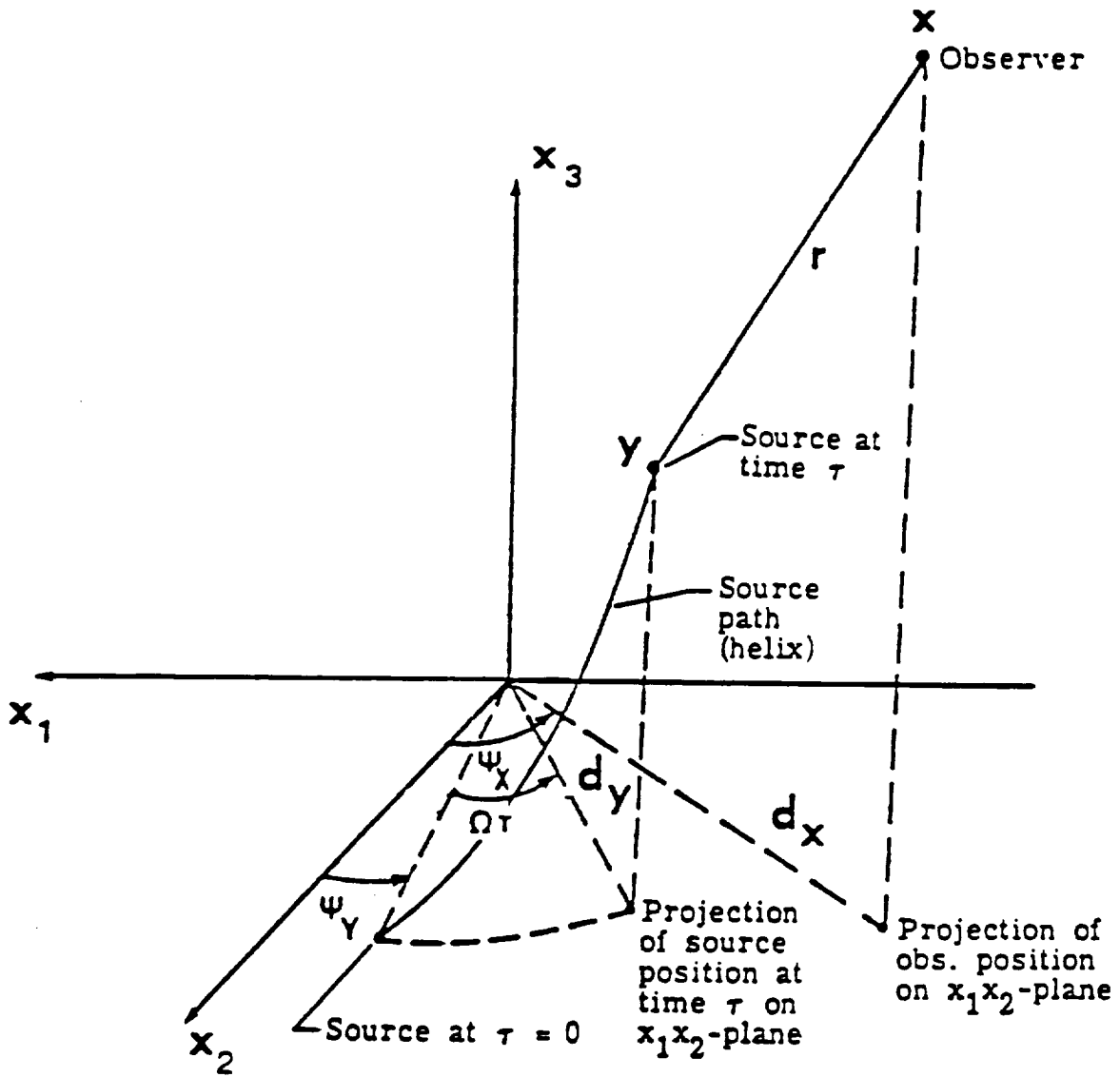


Figure 2.5-2. Geometry of Source and Observer Position Used for Calculation of Emission Time



After some manipulation,

$$r^2 = (x_3 - v_F t)^2 + d_x^2 + d_y^2 = 2d_x d_y \cos [\Omega(\tau - t) + \psi_y - (\psi_x - \Omega t)]$$

$$+ v_F^2(\tau - t)^2 - 2v_F(x_3 - v_F t)(\tau - t) = c^2(\tau - t)^2 \quad (2.5-39)$$

Let  $\phi = \Omega(\tau - t)$ , then Equation (2.5-39) is of the form

$$A\phi^2 + B\phi + C + \cos(\phi + D) = 0 \quad (2.5-40)$$

where

$$A = [c^2 - v_F^2] / [2\Omega^2 d_x d_y] \quad A > 0$$

$$B = v_F(x_3 - v_F t) / (\Omega d_x d_y)$$

$$C = - [(x_3 - v_F t)^2 + d_x^2 + d_y^2] / (2d_x d_y)$$

$$D = \psi_y - \psi_x + \Omega t$$

Based on the transformation given in Equation (2.5-34b), the term  $(x_3 - v_F t)$  can be replaced by the constant  $X_3$  for all times. Consequently the coefficients A, B, C in Equation (2.5-40) change only if source position changes. Thus, the solution to Equation (2.5-40) for one source point and source time will be a good initial guess for the same source point at a slightly later time.

Equation (2.5-40) can be solved very efficiently using Newton's method. Only the root such that  $(\tau - t) < 0$  is physically possible. However, with a good initial guess, Newton's method will converge to the proper root very quickly.

### 2.5.2.5 Full Blade Formulation

This section discusses the calculation of the periodic acoustic pressure signal for a single rotating blade with subsonic tip speed. The method described requires a detailed knowledge of the blade geometry and blade surface pressures. The following two sections discuss simplifying approximations to the full blade formulation which are used in the absence of full blade information.

The acoustic pressure at time  $t$  for an observer at point  $\vec{x}$  is given in Equation (2.5-20) and repeated here.

$$\begin{aligned}
 4\pi p_L'(\vec{x}, t) = & \frac{1}{c} \int_{f=0} \left[ \frac{\dot{\hat{r}}_i}{r(1-M_r)^2} \right]_{\tau^*} dS \\
 & + \int_{f=0} \left[ \frac{l_r - l_i M_i}{r^2(1-M_r)^2} \right]_{\tau^*} dS \\
 & + \frac{1}{c} \int_{f=0} \left[ \frac{l_r (r \dot{M}_i \hat{r}_i + c M_r - c M^2)}{r^2(1-M_r)^3} \right]_{\tau^*} dS, \quad (2.5-41a)
 \end{aligned}$$

$$4\pi p_T'(\vec{x}, t) = \int_{f=0} \left[ \frac{\rho_0 v_n (r \dot{M}_i \hat{r}_i + c M_r - c M^2)}{r^2 (1-M_r)^3} \right]_{\tau^*} dS \quad (2.5-41b)$$

with

$$4\pi p'(\vec{x}, t) = p_L'(\vec{x}, t) + p_T'(\vec{x}, t) \quad (2.5-41c)$$

Using symbols  $F_{L1}$ ,  $F_{L2}$ , and  $F_{L3}$  to stand for the integrands which are evaluated at retarded time  $\tau^*$ , the above equation is written in short form,

$$4\pi p_L' = \frac{1}{c} \int_{f=0} F_{L1}(\vec{\eta}) dS + \int_{f=0} F_{L2}(\vec{\eta}) dS + \frac{1}{c} \int_{f=0} F_{L3}(\vec{\eta}) dS \quad (2.5-42a)$$

$$4\pi p_T' = \int_{f=0} F_T(\vec{\eta}) dS \quad (2.5-42b)$$

Using the Geometry Generator Module, the surface of a propeller blade is described by a set of points  $\vec{\eta}_j$ . The three components of each point  $\vec{\eta}_j$  are tabulated with respect to spanwise and chordwise station parameters  $\xi_1$  and  $\xi_2$  (Figure 2.5-3). Thus each term in Equations (2.5-42) may be replaced by an equivalent double integral over  $\xi_1$  and  $\xi_2$ . For instance,

$$\int_{f=0} F_{L1}(\vec{\eta}) dS \equiv \int_0^1 \int_0^{2\pi} F_{L1}(\vec{\eta}(\xi_1, \xi_2)) J d\xi_2 d\xi_1 \quad (2.5-43)$$

where  $J$  is the Jacobian of the transformation,

$$J = \left| \frac{\partial \vec{\eta}}{\partial \xi_1} \times \frac{\partial \vec{\eta}}{\partial \xi_2} \right| \quad (2.5-44)$$

Equation (2.5-43) is evaluated by using a compound rule. Thus,

$$\int_0^1 \int_0^{2\pi} F_{L1} J d\xi_2 d\xi_1 = \int_0^1 G_{L1}(\xi_1) d\xi_1 \quad (2.5-45)$$

where

$$G_{L1}(\xi_1) = \int_0^{2\pi} F_{L1}(\xi_1, \xi_2) J(\xi_1, \xi_2) d\xi_2 \quad (2.5-46)$$

Each of the integrals over a single parameter is evaluated using a standard procedure based on cubic splines.

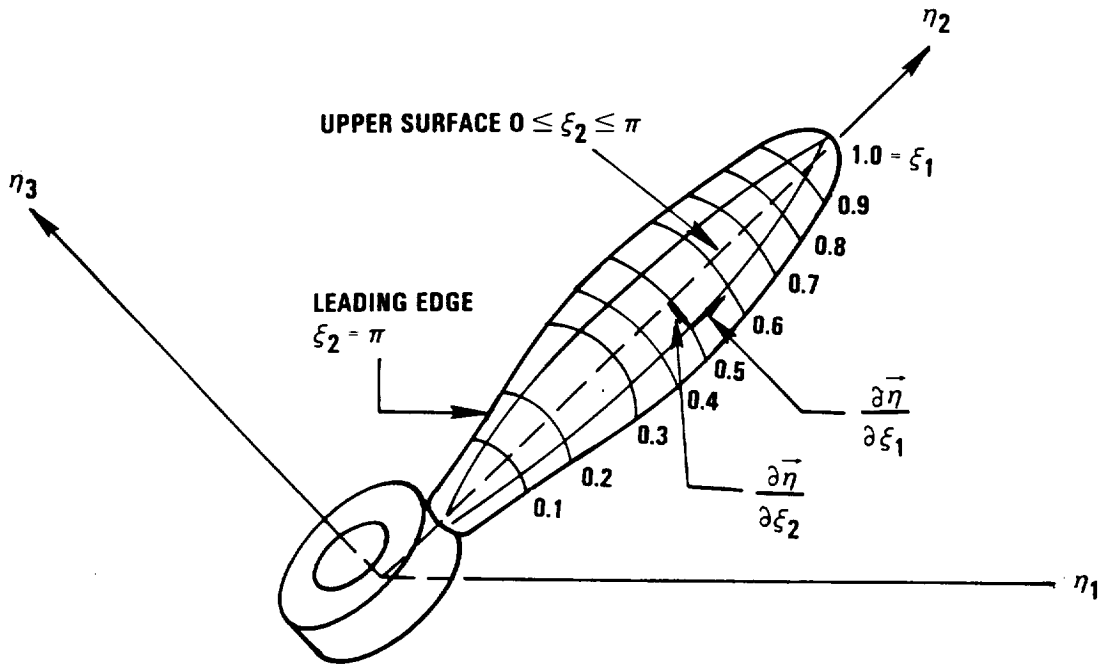


Figure 2.5-3. Blade Surface Coordinate System

Conceptually, the process of evaluating Equation (2.5-41) to get  $p'(\vec{x}, t)$  involves calculating the values of each integrand at a set of blade surface points  $\vec{\eta}(j)$  and accumulating them in a way that gives a sufficiently good approximation to the integral.

#### 2.5.2.6 Mean Surface Approximation

In this section, approximation formulas for noise calculations based on the assumption of thin airfoils are presented. The mean blade surface is defined to be that surface generated by the totality of mean section curves. The calculation procedure is derived from Equation (2.5-29) which are repeated here for convenience:

$$\begin{aligned}
 4\pi p'_L(\vec{x}, t) = & -\frac{1}{c} \int_{f_m=0} \left[ \frac{\Delta \dot{p} \cos \theta + \Delta p \tau_1}{r(1-M_r)^2} \right]_{\tau^*} dS \\
 & - \int_{f_m=0} \left[ \frac{\Delta p (\cos \theta - M_n)}{r^2 (1-M_r)^2} \right]_{\tau^*} dS \\
 & - \frac{1}{c} \int_{f_m=0} \left[ \frac{\Delta p \cos \theta (r \dot{M}_i \hat{r}_i + c M_r - c M^2)}{r^2 (1-M_r)^3} \right]_{\tau^*} dS, \quad (2.5-47a)
 \end{aligned}$$

$$4\pi p'_T(\vec{x}, t) = \int_{f=0} \left[ \frac{2p_0 \bar{v}_n (r \dot{M}_i \hat{r}_i + c M_r - c M^2)}{r^2 (1-M_r)^3} \right]_{\tau^*} dS \quad (2.5-47b)$$

and

$$p'(\vec{x}, t) = p'_L(\vec{x}, t) + p'_T(\vec{x}, t) \quad (2.5-47c)$$

We deal first with the thickness noise integral, Equation (2.5-47b), and derive equations for the various terms of the integrand as functions of known quantities.

### Coordinates of Mean Surface

For a given radial distance,  $\xi_1$ , allow  $\xi_2$  to vary in the range  $0 \leq \xi_2 \leq \pi$  and define a mean section line as

$$\bar{\eta}_i(\xi_1, \xi_2) = \frac{1}{2} [\eta_i(\xi_1, \xi_2) + \eta_i(\xi_1, 2\pi - \xi_2)], \quad (i = 1, 2, 3) \quad (2.5-48)$$

### Differential Element of Mean Surface

In terms of the independent variables  $\xi_1$  and  $\xi_2$ , the differential area element,  $dS$ , is given by:

$$dS = \bar{J}(\xi_1, \xi_2) d\xi_1 d\xi_2 = \sqrt{EG - F^2} d\xi_1 d\xi_2 \quad (2.5-49)$$

The symbols E, F, and G are written in terms of the mean surface coordinates as follows:

$$E = \left[ \frac{\partial \bar{\eta}_1}{\partial \xi_1} \right]^2 + \left[ \frac{\partial \bar{\eta}_2}{\partial \xi_1} \right]^2 + \left[ \frac{\partial \bar{\eta}_3}{\partial \xi_1} \right]^2 \quad (2.5-50)$$

$$F = \frac{\partial \bar{\eta}_1}{\partial \xi_1} \frac{\partial \bar{\eta}_1}{\partial \xi_2} + \frac{\partial \bar{\eta}_2}{\partial \xi_1} \frac{\partial \bar{\eta}_2}{\partial \xi_2} + \frac{\partial \bar{\eta}_3}{\partial \xi_1} \frac{\partial \bar{\eta}_3}{\partial \xi_2} \quad (2.5-51)$$

and

$$G = \left[ \frac{\partial \bar{\eta}_1}{\partial \xi_2} \right]^2 + \left[ \frac{\partial \bar{\eta}_2}{\partial \xi_2} \right]^2 + \left[ \frac{\partial \bar{\eta}_3}{\partial \xi_2} \right]^2 \quad (2.5-52)$$

We recall that  $\eta_2 = \xi_1$  and thus  $\bar{\eta}_2$  is independent of  $\xi_2$ . Equations (2.5-50) and (2.5-51) therefore simplify to

$$E = \left( \frac{\partial \bar{\eta}_1}{\partial \xi_1} \right)^2 + \left( \frac{\partial \bar{\eta}_3}{\partial \xi_1} \right)^2 + 1 \quad (2.5-53)$$

and

$$F = \frac{\partial \bar{\eta}_1}{\partial \xi_1} \frac{\partial \bar{\eta}_1}{\partial \xi_2} + \frac{\partial \bar{\eta}_3}{\partial \xi_1} \frac{\partial \bar{\eta}_3}{\partial \xi_2} \quad (2.5-54)$$

### Velocities and Mach Numbers

The position of a point on the blade is obtained by using the transformation defined by Equations (2.5-34)

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} \cos \Omega t & -\sin \Omega t & 0 \\ \sin \Omega t & \cos \Omega t & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ V_F t \end{bmatrix}. \quad (2.5-55)$$

The velocity of such a point is therefore expressed as

$$\vec{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \\ \dot{y}_3 \end{bmatrix} = \Omega \begin{bmatrix} -\sin \Omega t & -\cos \Omega t & 0 \\ \cos \Omega t & -\sin \Omega t & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ V_F \end{bmatrix}. \quad (2.5-56)$$

The position of a source point on the mean surface is, by analogy with Equation (2.5-55), given as follows:

$$\begin{bmatrix} \bar{y}_1 \\ \bar{y}_2 \\ \bar{y}_3 \end{bmatrix} = \begin{bmatrix} \cos \Omega t & -\sin \Omega t & 0 \\ \sin \Omega t & \cos \Omega t & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \bar{\eta}_1 \\ \bar{\eta}_2 \\ \bar{\eta}_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ v_F t \end{bmatrix} \quad (2.5-57)$$

The unit normal to the mean surface is

$$\hat{n} = \frac{\partial \bar{\eta}}{\partial \xi_1} \times \frac{\partial \bar{\eta}}{\partial \xi_2} / \left[ \frac{\partial \bar{\eta}}{\partial \xi_1} \times \frac{\partial \bar{\eta}}{\partial \xi_2} \right] \quad (2.5-58)$$

and the unit vector in the radiation direction from a point on the mean surface is written as

$$\hat{r} = (\vec{x} - \vec{y}) / |\vec{x} - \vec{y}|. \quad (2.5-59)$$

By analogy with Equation (2.5-56), the velocity of a point on the mean surface is represented by

$$\vec{v} = \Omega \begin{bmatrix} -\sin \Omega t & -\cos \Omega t & 0 \\ \cos \Omega t & -\sin \Omega t & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{\eta}_1 \\ \bar{\eta}_2 \\ \bar{\eta}_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ v_F \end{bmatrix} \quad (2.5-60)$$

The acceleration therefore becomes

$$\frac{d\vec{v}}{dt} = \Omega^2 \begin{bmatrix} -\cos \Omega t & \sin \Omega t & 0 \\ -\sin \Omega t & -\cos \Omega t & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{\eta}_1 \\ \bar{\eta}_2 \\ \bar{\eta}_3 \end{bmatrix} \quad (2.5-61)$$



Hence the local Mach number and its time rate-of-change are given by:

$$\begin{bmatrix} M_1 \\ M_2 \\ M_3 \end{bmatrix} = \frac{\Omega}{c} \begin{bmatrix} -\sin \Omega t & -\cos \Omega t & 0 \\ \cos \Omega t & -\sin \Omega t & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{bmatrix} + \frac{1}{c} \begin{bmatrix} 0 \\ 0 \\ v_F \end{bmatrix} \quad (2.5-62)$$

and

$$\begin{bmatrix} \dot{M}_1 \\ \dot{M}_2 \\ \dot{M}_3 \end{bmatrix} = \frac{\Omega^2}{c} \begin{bmatrix} -\cos \Omega t & -\sin \Omega t & 0 \\ -\sin \Omega t & -\cos \Omega t & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{\eta}_1 \\ \bar{\eta}_2 \\ \bar{\eta}_3 \end{bmatrix}. \quad (2.5-63)$$

The mean normal velocity and radiation Mach number are written as:

$$\bar{v}_n = \frac{1}{2} \left[ \vec{v}(\xi_1, \xi_2) \cdot \hat{n}(\xi_1, \xi_2) + \vec{v}(\xi_1, 2\pi - \xi_2) \cdot \hat{n}(\xi_1, 2\pi - \xi_2) \right] \quad (2.5-64)$$

and

$$M = \vec{M} \cdot \left( \frac{\vec{x} - \vec{y}}{|\vec{x} - \vec{y}|} \right). \quad (2.5-65)$$

#### Miscellaneous Integrand Terms

The differential pressure,  $\Delta p$ , is defined to be:

$$\Delta p = p(\xi_1, \xi_2) - p(\xi_1, 2\pi - \xi_2). \quad (2.5-66)$$

The radiation angle is given by

$$\cos\theta = \hat{n} \cdot \hat{r}, \quad (2.5-67)$$

and the symbol  $r_1$  is defined to be

$$r = \left[ \hat{n} \times \hat{n} \right] \cdot \left[ \hat{x} - \frac{\hat{t}}{y} \right]. \quad (2.5-68)$$

We summarize by expressing Equation (2.5-47) as double integrals in the variables  $\xi_1$  and  $\xi_2$ . The equations are

$$4\pi p_L^i(x, t) = -\frac{1}{c} \int_0^\pi \int_0^1 \left[ \frac{\Delta \dot{p} \cos\theta + \Delta p r_1^i}{r(1-M_r)^2} \right]_{r^*} J(\xi_1, \xi_2) d\xi_1 d\xi_2$$

$$-\int_0^\pi \int_0^1 \left[ \frac{\Delta p (\cos\theta - M_n)}{r^2 (1-M_r)^2} \right]_{r^*} J(\xi_1, \xi_2) d\xi_1 d\xi_2$$

$$-\frac{1}{c} \int_0^\pi \int_0^1 \left[ \frac{\Delta p \cos\theta \left[ r_1^i \hat{r}_1 + cM_r - cM^2 \right]}{r^2 (1-M_r)^3} \right]_{r^*} J(\xi_1, \xi_2) d\xi_1 d\xi_2 \quad (2.5-69)$$

$$4\pi p_L^i(\hat{x}, t) = \int_0^\pi \int_0^1 \left[ \frac{2\rho_0 \bar{v}_n \left[ r_1^i \hat{r}_1 + cM_r + cM^2 \right]}{r(1-M_r)^2} \right]_{r^*} J(\xi_1, \xi_2) d\xi_1 d\xi_2 \quad (2.5-70)$$

where

$$J = \sqrt{EG - F^2}$$

with E, G, and F having the definitions given by Equations (2.5-50) and (2.5-51).

### 2.5.2.7 Compact Chord Approximation

In this section, approximation formulas based on the assumption of a line source are presented. The basis for the calculation procedure is the set of Equation (2.5-32) which we recall for our use

$$4\pi p'(\vec{x}, t) = \int_{R_i}^{R_o} \left[ F_L'(\vec{x}, t, \xi_1) + F_T'(\vec{x}, t, \xi_1) \right] d\xi_1 \quad (2.5-71)$$

where

$$F_L'(\vec{x}, t, \xi_1) = \frac{1}{cr(1-M_r)^2} \left\{ \hat{r}_i \dot{L}_i + \frac{\hat{r}_i L_i}{(1-M_r)} \left[ \dot{M}_r + \frac{c}{r} (1-M^2) \right] - \frac{c}{r} M_i L_i \right\} \quad (2.5-72)$$

and

$$F_T'(\vec{x}, t, \xi_1) = \frac{\rho_o A(\xi_1)}{r(1-M_r)^4} \left\{ (1-M_r) \left[ \dot{M}_r - 3 \frac{c}{r} \dot{M} \cdot \dot{M} \right] + 3 \left( \frac{c}{r} M_r \right)^2 + 3 \left[ \dot{M}_r - \frac{c}{r} M^2 \right]^2 + \frac{c}{r} \left[ \dot{M}_r - \frac{c}{r} M^2 \right] \left[ 1 + 4M_r + M_r^2 \right] \right\}. \quad (2.5-73)$$

Considering Equation (2.5-73), we observe that the quantities  $\vec{M}$ ,  $\dot{\vec{M}}$ ,  $M_r$ , and  $r$  are computed as in Equations (2.5-55) through (2.5-64) of the mean-surface approximation section. We take as our line source the leading edge curve of the propeller, so that all positions,  $\vec{r}$ , are on the leading edge curve. The quantity  $A(\xi_1)$  which represents the blade volume per unit

radial distance is equivalent to and numerically equal to the cross-sectional area of a given blade station. This quantity is calculated as a one-dimensional integral by using Stoke's Theorem. The value is

$$A(\xi_1) = \int_0^{2\pi} \eta_1(\xi_1, \xi_2) \frac{\partial \eta_3(\xi_1, \xi_2)}{\partial \xi_2} d\xi_2 \quad (2.5-74)$$

Next we examine the loading given by Equation (2.5-72). The key quantities to be calculated in this expression are  $\vec{L}'$  and  $d\vec{L}'/dt$  which represent the loading per unit length and its time rate-of-change. The loading per unit length is given by

$$\begin{bmatrix} L'_1 \\ L'_2 \\ L' \end{bmatrix} = \begin{bmatrix} -\cos \Omega t F_\psi(\xi_1, t) \\ -\sin \Omega t F_\psi(\xi_1, t) \\ F(\xi_1, t) \end{bmatrix}, \quad (2.5-75)$$

where  $F_\psi$  and  $F_z$  are provided by the loading module. The time rate-of-change of  $\vec{L}'$  becomes

$$\frac{d}{dt} \begin{bmatrix} L'_1 \\ L'_2 \\ L'_3 \end{bmatrix} = \begin{bmatrix} -\Omega \sin \Omega t F_\psi - \cos \Omega t \frac{\partial F_\psi}{\partial t} \\ -\Omega \cos \Omega t F_\psi - \sin \Omega t \frac{\partial F_\psi}{\partial t} \\ \frac{\partial F_z}{\partial t} \end{bmatrix} \quad (2.5-76)$$

The quantities  $\partial F_\psi/\partial t$  and  $\partial F_z/\partial t$  must be calculated as needed.

#### 2.5.2.8 Point Source Approximation

In this section, approximation formulas based on the assumption of a single noise source representing each blade are presented. The basis for

the calculation procedure is Equation (2.5-30) and (2.5-31) which we now recall

$$4\pi p_L'(\vec{x}, t) = \left\{ \frac{1}{cr(1-M_r)^2} \left[ \hat{r}_i \dot{L}_i + \frac{\hat{r}_i L_i}{(1-M_r)} \left[ \dot{M}_r + \frac{c}{r} (1-M^2) \right] - \frac{c}{r} M_i L_i \right] \right\} \Big|_{\tau = \tau^*} \quad (2.5-77)$$

and

$$4\pi p_T'(\vec{x}, t) = \left\{ \frac{p_0 \Psi}{r(1-M_r)^4} \left[ (1-M_r) (\dot{M}_r - 3 \frac{c}{r} \dot{M} \cdot \dot{M}) + 3 \left( \frac{c}{r} M_r \right)^2 + 3 \left( \dot{M}_r - \frac{c}{r} M^2 \right)^2 + \frac{c}{r} \left( \dot{M}_r - \frac{c}{r} M^2 \right) (1+4M_r + M_r^2) \right] \right\} \Big|_{\tau = \tau^*} \quad (2.5-78)$$

The local quantities such as  $r$ ,  $M$ ,  $M_r$ , etc., are evaluated at some effective radial distance,  $\xi_1 = R_E$ , usually about 0.8 of the propeller radius. We now derive explicit expressions for the various terms on the right-hand side of Equations (2.5-77) and (2.5-78). We begin with Equation (2.5-77); the effective blade load is given by

$$\begin{bmatrix} L_1 \\ L_2 \\ L_3 \end{bmatrix} = \begin{bmatrix} -H \cos \Omega t \\ -H \sin \Omega t \\ T \end{bmatrix}, \quad (2.5-79)$$

where  $H$  and  $T$  are the azimuthal force and net thrust from a single blade and are provided by the loading module. The time rate-of-change of  $\vec{L}$  becomes

$$\begin{bmatrix} \dot{L}_1 \\ \dot{L}_2 \\ \dot{L}_3 \end{bmatrix} = \begin{bmatrix} -\Omega H \sin \Omega t & -\dot{H} \cos \Omega t \\ -\Omega H \cos \Omega t & -\dot{H} \sin \Omega t \\ \dot{T} \end{bmatrix}. \quad (2.5-80)$$

The quantities  $\dot{R}$  and  $\dot{t}$  must be calculated as needed. If it is assumed that the effective source point is located on the leading edge at the effective radius,  $R_E$ , then the local Mach number at this point is given by

$$\begin{bmatrix} M_1 \\ M_2 \\ M_3 \end{bmatrix} = \frac{\Omega}{c} \begin{bmatrix} -\sin \Omega t & -\cos \Omega t & 0 \\ \cos \Omega t & -\sin \Omega t & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \eta_{1\ell}(R_E) \\ R_E \\ \eta_{3\ell}(R_E) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ V_F/c \end{bmatrix} \quad (2.5-81)$$

The first and second time rates-of-change become:

$$\begin{bmatrix} \dot{M}_1 \\ \dot{M}_2 \\ \dot{M}_3 \end{bmatrix} = \frac{\Omega^2}{c} \begin{bmatrix} -\cos \Omega t & \sin \Omega t & 0 \\ -\sin \Omega t & -\cos \Omega t & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \eta_{1\ell}(R_E) \\ R_E \\ \eta_{3\ell}(R_E) \end{bmatrix} \quad (2.5-82)$$

and

$$\begin{bmatrix} \ddot{M}_1 \\ \ddot{M}_2 \\ \ddot{M}_3 \end{bmatrix} = \frac{\Omega^3}{c} \begin{bmatrix} \sin \Omega t & \cos \Omega t & 0 \\ -\cos \Omega t & \sin \Omega t & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \eta_{1\ell}(R_E) \\ R_E \\ \eta_{3\ell}(R_E) \end{bmatrix} \quad (2.5-83)$$

The position of the source is provided by

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} \cos \Omega t & -\sin \Omega t & 0 \\ \sin \Omega t & \cos \Omega t & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \eta_{1\ell}(R_E) \\ R_E \\ \eta_{3\ell}(R_E) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ V_F t \end{bmatrix}, \quad (2.5-84)$$

and the unit vector in the radiation direction is given by the usual expression

$$\hat{\vec{r}}(R_E) = [\vec{x} - \vec{y}(R_E)] / |\vec{x} - \vec{y}(R_E)| \quad (2.5-85)$$

Thus we have the following for  $\dot{\vec{M}}_r$ ,  $\ddot{\vec{M}}_r$ , and  $\hat{\ddot{\vec{M}}}_r$ :

$$\dot{\vec{M}}_r = \dot{\vec{M}} \cdot \hat{\vec{r}} \quad (2.5-86)$$

$$\ddot{\vec{M}}_r = \ddot{\vec{M}} \cdot \hat{\vec{r}} + \dot{\vec{M}} \cdot (\widehat{d\hat{\vec{r}}/dt}) \quad (2.5-87)$$

and

$$\hat{\ddot{\vec{M}}}_r = \ddot{\vec{M}} \cdot \vec{r} + 2 \dot{\vec{M}} \cdot (\widehat{d\hat{\vec{r}}/dt}) + \dot{\vec{M}} \cdot (\widehat{d^2\hat{\vec{r}}/dt^2}) \quad (2.5-88)$$

The unit vector derivatives are expressed as:

$$\widehat{d\hat{\vec{r}}/dt} = \frac{\dot{\vec{r}}}{r} - \frac{\vec{r}}{r^2} \dot{r} \quad (2.5-89)$$

and

$$\widehat{d^2\hat{\vec{r}}/dt^2} = \frac{\ddot{\vec{r}}}{r} - 2 \frac{\dot{\vec{r}} \cdot \dot{\vec{r}}}{r^2} - \frac{\vec{r} \cdot \ddot{\vec{r}}}{r^2} + \frac{\dot{r}(\dot{r})^2}{r^3} \quad (2.5-90)$$

The various quantities in Equation (2.5-90) being given by

$$\dot{\vec{r}} = -\dot{\vec{y}}(R_E) \quad (2.5-91a)$$

$$\ddot{\vec{r}} = -\ddot{\vec{y}}(R_E) \quad (2.5-91b)$$

and

$$\dot{\vec{r}} = -\dot{\vec{y}} \cdot \hat{r} \quad (2.5-91c)$$

$$\ddot{\vec{r}} = -\ddot{\vec{y}} \cdot \hat{r} + \left| \frac{\dot{\vec{y}}}{r} \right|^2 - \left| \frac{\dot{\vec{y}} \cdot \hat{r}}{r} \right| \quad (2.5-91d)$$

The remaining quantity to be evaluated is the blade volume,, which is by definition:

$$\Psi = \int dV \quad (2.5-92)$$

By use of the divergence theorem, this can be expressed as a surface integral

$$\Psi = \int_{f=0} dV = \int \eta_3 \hat{n}_3 dS, \quad (2.5-93)$$

where  $\hat{n}_3$  is the component of the unit surface normal in the direction of  $n_3$ . Equation (2.5-93) can be further expressed as

$$\Psi = \int_0^1 \int_0^{2\pi} \eta_3(\xi_1, \xi_2) \hat{n}_3(\xi_1, \xi_2) J(\xi_1, \xi_2) d\xi_2 d\xi_1, \quad (2.5-94)$$

where the Jacobian, J, is defined to be

$$J = \left| \frac{\partial \vec{\eta}}{\partial \xi_1} \times \frac{\partial \vec{\eta}}{\partial \xi_2} \right|. \quad (2.5-95)$$

### 2.5.2.9 Roots of Retarded Time Equation

The retarded time equation is written

$$A_\phi^2 + B_\phi + C + \cos(\phi+D) = 0 \quad (2.5-96)$$



where only the root  $(\phi/\Omega) < 0$  is of interest. Since both Equation (2.5-96) and its first derivative have an oscillating term, a close initial guess is required to ensure the convergence of Newton's method. This is accomplished by replacing the cosine term by a parabola and solving the resulting quadratic equation. Details are given below.

### Step 1. Bracket the Root

Figure 2.5-4 illustrates that the roots of Equation (2.5-96) are the intersection of a parabola and a cosine curve. As a first approximation to the root, find  $\phi_1$  such that

$$A_{\phi_1}^2 + B_{\phi_1} + C = 0 \quad (\phi_1 < 0) \quad (2.5-97)$$

Next determine which half cycle of the cosine curve to replace by a quadratic. Let

$$= \begin{cases} \text{INT} [\phi_1 + D]/\pi & \text{if } (\phi_1 + D) \geq 0 \\ \text{INT} [\phi_1 + D]/\pi - 1 & \text{if } (\phi_1 + D) < 0 \end{cases} \quad (2.5-98)$$

where  $\ell$  is the integer half cycle number and INT stands for the truncation integer function.

### Step 2. Determine Initial Guess

The cosine term is replaced either by a straight line or by a parabola depending on whether  $\ell$  is an even integer or an odd integer.

$$\cos(\phi+D) = \begin{cases} -\phi - D + (\ell + 1/2)\pi & \text{if } \ell \text{ even} \\ 1 - 4[\phi + D + (\ell+1)\pi]^2/\pi^2 & \text{if } \ell \text{ odd} \end{cases} \quad (2.5-99)$$

With the above substitution, Equation (2.5-96) can be solved, using the quadratic formula, to give an initial guess  $\phi_2$ .

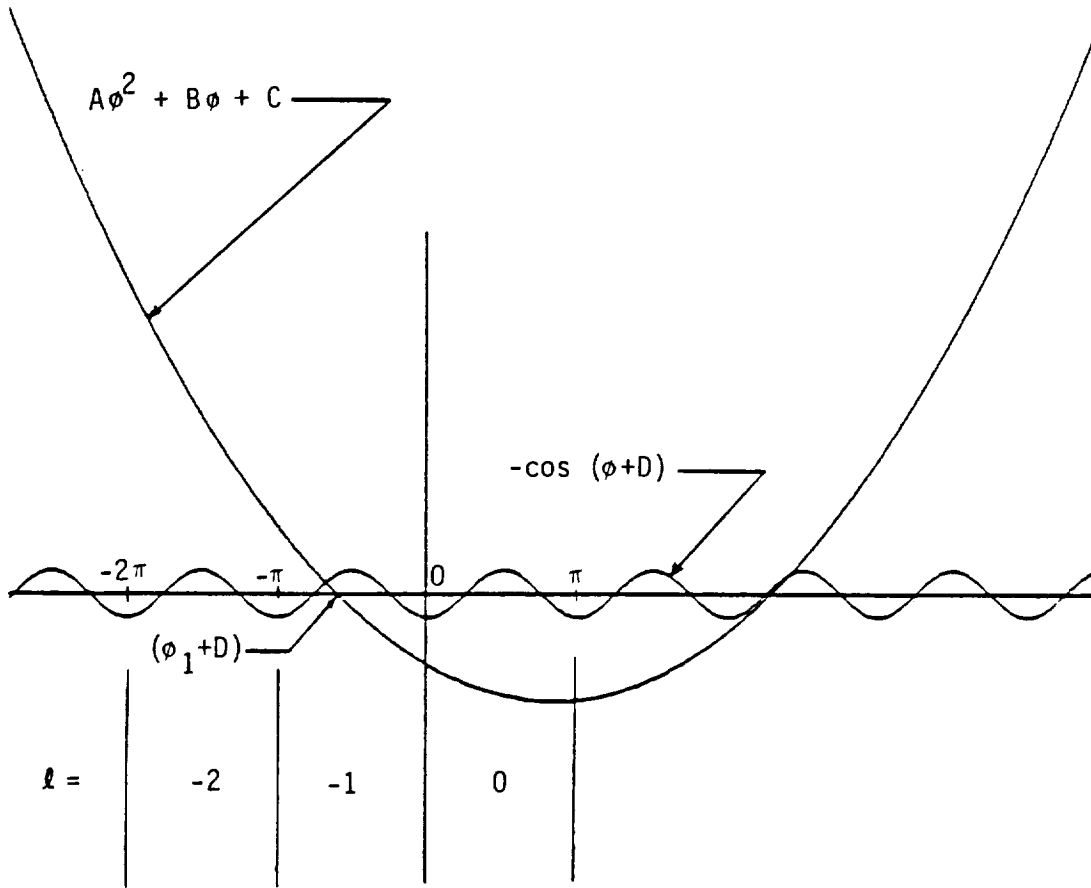


Figure 2.5-4. Roots of the Retarded Time Equation

### Step 3. Newton's Method

The roots of Equation (2.5-96) can now be determined through an iterative process,

$$\phi_{n+1} = \phi_n - \frac{[A\phi_n^2 + B\phi_n + C + \cos(\phi_n + D)]}{[2A\phi_n + B - \sin(\phi_n + D)]} \quad (2.5-100)$$

which terminates when  $|\phi_{n+1} - \phi_n|$  is smaller than a user specified convergence criterion or, if the number of iterations exceeds a limit set by the user.

### Step 4. Return Answer

If  $(\phi_{n+1}/\Omega) < 0$  or if  $\phi_{n+1}$  substituted in Equation (2.5-96) does not return a value less in magnitude than a user specified tolerance, then an error condition results; otherwise,  $\phi_{n+1} = \Omega(\tau^* - t)$ .

Note that the retarded time equation is solved for one source point at many receiver times. Thus,  $\phi_{n+1}$  may be a close initial guess to the next required root. In this case, Step 1 and Step 2 can be skipped.

#### 2.5.2.10 Acoustic Spectrum Calculation

The acoustic spectrum is attained by calculating the amplitude of the complex coefficients of the Fourier series for  $p_p(\vec{x}, t)$ . A standard FFT routine based on the Cooley-Tukey algorithm is used for this purpose.

Let  $N_S$  be the number of harmonics desired. Find  $m$  such that  $2^{m-1} < 2N_S \leq 2^m$  and form the complex array:

$$Z_j = (p_p(\vec{x}, t_j), 0) \quad , \quad (j = 1, 2, \dots, N_t - 1) \quad (2.5-101)$$

where

$$N_t = 2^m \text{ and } t_j = (2\pi j / N_t) \quad (2.5-102)$$

Now find the inverse Fourier transform

$$C = \frac{1}{N_t} \sum_{j=0}^{N_t-1} z_j e^{-2\pi i k j / N_t} \quad (k = 0, 1, \dots, N_t) \quad (2.5-103)$$

The acoustic spectrum is given by

$$SPL_k = 10 \log [2 |c_k|^2 / p_{ref}^2] \quad (k = 1, 2, \dots, N) \quad (2.5-104)$$

#### 2.5.2.11 SPN Input

The computation of subsonic propeller noise requires data of the blade shape characteristics, aerodynamic characteristics, flow and operating parameters and observer geometry. The noise signature is calculated using (1) full blade formulation, (2) mean surface approximation, (3) compact chord approximation, or (4) compact source approximation. Option numbers in parenthesis identify the inputs which correspond to each method.

#### User Parameters (1, 2, 3, 4)

B	Blade length measured from axis to tip, m (ft)
N <sub>b</sub>	Number of blades
N <sub>s</sub>	Highest harmonic number desired
V <sub>F</sub>	Aircraft forward velocity, m/s (ft/s)
θ <sub>0</sub>	Root pitch change angle, rad
ψ <sub>0</sub>	Initial azimuth angle of first blade, rad
Ω	Angular velocity of blade, rad/s

Blade Shape Table (1, 2, 3, 4)

$\xi_1$	Spanwise location, normalized by B
$\xi_2$	Chordwise location, rad
$\eta_1(\xi_1, \xi_2)$	Blade surface abscissa, normalized by B
$\eta_3(\xi_1, \xi_2)$	Blade surface abscissa, normalized by B
$\phi_1, \phi_2$	Slopes of basis functions at nodes

Leading Edge Table (3)

$\eta_1(\xi_1)$	Abscissa of leading edge, normalized by B
$\eta_3(\xi_1)$	Abscissa of leading edge, normalized by B

Aerodynamic Characteristics Table (1, 2)

$\xi_1$	Spanwise location, normalized by B
$\xi_2$	Chordwise location, rad
$\psi$	Azimuthal location, rad
$p(\xi_1, \xi_2, \psi)$	Pressure loading normalized by $\rho_0(B\Omega)^2$
$\sigma(\xi_1, \xi_2, \psi)$	Skin function loading, normalized by $\rho_0(B\Omega)^2$

Observer Table (1, 2, 3, 4)

$\vec{x}$	Initial observer position, m (ft)
-----------	-----------------------------------

### Atmospheric Properties Table (1, 2, 3, 4)

$\rho_0$  Ambient fluid density, kg/m<sup>3</sup> (lbm/ft<sup>3</sup>)

c Local sound speed, m/s (ft/s)

### Performance Characteristics Table

$F_z(\psi)$  Axial section force, normalized by  $\rho_0 B^3 \Omega^2$  (3)

$F_\psi(\psi)$  Azimuthal section, normalized by  $\rho_0 B^3 \Omega^2$  (3)

$T(\psi)$  Net thrust for a single blade, normalized by  $\rho_0 B^4 \Omega^2$  (4)

$H(\psi)$  Net azimuthal force for a single blade, normalized by  $\rho_0 B^4 \Omega^2$  (4)

$\xi_T(\psi)$  Effective radius based on net thrust, normalized by B (4)

$\xi_Q(\psi)$  Effective radius based on azimuthal force, normalized by B (4)

#### 2.5.2.12 Program Validation

The SPN program has been verified for both near-field and far-field conditions by comparison with experimental data for the 101/16 and SR-3 propellers. Near-field data for the 101/16 propeller were taken in flight using a wing-mounted microphone on an Aero-Commander (Ref. 2.5-8) as shown in Figure 2.5-5. Pressure signatures and overall noise spectra for three different microphone boom positions (X3), as calculated using SPN, are compared with flight data in Figures 2.5-6 through 2.5-11. Also shown for comparison purposes are calculations carried out at Ohio State University using the Woan-Gregorek code (Ref. 2.5-5). Flight Mach number ( $M_\infty$ ), pressure altitude (PALT), number of propeller blades (B), and propeller RPM ( $n$ ) were as shown in the figures. Overall good agreement of the acoustic pressure signatures (Figures 2.5-6, 2.5-8, and 2.5-10) with flight measurement may be noted in all three cases. In addition, agreement of the noise spectra results is excellent for the first five harmonics but decreases

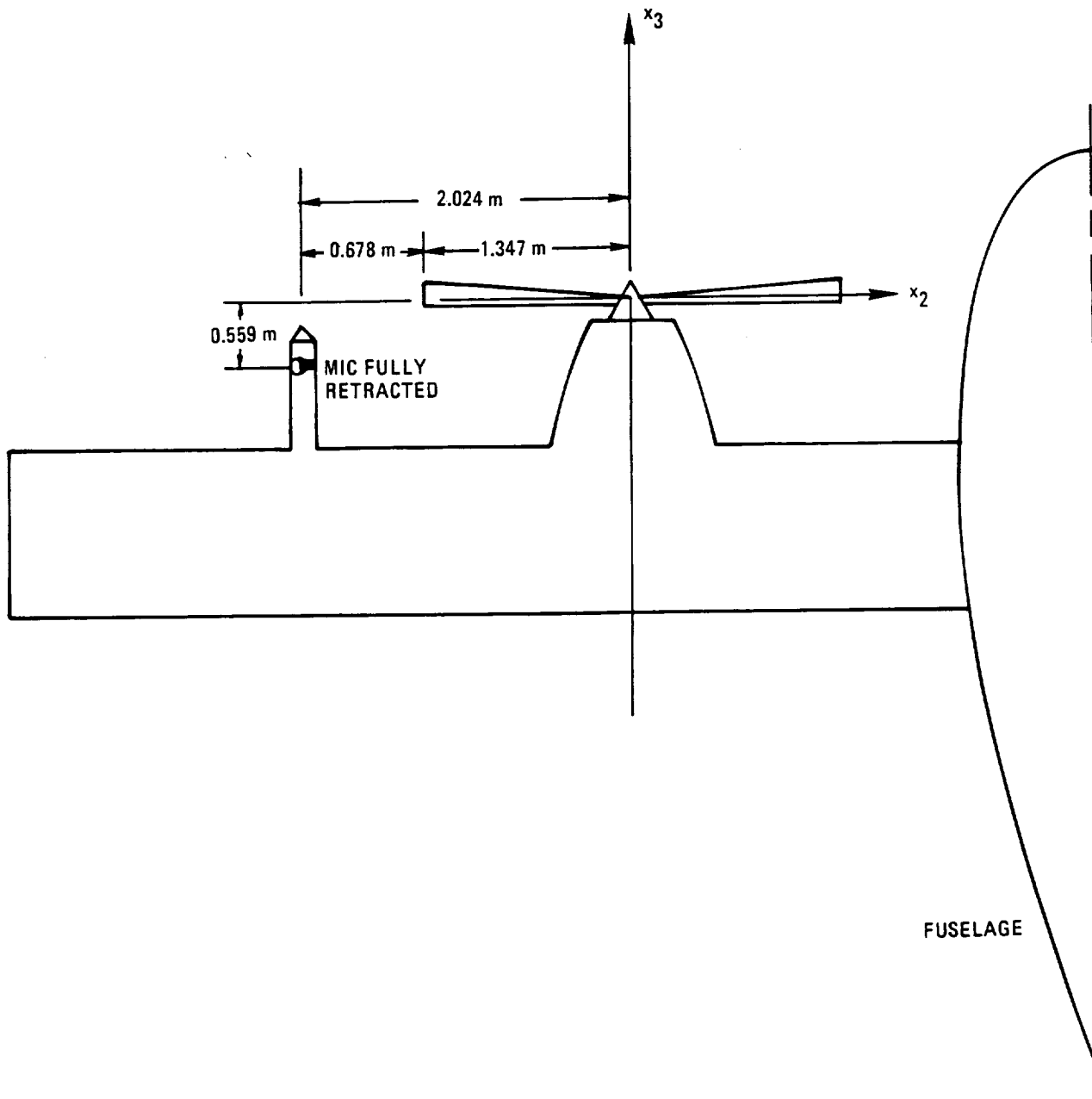


Figure 2.5-5. Wing-Mounted Microphone Position 101/16 Propeller Acoustic Measurements Near Field

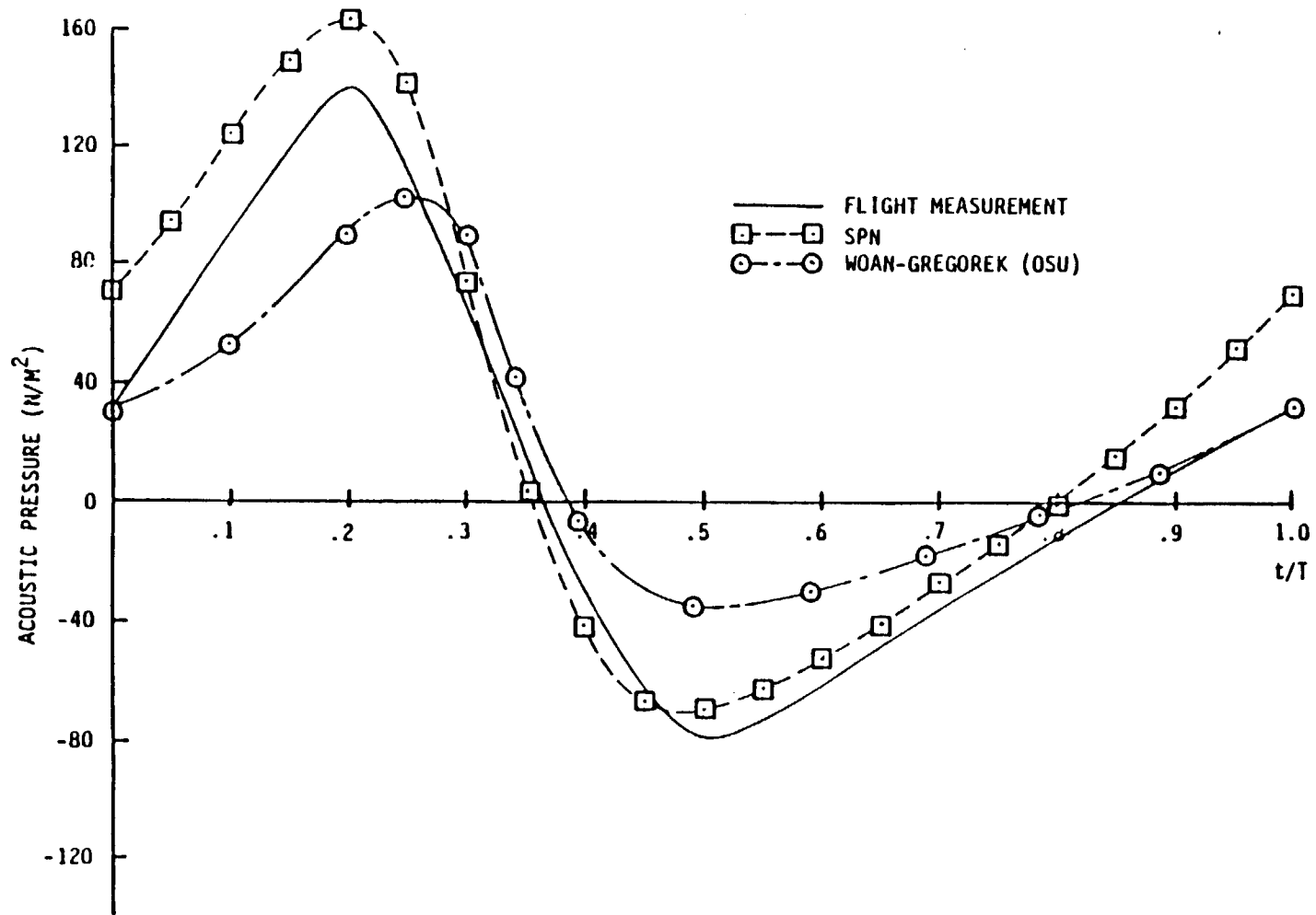


Figure 2.5-6. Acoustic Pressure Signature 101/16 Propeller Near Field (Run No. 11)  
 $M_\infty = 0.317$ , PALT = 3043 M, B = 3, n = 1591 rpm, boom position (M)

X1	X2	X3
0.0	-2.024	-0.279



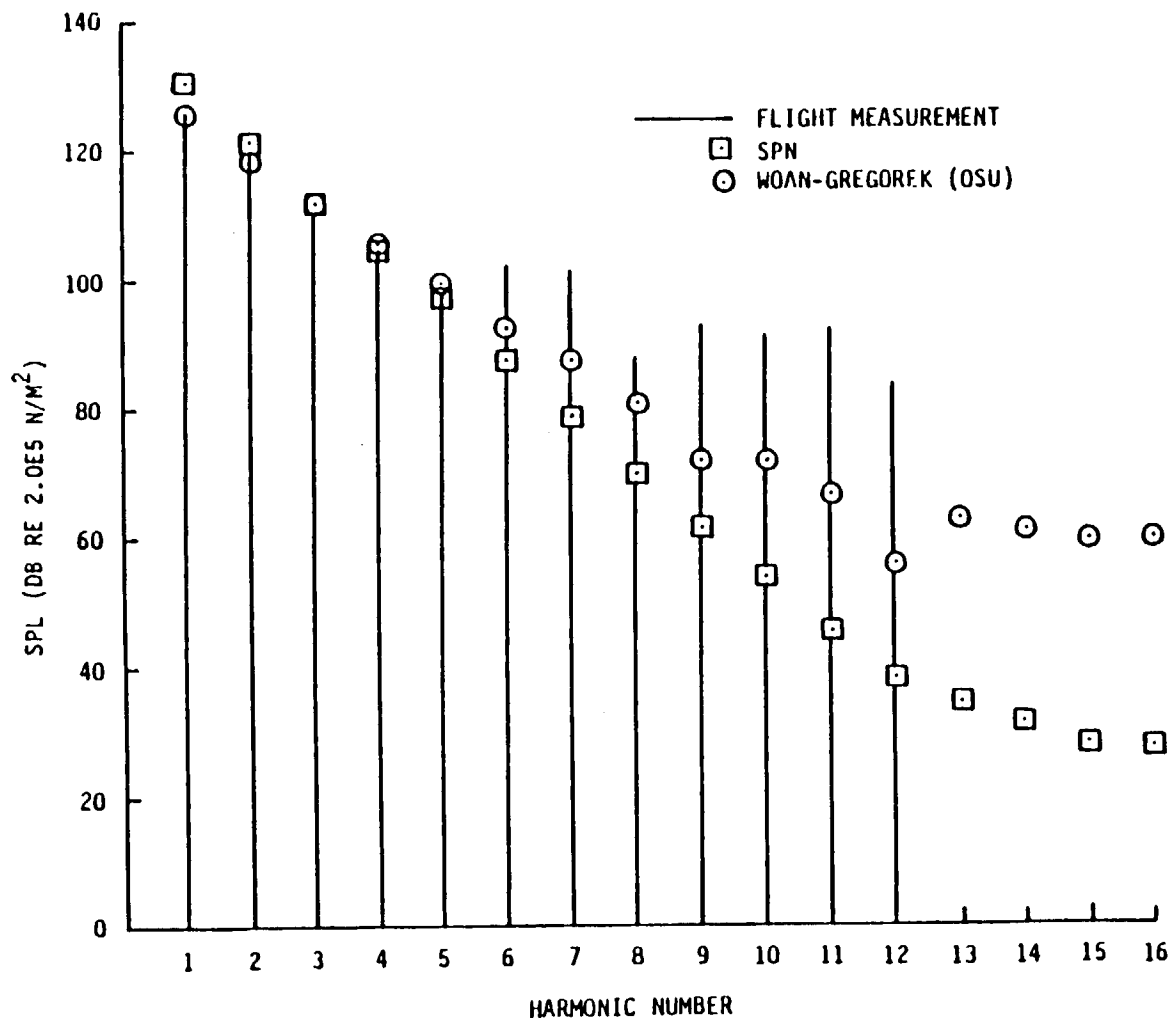


Figure 2.5-7. Overall Noise Spectra 101/16 Propeller Near Field (Run No. 11)  
 $M_{\infty} = 0.317$ , PALT = 3043 M, B = 3, n = 1591 rpm, boom position (M)

X1	X2	X3
0.0	-2.024	-0.279

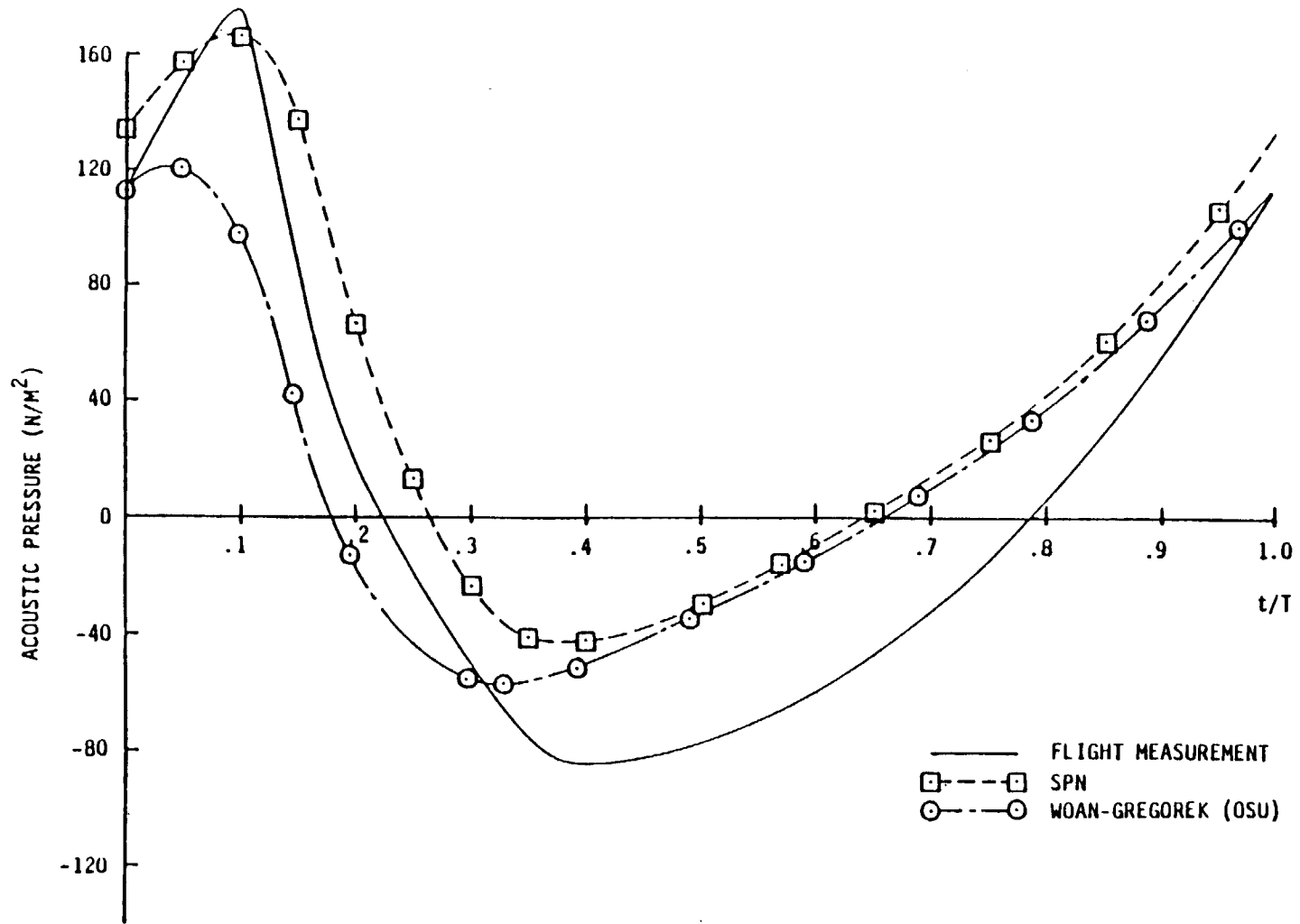


Figure 2.5-8. Acoustic Pressure Signature 101/16 Propeller Near Field (Run No. 12)  
 $M_{\infty} = 0.317$ , PALT = 3043 M, B = 3, n = 1591 rpm, boom position (M)

X1	X2	X3
0.0	-2.024	-0.406

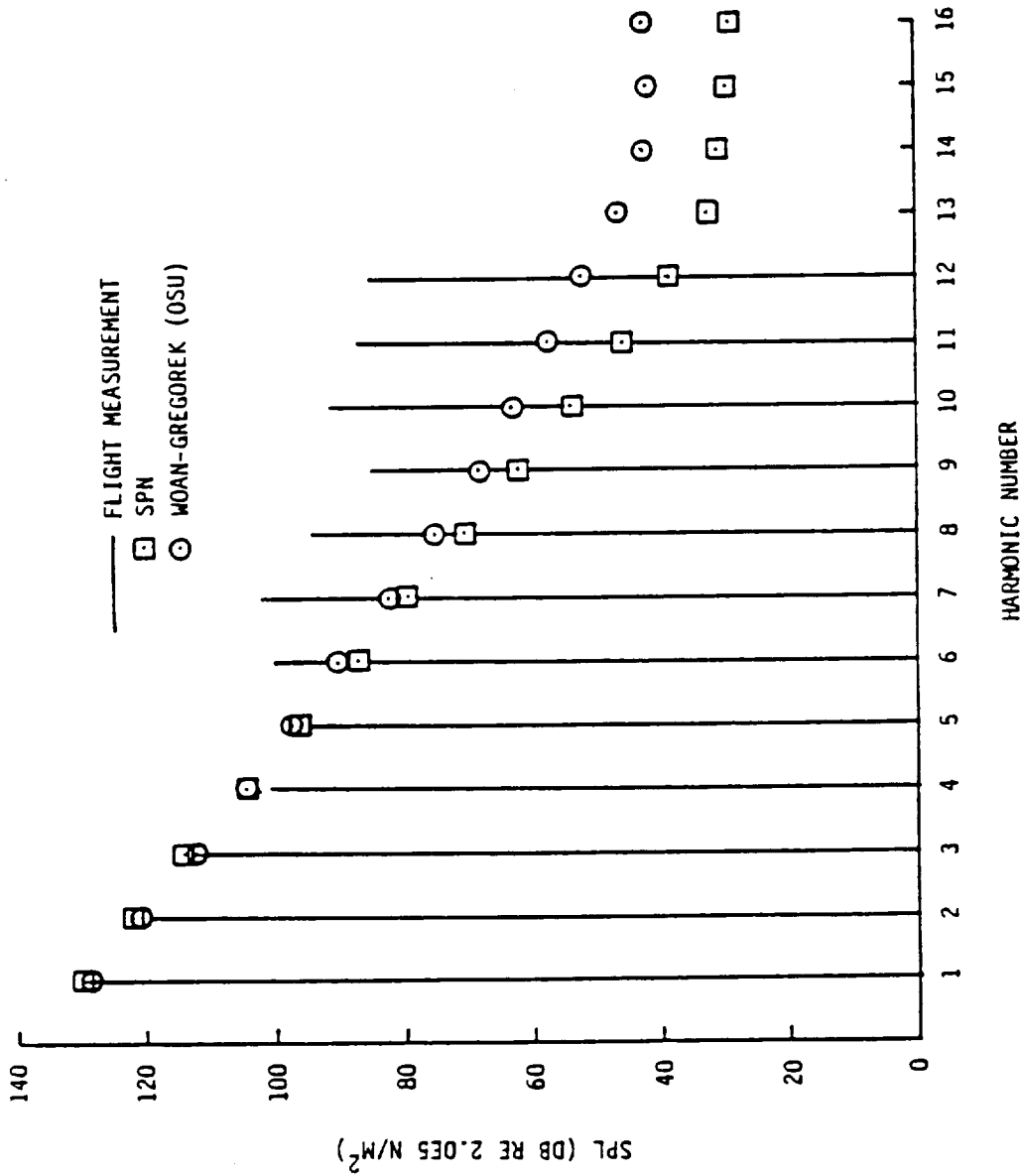


Figure 2.5-9. Overall Noise Spectra 101/16 Propeller Near Field (Run No. 12)  
 $M_{\infty} = 0.317$ ,  $PALT = 3043$  M,  $B = 3$ ,  $n = 1591$  rpm, boom position (M)  
 $X1 \quad X2 \quad X3$   
 $0.0 \quad -2.024 \quad -0.406$

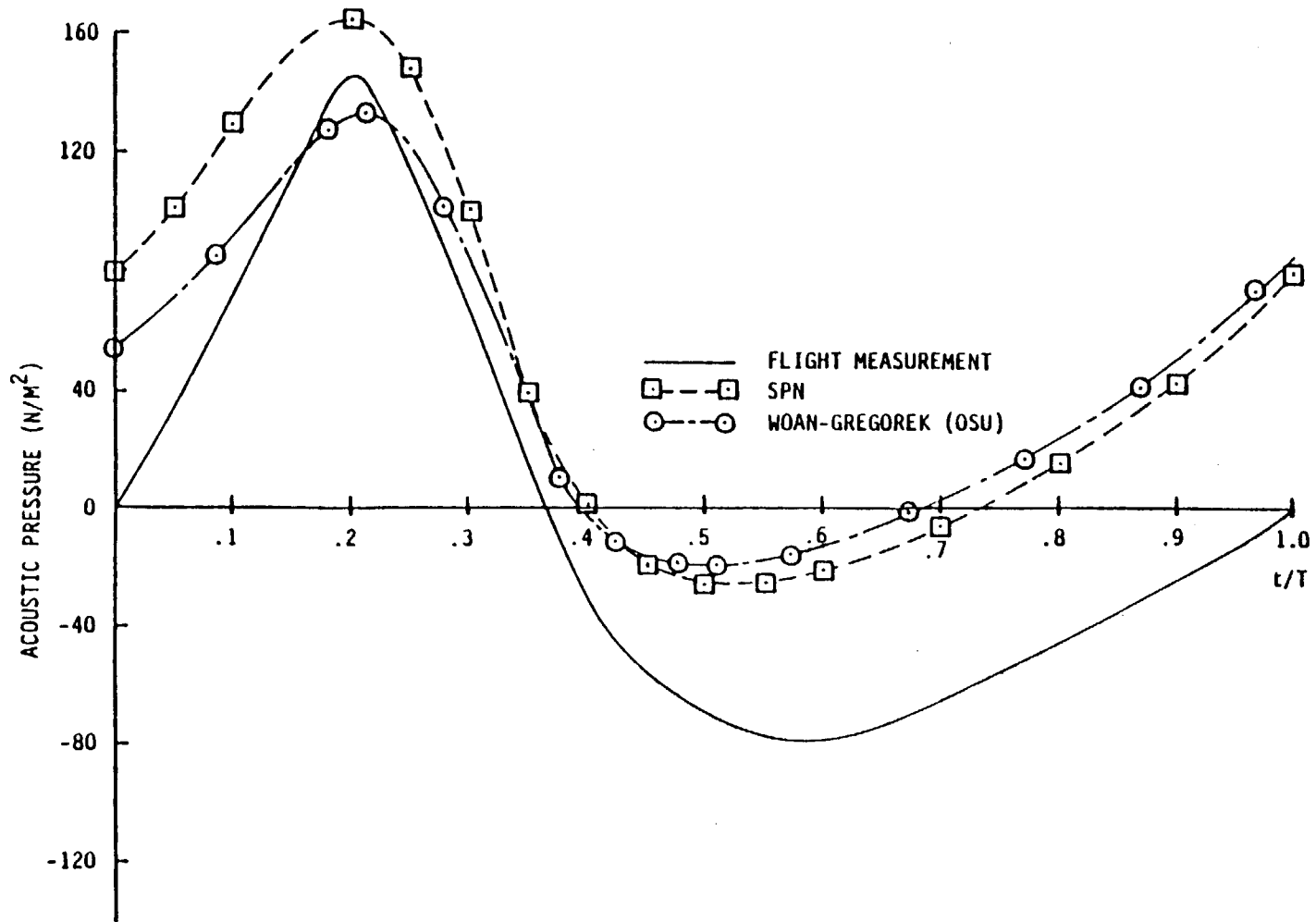


Figure 2.5-10. Acoustic Pressure Signature 101/16 Propeller Near Field (Run No. 13)  
 $M_{\infty} = 0.317$ , PALT = 3043 M, B = 3, n = 1591 rpm, boom position (M)

X1	X2	X3
0.0	-2.024	-0.559

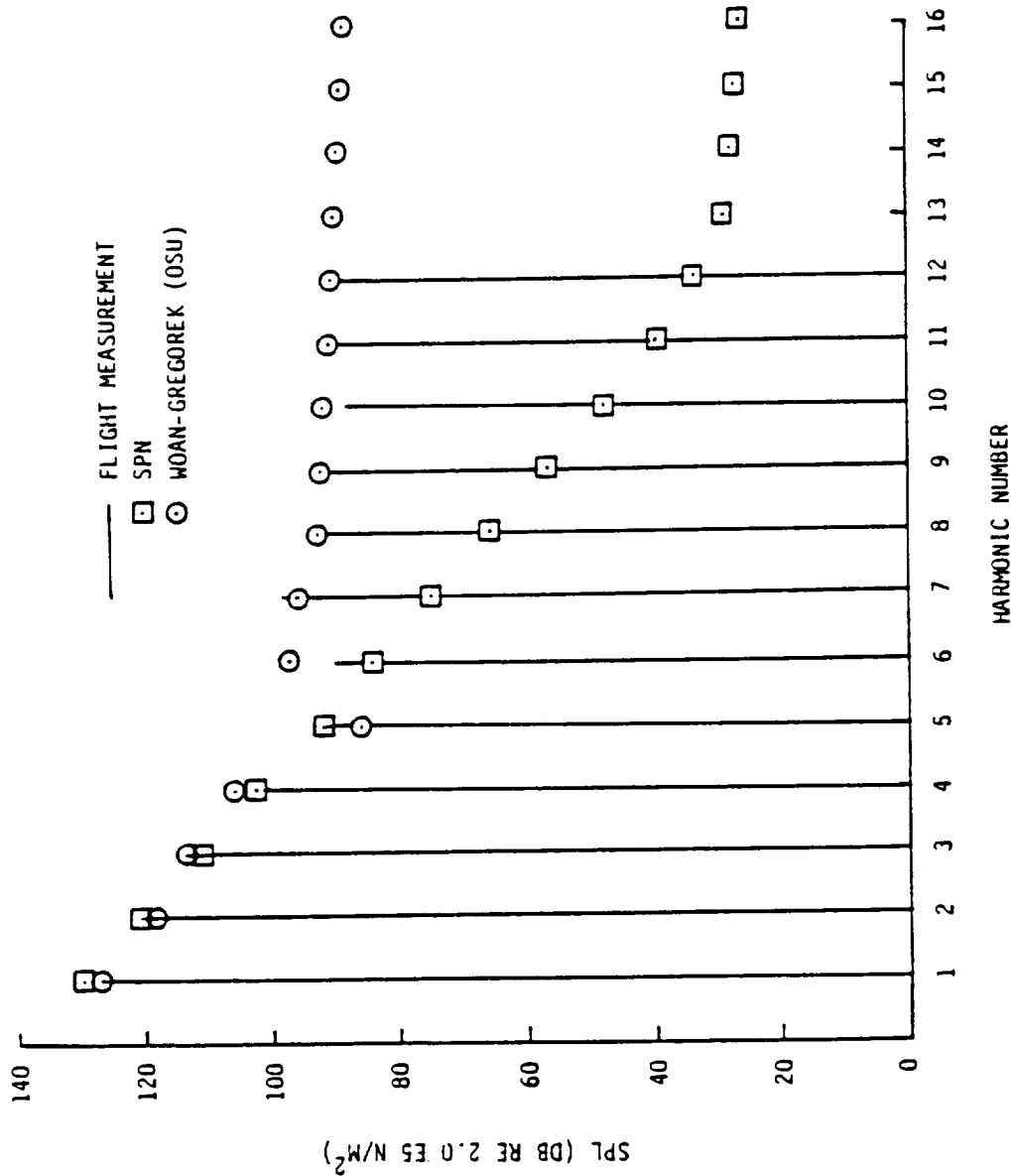


Figure 2.5-11. Overall Noise Spectra 101/16 Propeller Near Field (Run No. 13)  
 $M_{\infty} = 0.317$ ,  $PAL1 = 3043 M$ ,  $B = 3$ ,  $n = 1591$  rpm, boom position (M)

$$\frac{X1}{0.0} \frac{X2}{-2.024} \frac{X3}{-0.559}$$

substantially for harmonic numbers greater than five. This is not of practical significance, however, since most of the acoustic energy is in the first few harmonics. Moreover, the OASPL (overall sound pressure level) results for the three 101/16 propeller cases are shown in Table 2.5-1. Substantially better agreement with flight data of the SPN calculation over that of the Woan-Gregorek code is clearly evident.

Table 2.5-1. OASPL Results - 101/16 Propeller Near-Field Conditions

$$(OASPL = 10\text{Log}_{10} \sum_{n=1}^N (P_{rms}/P_{ref})^2; N = \text{Total number of harmonics})$$

<u>Run No. 11</u>	Flight Data	131.15 dB
	SPN	131.54 dB
	Woan-Gregorek	126.85 dB
<u>Run No. 12</u>	Flight Data	130.86 dB
	SPN	130.67 dB
	Woan-Gregorek	129.18 dB
<u>Run No. 13</u>	Flight Data	130.95 dB
	SPN	129.96 dB
	Woan-Gregorek	127.66 dB

Turning now to results for the SR-3 propeller, comparison of SPN calculations and experimental data at 1.5 tip diameters for an eight-bladed propeller with an advance Mach number 0.6 and helical tip Mach number of 0.86 are shown in Figure 2.5-12. Computational results shown in this figure are those calculated at TRW using NASA-Langley developed aerodynamic modules to calculate propeller-blade loading (ANOPP Level 03/01/03).

Additional near-field results for a two-bladed SR-3 propeller at 0.8 diameter tip clearance having an advance Mach number of 0.203 and a helical tip Mach number of 0.687 are shown in Figures 2.5-13 and 2.5-14. Note the

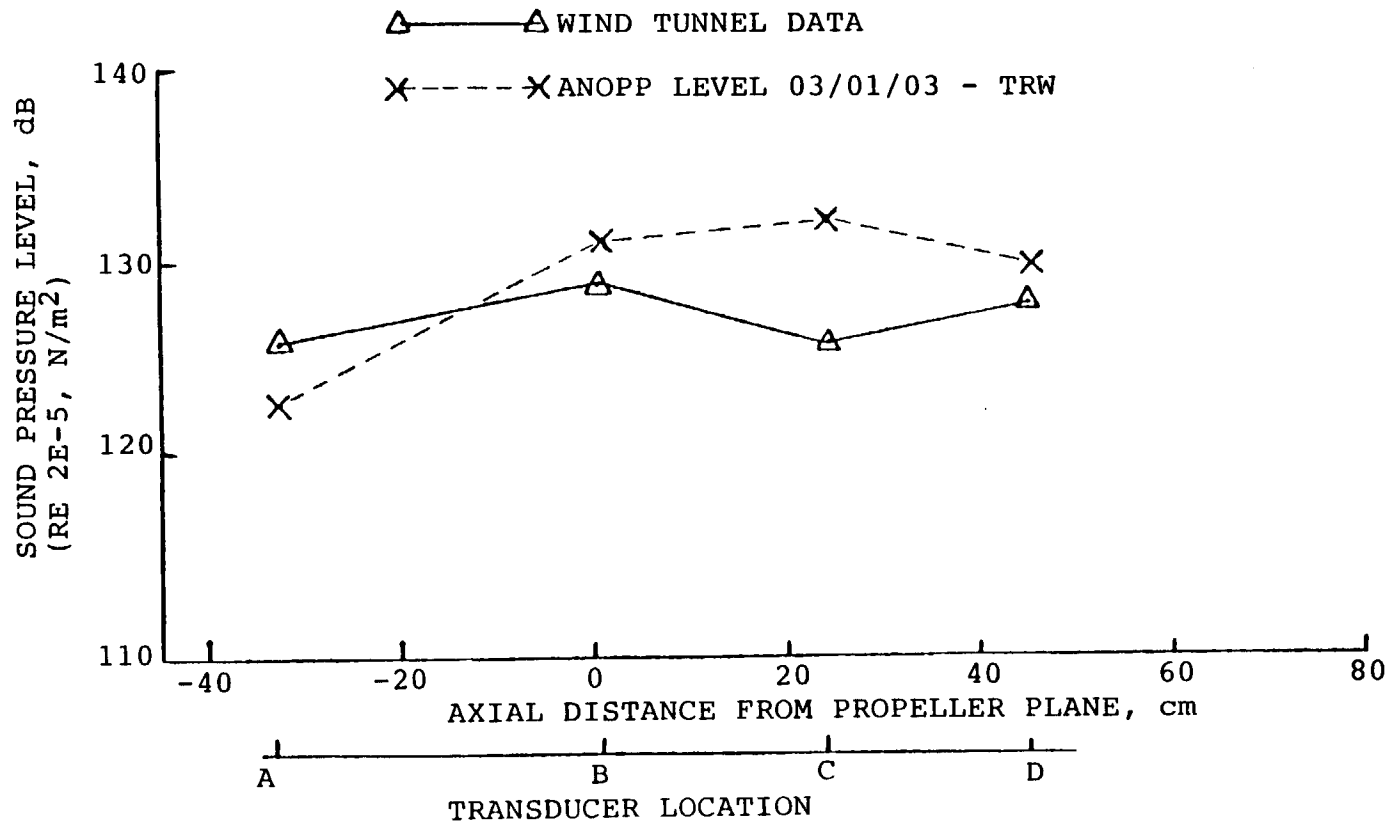


Figure 2.5-12. Acoustic Verification Results SR-3 Propeller  
B = 8,  $\beta_{.75R} = 61.3^\circ$ ,  $M_0 = 0.60$ ,  $J = 3.06$ ,  $M_{TH} = 0.86$

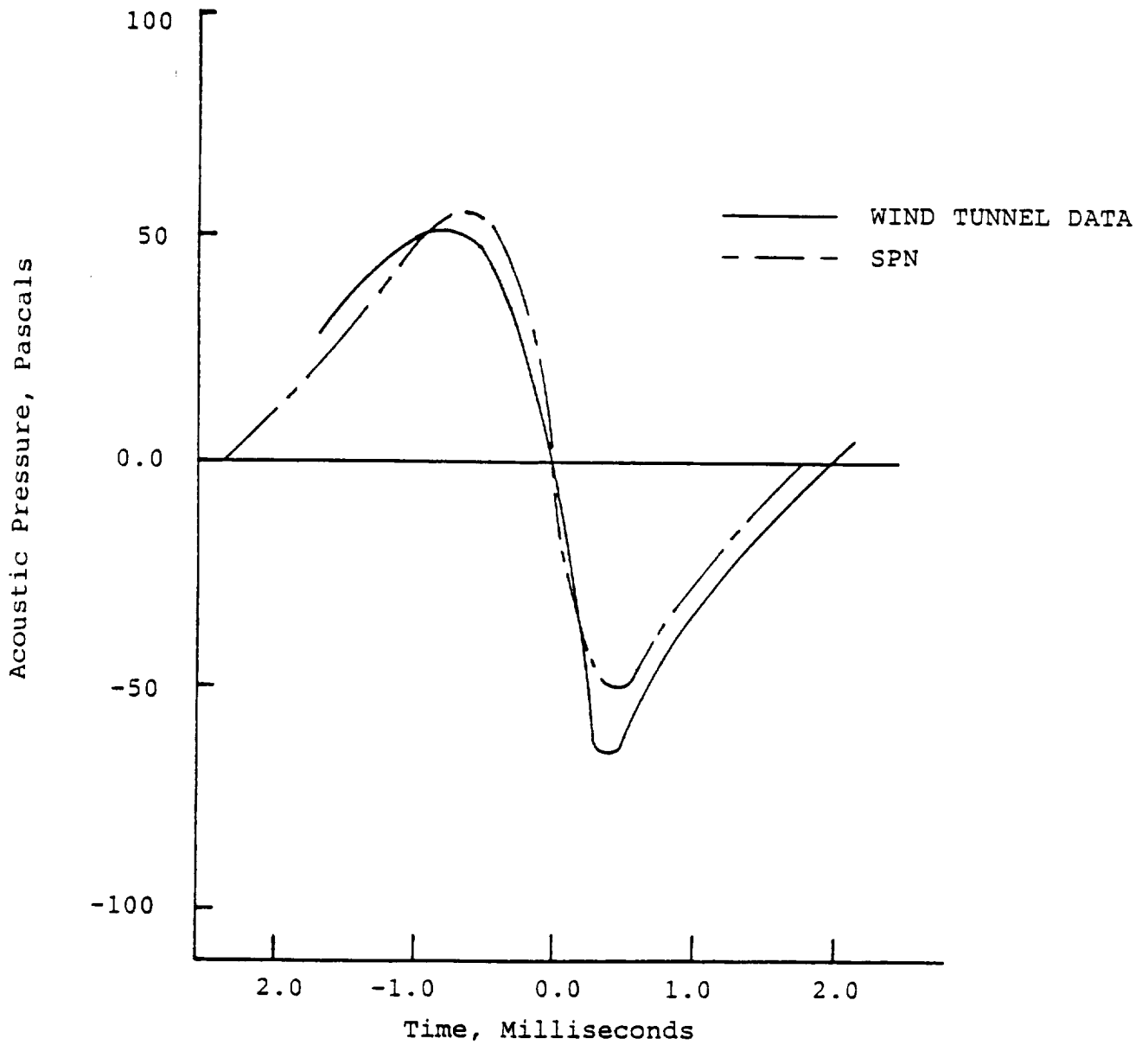


Figure 2.5-13. Propeller Plane Near-Field Pressure Signature at 0.8 D Tip Clearance Radial Location; SR-3 Propeller,  $N_b = 2$ ,  $M_0 = 0.203$ , 6700 rpm,  $M_{TH} = 0.687$



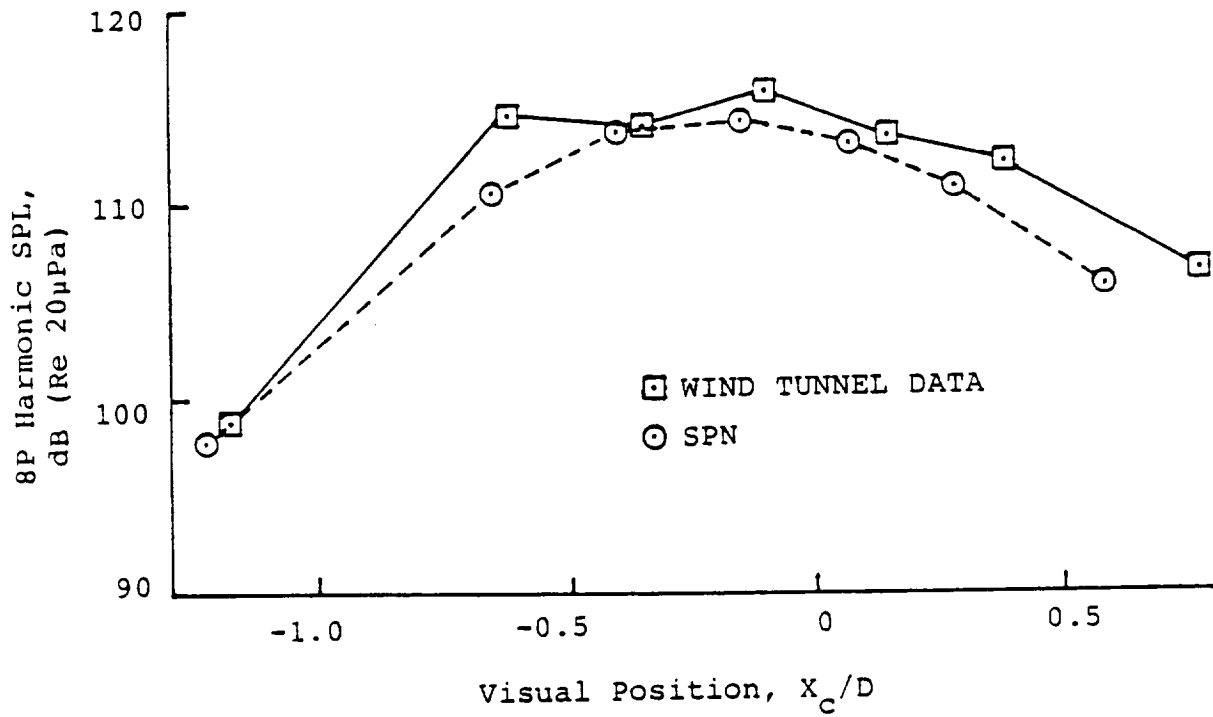


Figure 2.5-14. 8P Harmonic Sound Pressure Level at 0.8 D Tip Clearance; SR-3 Propeller,  $N_b = 2$ ,  $M_0 = 0.203$ , 6700 rpm,  $M_{TH} = 0.687$

good agreement of these calculations with experiment for both the acoustic pressure signature and the 8P harmonic sound pressure level.

Finally, a far-field results for the SR-3 propeller are compared with experimental data taken in the UTRC ART tunnel for helical tip Mach numbers of 0.901 and 0.671 in Figures 2.5-15 and 2.5-16, respectively. Results are shown for the 8P harmonic sound pressure level calculated using the NASA-Langley ANOPP code as run at TRW, which includes Langley aerodynamic loading and performance modules. Note the better agreement with data for the helical tip Mach number 0.761 case than for the Mach 0.901 case, which is as might be expected due to inaccuracies in calculating the transonic blade aerodynamics in the Mach 0.901 case.

### 2.5.3 Elliptic Coordinate System

For the acoustics module it is important to be able to interpolate the pressure obtained by the airfoil loading module in an elliptic coordinate system, as these coordinates are used to model the airfoil since the leading edge can be closely approximated by an ellipse. The section data are transformed into an elliptic coordinate system through the use of the inverse Joukowski transformation.

The Joukowski transformation, defined by

$$z = \zeta + \frac{b^2}{\zeta} \quad (2.5-105)$$

maps the airfoil section from the complex  $z$ -plane to the complex  $\zeta$ -plane where  $\zeta$  is given by

$$\zeta = b e^{\psi+i\theta} \quad (2.5-106)$$

In the  $\zeta$ -plane the airfoil shape is nearly circular. Recall from complex variables that  $z = re^{i\theta}$  is the equation of a circle. If  $\psi$  is constant in Equation (2.5-106) then the airfoil shape in the  $\zeta$ -plane is exactly a circle. In general however,  $\psi$  is not constant but is a function of  $\theta$ . The values of  $\psi$  are very small so that  $\zeta$  does not deviate very far from a circle, however.

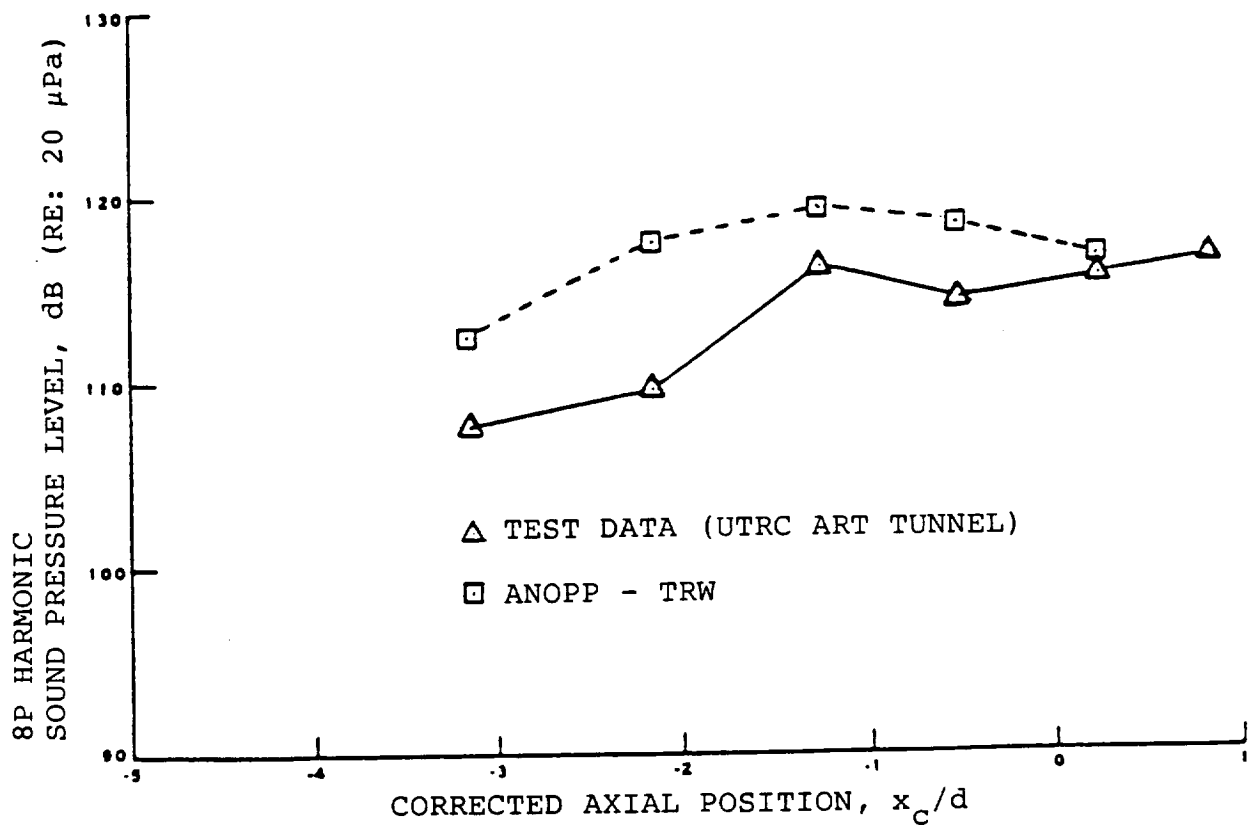


Figure 2.5-15. 8P Harmonic Sideline Directivity at 4.4 D Tip Clearance  
 SR-3,  $N_b = 4$ ,  $M_o = 0.321$ , 8550 rpm,  $M_{TH} = 0.901$

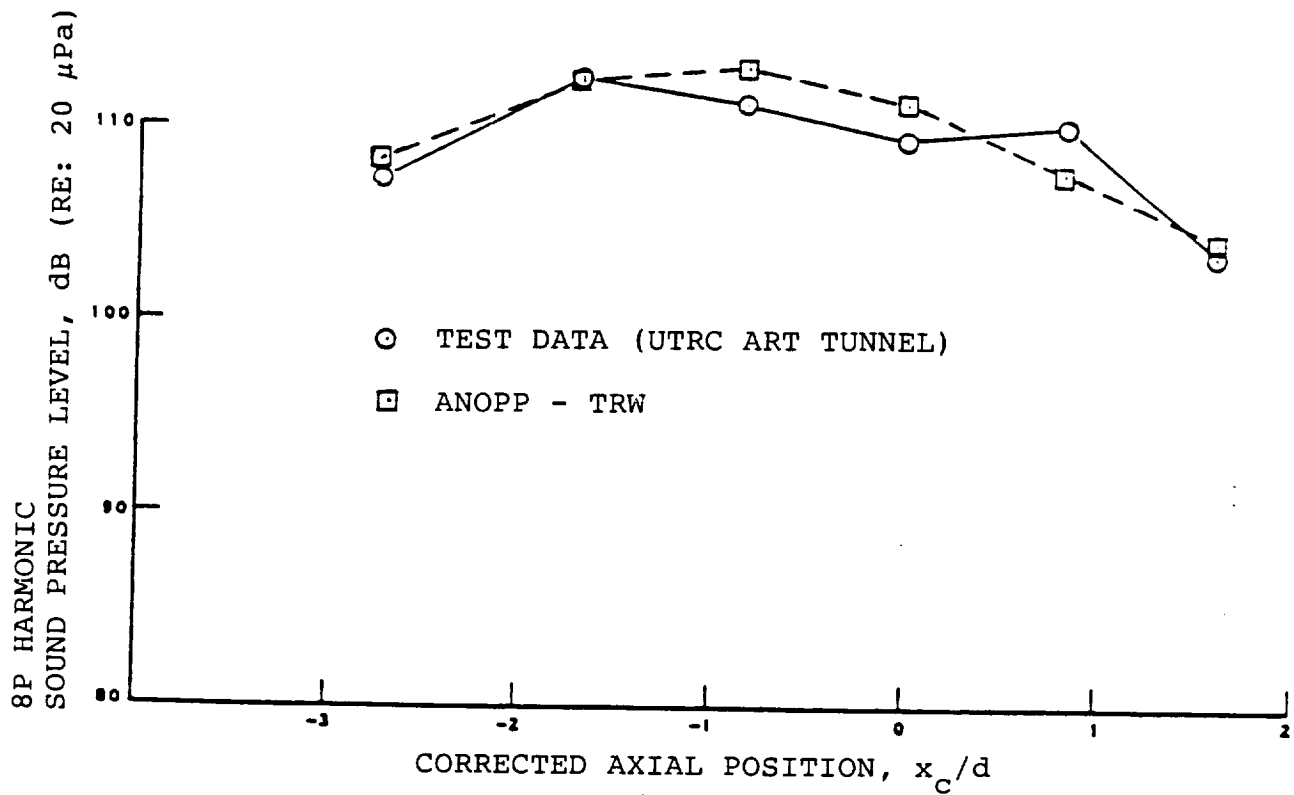


Figure 2.5-16. 8P Harmonic Sideline Directivity at 4.4 D Tip Clearance  
SR-3,  $N_b = 2$ ,  $M_o = 0.203$ , 7500 rpm,  $M_{TH} = 0.761$

To find the representation of an airfoil section in terms of  $\psi$  and  $\theta$ ,  $z$  can be rewritten as

$$z = be^{\psi+i\theta} + b e^{-\psi-i\theta} \quad (2.5-107)$$

or in terms of hyperbolic functions as

$$z = 2b \cosh\psi \cos\theta + i 2b \sinh\psi \sin\theta \quad (2.5-108)$$

Since  $z = x + i y$  in cartesian coordinates, the coordinates of the airfoil section  $x, y$  are given by

$$x = 2b \cosh\psi \cos\theta \quad (2.5-109a)$$

and

$$y = 2b \sinh\psi \sin\theta \quad (2.5-109b)$$

Solving for  $\cos\theta$  and  $\sin\theta$  in terms of  $x, y$ , and  $\psi$ , the following two relations are found

$$\cos\theta = \frac{x}{2b \cosh\psi} \quad (2.5-110a)$$

and

$$\sin\theta = \frac{y}{2b \sinh\psi} \quad (2.5-110b)$$

Using the identity  $\cos^2\theta + \sin^2\theta=1$  together with Equations (2.5-110a) and (2.5-110b) yields

$$\left(\frac{x}{2b \cosh\psi}\right)^2 + \left(\frac{y}{2b \sinh\psi}\right)^2 = 1 \quad (2.5-111)$$

For constant  $\psi$ , Equation (2.5-111) is the equation on an ellipse with foci located at  $\pm 2b$ . Consequently, lines of constant  $\psi$  form ellipses in the

z-plane. Similarly, a relation for  $\cosh\psi$  and  $\sinh\psi$  can be found by solving Equations (2.5-109a) and (2.5-109b) in terms of  $x$ ,  $y$ , and  $\theta$  to yield

$$\cosh\psi = \frac{x}{2b \cos\theta} \quad (2.5-112a)$$

and

$$\sinh\psi = \frac{y}{2b \sin\theta} \quad (2.5-112b)$$

Using the hyperbolic identity  $\cosh^2\psi - \sinh^2\psi=1$  results in

$$\left[ \frac{x}{2b \cos\theta} \right]^2 - \left[ \frac{y}{2b \sin\theta} \right]^2 = 1 \quad (2.5-113)$$

Equation (2.5-113) is the equation of a hyperbola with foci also located at  $\pm 2b$ . Lines of constant  $\theta$  form an infinite set of hyperbolas in the z-plane.

Figure 2.5-17 shows an airfoil section plotted in the complex z-plane with the elliptic and hyperbolic coordinates superimposed on the airfoil. A description of the airfoil surface is sought in terms of coordinates  $\psi$  and  $\theta$  rather than  $x$  and  $y$ .

In mapping the airfoil to the complex z-plane, the leading and trailing edges must be located in the elliptic coordinate system. Note that the derivative of the Joukowski transformation

$$\frac{dz}{d\xi} = 1 - \frac{b^2}{\xi^2} \quad (2.5-114)$$

becomes infinite for  $\xi=0$  and zero for  $\xi= \pm b$ . By choosing  $\xi_T=b$ , the trailing edge becomes a stagnation point. This choice satisfies the Kutta-Joukowski condition, which requires the flow field in the neighborhood of the trailing edge to be finite even though there is a discontinuity in the

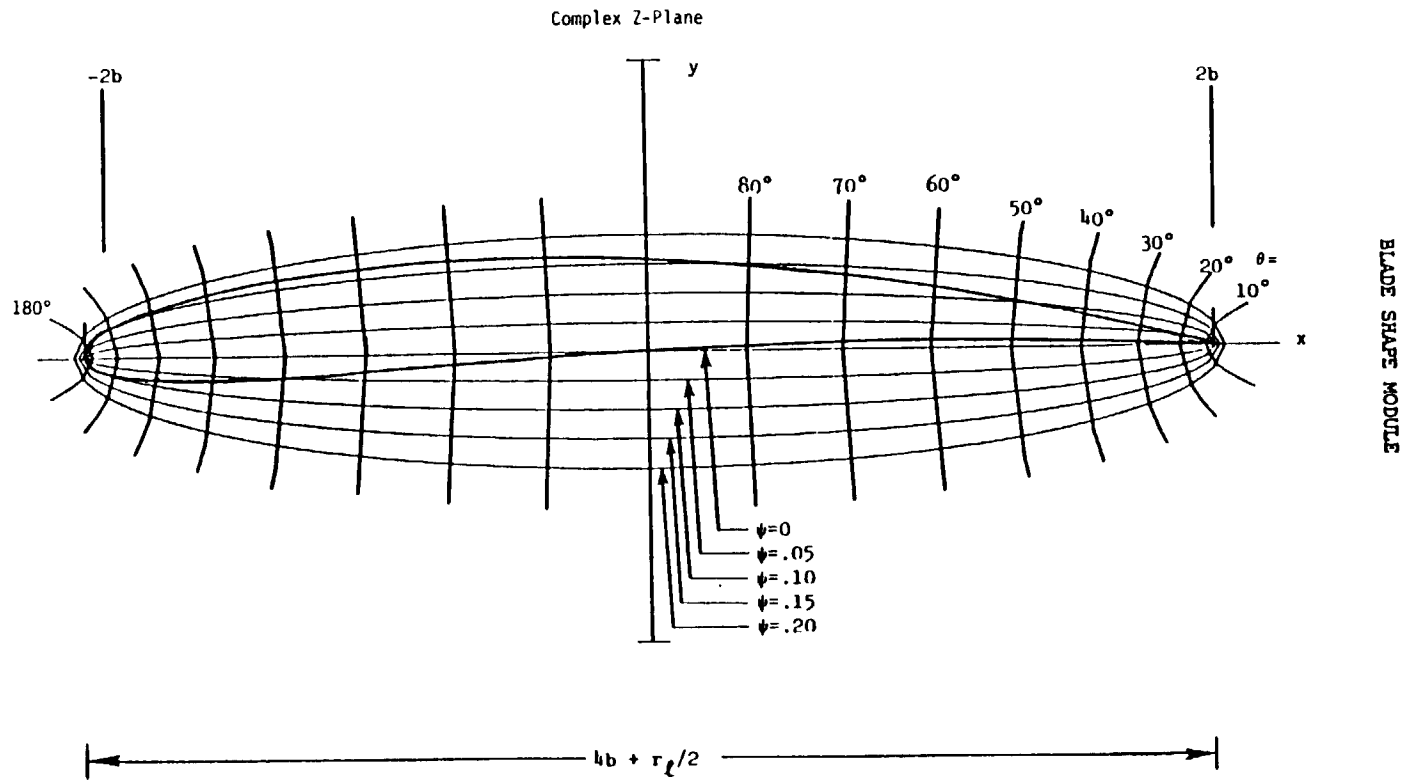


Figure 2.5-17. Typical Airfoil Section in Complex z-Plane

slope of the surface at the trailing edge. In the z-plane the trailing edge is mapped to the point

$$x_T = 2b. \quad (2.5-115)$$

Next, the leading edge must be located such that the remaining discontinuity is within the airfoil and thus will not cause a singularity in the flow field.

Recall from Equation (2.5-111) that lines of constant  $\psi$  form ellipses in the z-plane. The radius of curvature along  $\psi = \text{constant}$  is given by

$$r = \frac{2b (\cosh^2 \psi \sin^2 \theta + \sinh^2 \psi \cos^2 \theta)^{3/2}}{\sinh \psi \cosh \psi} \quad (2.5-116)$$

With the leading edge located at  $\theta = \pi$ , the radius of curvature becomes

$$r = 2b \frac{\sinh^2 \psi}{\cosh \psi} \quad (2.5-117)$$

Retaining the first terms of the Taylor series expansions for  $\sinh \psi$  and  $\cosh \psi$ , the radius of curvature is approximately given by

$$r \simeq 2b\psi^2 \quad (2.5-118)$$

The location of the leading edge in the z-plane can be found by substituting  $y=0$  into Equation (2.5-111) and solving for  $x$  to yield

$$x_L = -2b \cosh \psi \quad (2.5-119)$$

Again using the Taylor series expansion for  $\cosh \psi$ , Equation (2.5-119) becomes

$$x_L = -2b \left( 1 + \frac{\psi^2}{2} \right) \quad (2.5-120)$$



or in terms of the radius of curvature the location of the leading edge is given by

$$x_L = -2b - \frac{r}{2} \quad (2.5-121)$$

Now the chord length of the airfoil section can be found from

$$C = x_T - x_L = 4b + \frac{r}{2} \quad (2.5-122)$$

solving for b

$$b = (c - \frac{r}{2})/4$$

In the interpolation of the surface pressure and skin friction, the user specifies values of  $\theta$  from 0 to  $2\pi$ , corresponding to starting at the trailing edge and traversing the upper surface to the leading edge and back along the lower surface to the trailing edge (Figure 2.5-17).

Computationally, the first task in determining the values of x and y that correspond to a given  $\theta$  is to find the leading edge point and align the coordinates. The local coordinates are measured from the nose with the x-direction pointing through the trailing edge. Since now the local x and y coordinates are functions of the arc-length interpolation variable, s, then we can find the value of  $\sin^2\theta$  as a function of s by rearranging Equation (2.5-113).

$$\sin^4\theta + \sin^2\theta \left[ \left(\frac{x}{2b}\right)^2 + \left(\frac{y}{2b}\right)^2 - 1 \right] - \left(\frac{y}{2b}\right)^2 = 0 \quad (2.5-123)$$

Solving the quadratic for  $\sin^2\theta$

$$\sin^2\theta = - \left[ -T1 + \sqrt{T1^2 + y^2/b^2} \right] \quad (2.5-124)$$

where

$$T1 = \left(\frac{x}{2b}\right)^2 + \left(\frac{y}{2b}\right)^2 - 1.$$

We can set up a function FELLIP(S)

$$\text{FELLIP}(S) = \sin^2\theta(s) - \sin^2\theta_i$$

where  $\theta_i$  is the input value supplied by the user. The ZEROIN subroutine is used to find the root of this equation between specified bounds. Care must be exercised since each value of  $\sin^2\theta_i$  could have four possible roots corresponding to the four quadrants of the airfoil (upper/lower), leading/trailing). Once the value of  $s$  has been determined, the remaining values corresponding to that point can be interpolated (pressure, skin friction etc.).

#### 2.5.4 Transonic Propeller Noise Module

This module computes the periodic acoustic pressure signature and spectrum of a propeller with transonic tip speed. The computation is based on a solution of Ffowcs Williams-Hawkings equation (FW-H eq.) without the quadrupole source term (Refs. 2.5-1 through 2.5-3). The blade surface pressure is assumed to be steady and must be specified on a grid of points covering the blade surface. The observer is always assumed to be moving with the aircraft; only in such a frame is the acoustic pressure periodic. The solution to the FW-H equation is given by two different formulations which assume a source distribution on the full blade surface. The formulation which is suitable for blade sections moving at transonic speeds is used only when necessary. A list of symbols is provided below in order that the reader may follow the analysis.

#### Symbols

- c      Ambient speed of sound, m/s (ft/s)
- f      Function defining the blade surface

$f_k$	Frequency, Hz
$g$	Equation $g = 0$ describes relationship between source and observer time
$k$	Harmonic number
$\uparrow$	Local force per unit area of the blade acting on the fluid, Pa (lbf/ft <sup>2</sup> )
$m$	Exponent; $2^m$ is the number of time points used by FFT routine
$M$	Source Mach number
$\hat{n}$	Blade surface unit normal vector
$N_b$	Number of blades
$N_s$	Maximum harmonic number of interest
$N_t$	Number of time points in time history
$p$	Surface pressure ( $p = p_a - p_0$ ), Pa (lbf/ft <sup>2</sup> )
$p'$	Acoustic pressure, Pa (lbf/ft <sup>2</sup> )
$p_a$	Absolute surface pressure, Pa (lbf/ft <sup>2</sup> )
$p_0$	Ambient pressure, Pa (lbf/ft <sup>2</sup> )
$p'_L$	Acoustic pressure produced by loading, Pa (lbf/ft <sup>2</sup> )
$p'_p$	Total acoustic pressure produced by propeller, Pa (lbf/ft <sup>2</sup> )
$p'_T$	Acoustic pressure produced by thickness, Pa (lbf/ft <sup>2</sup> )

$p_{\text{ref}}$	Reference pressure for definition of dB, $2 \times 10^{-5}$ Pa
$\vec{r}$	Vector from source point at emission time to observer, m (ft) ( $\vec{r} =  \vec{x} - \vec{y}(\tau^*) $ )
$\hat{\vec{r}}$	Unit vector in the direction of $\vec{r}$
$\text{SPL}_k$	Acoustic spectrum, dB
$t$	Time at which noise signal is received by observer, s
$v$	Source velocity, m/s (ft/s)
$V_F$	Forward velocity of the aircraft, m/s (ft/s)
$\vec{x}$	Observer position in the ground-fixed frame, m (ft)
$\vec{X}$	Observer position in the aircraft-fixed frame, m (ft)
$\vec{y}$	Source position in the ground-fixed frame, m (ft)
$\vec{Y}$	Source position in the aircraft-fixed frame, m (ft)
$\vec{\eta}$	Source position in the blade-fixed frame, m (ft)
$\theta_0$	Change in root pitch angle, rad
$\theta$	Angle between the radiation vector and the surface normal vector, rad
$\xi_1, \xi_2$	Elliptical surface coordinates
$\rho_0$	Ambient density, $\text{kg/m}^3$ (slug/ft <sup>3</sup> )
$\tau$	Time at which noise signal is emitted at source position, s

- $\tau^*$  Solution to the retarded time equation,  $g = 0$ , or a given observer time, s
- $\phi$  Solution to retarded time equation as solved by Newton's method ( $\phi = \Omega(\tau - t)$ ), s
- $\psi$  Angle between the  $x_1$  and  $\eta_1$  axes, rad

### Superscripts

- $\rightarrow$  Vector
- $\text{---}$  Mean
- $\hat{+}$  Unit vector
- $\bullet$  Time derivative
- $*$  Complex conjugate; specific solution to retarded time equation

### Subscripts

- $i$  Component along the  $i^{\text{th}}$  coordinate axis  
(note:  $\hat{r}_i$  implies summation convention)
- $n$  Component in the direction of the surface normal
- $o$  Ambient condition
- $r$  Component in the direction of the radiation vector

### 2.5.4.1 Acoustic Formulation

The equation used to calculate the acoustic pressure time signature of a high speed propeller, derived in the subsonic propeller noise module, is

$$4\pi p'(\vec{x}, t) = \frac{1}{c} \frac{\partial}{\partial t} \int_{f=0} \left[ \frac{\rho_0 c v_n + 1}{r |1-M_r|} \right]_{\tau^*} dS + \int_{f=0} \left[ \frac{1}{r^2 |1-M_r|} \right]_{\tau^*} dS \quad (2.5-125)$$

Equation (2.5-125) contains a potentially singular term,  $|1-M_r|$ , in the denominator. As long as each point on the blade moves subsonically in the direction of the observer, the solution behaves nicely. In this case, Equation (2.5-125) can be written in the following computationally efficient form:

$$\begin{aligned} 4\pi p'_L(\vec{x}, t) &= \frac{1}{c} \int_{f=0} \left[ \frac{\hat{i}_i \hat{r}_i}{r(1-M_r)^2} \right]_{\tau^*} dS \\ &+ \int_{f=0} \left[ \frac{1_r - 1_i M_i}{r^2 (1-M_r)^2} \right]_{\tau^*} dS \\ &+ \frac{1}{c} \int_{f=0} \left[ \frac{1_r (r \hat{M}_i \hat{r}_i + c M_r - c M^2)}{r^2 (1-M_r)^3} \right]_{\tau^*} dS, \quad (2.5-126a) \end{aligned}$$

$$4\pi p'_T(\vec{x}, t) = \int_{f=0} \left[ \frac{\rho_0 v_n (r \hat{M}_i \hat{r}_i + c M_r - c M^2)}{r^2 (1-M_r)^3} \right]_{\tau^*} dS, \quad (2.5-126b)$$

and

$$p'(\vec{x}, t) = p'_L(\vec{x}, t) + p'_T(\vec{x}, t) \quad (2.5-126c)$$

where  $P_L^i$  and  $P_T^i$  denote the loading and thickness noise, respectively. The symbols  $M_i$ ,  $\dot{M}_i$ ,  $l_i$ , and  $\dot{l}_i$  are defined as follows:

$$M_i = \frac{v_i}{c} \quad (2.5-127)$$

$$\dot{M}_i = \frac{\dot{v}_i}{c} \quad (2.5-128)$$

$$l_i = p \hat{n}_i \quad (2.5-129)$$

$$\dot{l}_i = \dot{p} \hat{n}_i + p \hat{n}_i^{\times} \quad (2.5-130)$$

where

$$\hat{n}_i^{\times} = (\hat{\Omega} \times \hat{n})_i \quad (2.5-131)$$

When the radiation Mach number approaches or exceeds unity for a given point on the blade at a given time, then the following equation is used in place of Equation (2.5-126).

$$4\pi p_L^i(\vec{x}, t) = \frac{\partial}{\partial t} \int_{f=0}^g \int_{g=0}^r \frac{r}{r \sin\theta} d\Gamma d\tau + \int_{f=0}^g \int_{g=0}^r \frac{c}{r^2} \frac{r}{\sin\theta} d\Gamma d\tau \quad (2.5-132a)$$

$$4\pi p_T^i(x, t) = \frac{\partial}{\partial t} \int \frac{\rho_0 c v_n}{r \sin\theta} d\Gamma d\tau \quad (2.5-132b)$$

$$p_p^i(\vec{x}, t) = p_L^i(\vec{x}, t) + p_T^i(\vec{x}, t) \quad (2.5-132c)$$

where  $\theta$  is the angle between the blade normal,  $n$ , and the radiation vector,  $r$ . The symbol  $\Gamma$  is used to mean the locus of points on the blade that are

at a distance  $r$  from the observer at emission time  $\tau$ . In other words,  $\Gamma$  is the curve of intersection of the surfaces  $f = 0$  and  $g = \tau - t + (r/c) = 0$ .

The transonic propeller noise module uses both Equations (2.5-126) and (2.5-132) to calculate  $p'(\vec{x}, t)$ . Equation (2.5-126) is used whenever possible because the computer implementation is simple and efficient. A thorough discussion of these and other acoustic formulas for calculating propeller noise is contained in Reference 2.5-3.

#### 2.5.4.2 Frames of Reference

In this section the coordinate frames used in the acoustic calculations will be discussed. Basically there are two reference frames: the ground-fixed  $\vec{x}$ -frame and the blade-fixed  $\vec{\eta}$ -frame. The  $\vec{x}$ -frame, which remains fixed to the undisturbed medium, is set up as follows:

- The origin of this frame is at the propeller center at time  $t = 0$ .
- The propeller disk is in the  $x_1x_2$ -plane and the  $x_2$  axis coincides with the propeller axis, with positive direction in the flight direction.
- The  $x_1$ -axis is assumed to be orthogonal to  $x_3$ , upward and parallel to the plane of symmetry of the aircraft.
- The  $x_2$ -axis is defined in such a way that the  $\vec{x}$ -frame is right-handed.

The observer position is always specified in this frame as will be explained shortly.

The blade-fixed frame is set up as follows:

- The origin coincides with the origin of the  $\vec{x}$ -frame at  $t=0$ .
- The  $\eta_3$ -axis coincides with the  $x_3$ -axis.
- The  $\eta_2$ -axis lies on the pitch change axis of the blade.
- The  $\eta_1\eta_2$ -plane is normal to the propeller shaft.
- The  $\eta_1$ -axis is defined to make the  $\vec{\eta}$ -frame right-handed.

Note that the blade mean surface does not lie in the  $\eta_1\eta_2$ -plane but makes an angle equal to the geometry pitch angle at each radial position. Note also that the  $\vec{\eta}$ -frame rotates with angular velocity  $\Omega$ .



In the acoustic formulas presented in the previous section, many operations involving dot products or cross products of two vectors are required. When the products involve blade normals or surface pressures, it is convenient to transform the components of these vectors to the  $\vec{\eta}$ -frame and then perform the operation. However, note that the source position  $\vec{y}$  always refers to a source point in  $\vec{x}$ -frame with a known position  $\vec{\eta}$  in the blade-fixed frame. The quantities  $\hat{r} = \vec{x} - \vec{y}$  and  $\hat{r} = \vec{r}/r$  are calculated in the  $\vec{x}$ -frame, and then the vector components are calculated in the  $\vec{\eta}$ -frame. Note that for any variable which has a velocity term, such as  $M_r = M_i \hat{r}_i$  and  $v_n = \vec{v} \cdot \hat{n}$ , the velocity is with respect to the  $\vec{x}$ -frame. In other words

$$v_i = \frac{\partial y_i}{\partial \tau} \quad (2.5-133)$$

The components of this vector must then be calculated in the  $\vec{\eta}$ -frame to get  $M_r$  and  $v_n$ .

Another frame of reference which is used conceptually is the aircraft-fixed, nonrotating  $\vec{X}$ -frame. This frame coincides with the ground-fixed  $\vec{x}$ -frame at  $t=0$ . The acoustic pressure  $p'(\vec{X}, t)$  for an observer in the aircraft-fixed frame with the position vector  $\vec{X}$  is found as follows: Since both  $\vec{X}$  and  $t$  are known, the relative position of the ground-fixed frame and the aircraft-fixed-frame is known. From this, calculate the observer position  $\vec{x}$  in the ground-fixed frame. Using this  $\vec{x}$  and  $t$ , calculate  $p'(\vec{x}, t)$  which is the same as the acoustic pressure at position  $\vec{X}$  in the aircraft-fixed frame and at time  $t$ . Thus, the  $\vec{X}$ -frame itself is never used in acoustic calculations, even though the pressure signal which results is  $p'(\vec{X}, t)$ .

The transformation from one frame to the other at time  $t$ , can be written in matrix notation as

$$\begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{bmatrix} \quad (2.5-134a)$$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ V_F t \end{bmatrix} \quad (2.5-134b)$$

where  $(\eta_1, \eta_2, \eta_3)$  is a point in the rotating  $\vec{\eta}$ -frame,  $(X_1, X_2, X_3)$  is the corresponding point in the moving  $\vec{X}$  frame,  $(x_1, x_2, x_3)$  is the corresponding point in the ground-fixed frame, and  $\psi(=\Omega t)$  is the angle between the  $X_1$  and  $\eta_1$  axes.

A final coordinate transformation is necessary to account for changes in the root pitch setting. This rotation through a small angle,  $\theta_0$ , is applied to the blade surface coordinates which are calculated by the blade shape module.

$$\vec{\eta}(\xi_1, \xi_2) = \begin{bmatrix} \cos\theta_0 & 0 & \sin\theta_0 \\ 0 & 1 & 0 \\ -\sin\theta_0 & 0 & \cos\theta_0 \end{bmatrix} \begin{bmatrix} \eta_1'(\xi_1, \xi_2) \\ \xi_1 \\ \eta_2'(\xi_1, \xi_2) \end{bmatrix} \quad (2.5-135)$$

Whenever the  $\vec{\eta}$ -frame or pitch axis coordinate system is mentioned herein, it is assumed that the above correction transformation has occurred. Often the blade shape module and transonic propeller noise module will be executed without change to the collective pitch setting. Thus,  $\theta_0=0$  and the last transformation is unnecessary.

#### 2.5.4.3 Computational Strategy

Three elements that influence the computational strategy for noise prediction are numerical accuracy, computational efficiency and conceptual simplicity. The trade-offs are never straightforward but the simplest scheme which gives numerically stable answers in a reasonable amount of computer time is desired. A good balance is particularly crucial to the transonic propeller noise module. Moreover, it is important that the transonic and subsonic propeller noise methods give identical results for propellers with subsonic Mach numbers. These thoughts guide the computational strategy.

The first step in computing an acoustic pressure time history is to discretize one full revolution of the blade into even time increments,  $t_i$ , and to represent the blade by a set of source points,  $\vec{\eta}(j)$ . Even though the tip of the blade is moving supersonically, some portion of the blade is subsonic. It will be helpful to define a subset of source points,  $\vec{\eta}_s$ , that move subsonically. That is

$$\vec{\eta}_s = \{[\eta_1, \eta_2, \eta_3] : \frac{(\Omega \eta_2)^2 + v_f^2}{c^2} < 1.0\} \quad (2.5-136)$$

For each time point, the noise produced by the part of the blade represented by  $\vec{\eta}_s$  can be calculated using Equation (2.5-126). The numerical evaluation of Equation (2.5-126) is discussed in detail in the subsonic propeller noise module. The noise produced by the supersonic tip of the blade is calculated by either Equation (2.5-126) or Equation (2.5-132), depending on the value of  $M_r$  for each source point. If  $M_r$  exceeds some transitional value (usually 0.98) for any source point, then Equation (2.5-132) is used for the entire supersonic tip at that time. In practice, Equation (2.5-132) is used for less than one-fourth of the time points even in the worst case.

Equation (2.5-132) is evaluated by dividing the blade tip into panels and by defining a set of possible emission times,  $\{\tau_j\}$ . Each integral  $\iint [\Gamma] d\Gamma d\tau$  is approximated by applying a one-dimensional trapezoid rule as  $\tau_j$  varies. The partial derivative with respect to source time,  $t$ , is approximated by using a central difference formula after the integral over the supersonic tip has been calculated at times  $(t_i + \Delta t)$  and  $(t_i - \Delta t)$ .

The execution time required to compute Equation (2.5-132) is greatly influenced by the set of emission times,  $\tau_j$ , that are chosen. If the span of all possible emission times is divided evenly, a very large number of times will be required. A better plan is to determine one, two, or three time intervals during which the  $\Gamma$  curve intersects the current panel. Each of these intervals can then be divided evenly. The time intervals themselves can be determined by solving the retarded time equation at the four corners of the panel. This gives at least four and at most twelve times,

$\tau_j^*$ , at which the  $\Gamma$  curve is entering or leaving the panel. The time interval  $[\tau_j^*, \tau_{j+1}^*]$  will contribute to the integral over the current panel if the  $\Gamma$  curve is inside the panel during that time. It is sufficient to test the location of the curve at the median time,  $(\tau_j^* + \tau_{j+1}^*)/2$ . If the distance from the observer to the  $\Gamma$  curve is in the same range as the distance from the observer to each of the four corners of the panel then the time interval will contribute.

There are several sources of error in this prediction method which the user must understand. The most important involves the geometric model of the blade. The blade position vectors,  $\vec{\eta}$ , and normal vectors,  $\vec{n}$ , describe the blade shape very smoothly and accurately. Each panel of the supersonic tip is actually a curved surface. However, since the trapezoid rule is used to approximate the integrals and since linear interpolation is used to calculate the integrands at points along the edges of the panel, computationally the curved panel is replaced by a flat plate. It is very important that the user supply grid which has a large number of nodes in those regions where the blade is highly curved. One good way to accomplish this is to use the same chordwise grid that was used for the blade shape module but add extra points near the leading edge. It may be necessary to add extra spanwise grid points as well. It is wise to have at least five spanwise grid points in the subsonic region and five in the supersonic tip. Finally, the user must take care to specify enough time points to accurately represent the pressure time history. Unfortunately, the computation time will increase dramatically as the number of time points and grid points increases.

#### 2.5.4.4 Retarded Time Equation

As discussed in the subsonic propeller noise module, the retarded time equation can be written in the form

$$f(\phi) = A\phi^2 + B\phi + C + \cos(\phi+D) = 0 \quad (2.5-137)$$

where only the roots  $(\phi/\Omega) < 0$  are of interest. Figure 2.5-18 is a typical graph of the retarded time equation that often results when the source points are moving at close to sonic speeds. It is not clear from the

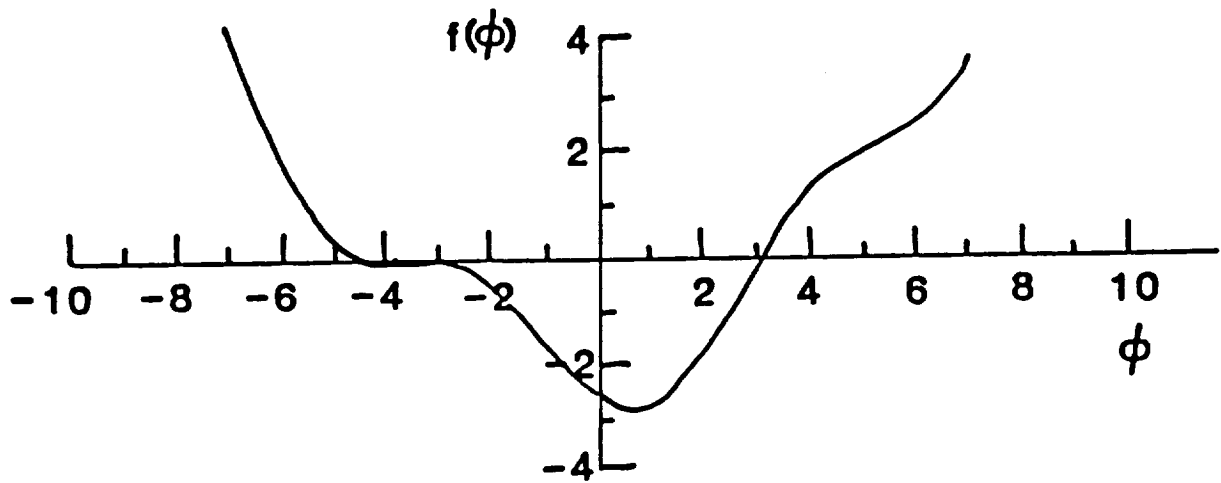


Figure 2.5-18. Graph of Typical Transonic Retarded Time Equation

figure whether this graph has one or more negative roots. In order to numerically solve for the roots using Newton's method, the number and approximate location of the roots must be determined.

Notice that Equation (2.5-137) could be easily solved if the cosine term were absent. The first step in finding approximate roots is to set  $\cos(\phi + D) = -1$  and solve for the negative root,  $\phi_0$ . The location of the actual roots must be in the interval  $[\phi_0, 0]$  because the cosine term cannot have a value smaller than -1. Next, approximate the cosine curve by a parabola which has the same minimum point and the same intercepts, and solve again. The resulting approximate retarded time equation may have one, two, or no real roots in the interval  $[\phi_0, 0]$ . If there is only one root to the approximate equation then it is a good initial guess at the one negative root to Equation (2.5-137). If there are no real roots to the approximate equation, then  $\phi_0$  will be a good initial guess to the one root of Equation (2.5-137). If, however, there are two roots to the approximate equation then the situation must be further clarified.

Figure 2.5-19 is an exploded view of the graph in Figure 2.5-18. Note that there are three roots in this particular case. The best way to find them is to first find the local minimum and maximum points on the curve. That is, solve

$$\frac{df}{d\phi} = 2A\phi + B - \sin(\phi+D) = 0 \quad (2.5-138)$$

Approximate solutions,  $\phi_1$  and  $\phi_2$ , to Equation (2.5-138) are adequate. If  $f(\phi_1)$  and  $f(\phi_2)$  have opposite signs, there is a root of  $f$  between them. If it is determined that three roots exist, then  $(\phi_0+\phi_1)/2$ ,  $(\phi_1 + \phi_2)/2$  and  $\phi_2/2$  will be very good initial guesses for use by Newton's method.

#### 2.5.4.5 TPN Input

The computation of transonic propeller noise requires definition of the blade shape characteristics, aerodynamic characteristics, flow and operating parameters, and observer coordinates. The noise signature is calculated using two different full blade formulations, depending on the radiation Mach number of the blade section. The input is consistent with

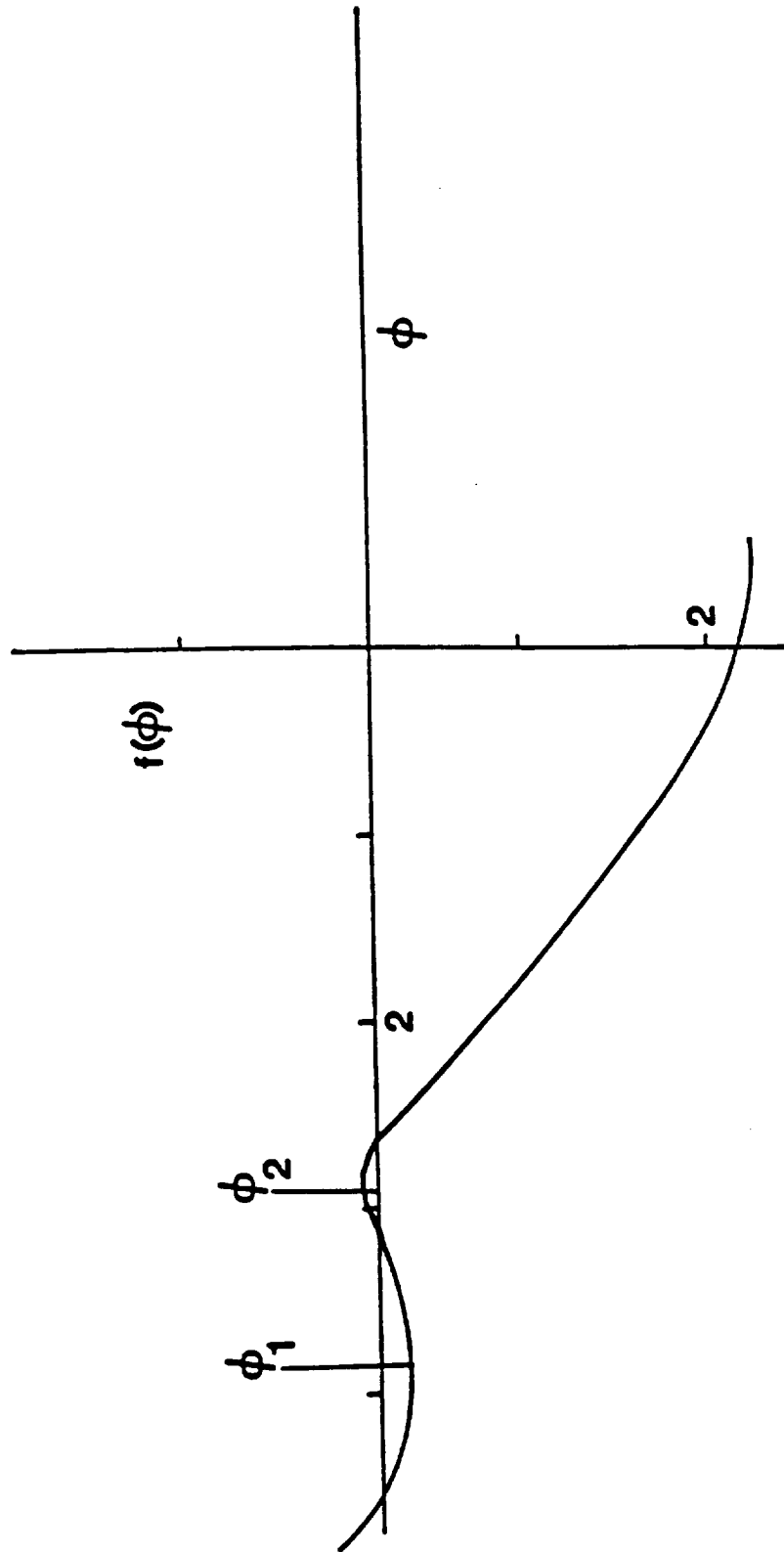


Figure 2.5-19. Exploded View of Retarded Time Equation

that for the subsonic propeller noise module and the two modules will give identical results for subsonic cases.

#### User Parameters

B	Blade length measured from axis to tip, m (ft)
$N_b$	Number of blades
$N_s$	Highest harmonic number desired
$M_z$	Aircraft Mach number
$\theta_0$	Change in root pitch angle, rad
$\psi_0$	Initial azimuth angle of first blade, rad
$\Omega$	Angular velocity of blade, rad/s

#### Blade Shape Table

$\xi_1$	Spanwise location, normalized with respect to B
$\xi_2$	Chordwise location, rad
$\eta_1(\xi_1, \xi_2)$	Blade surface abscissa, normalized with respect to B
$\eta_3(\xi_1, \xi_2)$	Blade surface ordinate, normalized with respect to B
$\phi_1, \phi_2$	Slopes of basis functions at nodes

#### Aerodynamic Characteristics Table

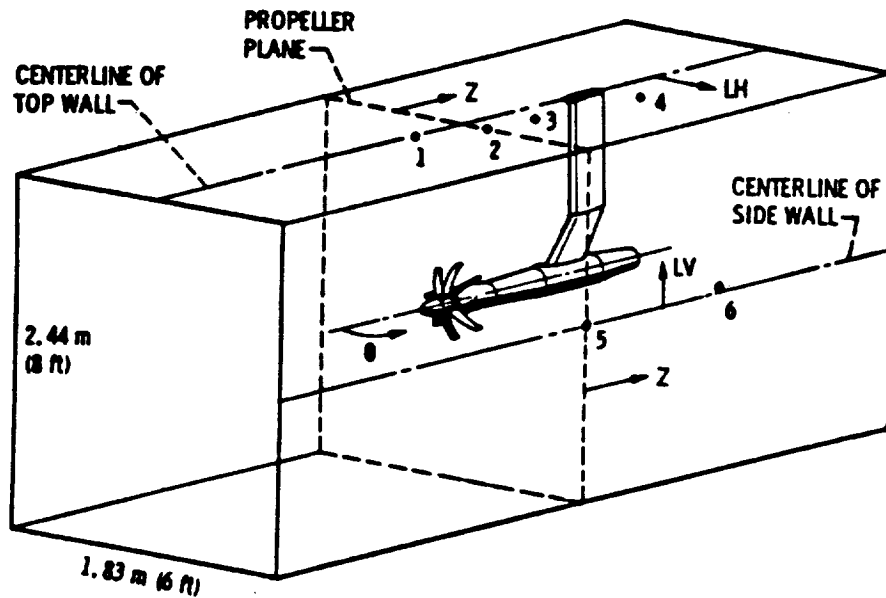
$\xi_1$	Spanwise location, normalized with respect to B
$\xi_2$	Chordwise location, rad



#### 2.5.4.6 TPN Program Validation

The TPN program has been verified for near-field conditions by comparison with experimental data taken in the NASA-Lewis 8' x 6' wind tunnel for an eight-bladed SR-3 propeller. Pressure signatures and overall noise spectra at four different near-field stations (Figure 2.5-20) are shown in Figures 2.5-21 through 2.5-28 for an advance Mach number of 0.80, a helical tip Mach number of 1.14 and a power coefficient of 1.71, and an advance ratio of 3.06. Overall sound pressure levels calculated at the four stations are compared with data for Figure 2.5-29. The 11- and 13-station blade description notations in Figures 2.5-21 through 2.5-28 refer to the number of spanwise stations utilized to calculate blade loading, and the parameter  $a$ , defines the assumed camber-line loading distribution (see Figure 2.5-30). Note that the results for the pressure signature at Station No. 1 (Figure 2.5-21) does not show a single sinusoidal-type cyclic behavior over one period as noted at the other stations (Figures 2.5-23, 2.5-25, 2.5-27) and theoretically anticipated. This is due to difficulties with the iterative method utilized in TPN to solve the retarded time equation. Instances in which this method results in an incorrect solution are shown in Figures 2.5-31 and 2.5-32. Figure 2.5-31 exhibits a situation wherein the initial guess for the solution was -3.40312 and the method picks the oscillating root at -7.8999 as the solution after the iteration limit is reached, whereas the correct solution root is -6.40. Figure 2.5-32 depicts a case where, even though the initial guess for the solution was -3.7717, the method yields a final solution of +4.2245, whereas the graphical solution is shown to be -5.55. Clearly, refinement of the solution method is needed. Note, however, that even though the pressure signature is incorrectly predicted at Station 1, the first-harmonic sound-pressure level is within about 2% of the measured data (Figure 2.5-22).

Difficulties in solving the retarded-time equation were not encountered at Stations 2, 3, and 4, and these results (Figures 2.5-23 through 2.5-28) show well-behaved pressure signatures and good-to-excellent agreement with noise-spectra data.



TRANSDUCER POSITIONS, cm/in.

POSITIONS	1		2		3		4	
Z	-27.7	-10.9	0.953	0.375	45.2	17.8	104.4	41.1
LH	2.54	1.0	10.2	4.0	7.62	3.0	31.5	12.4
LV	121.9	48.0	121.9	48.0	121.9	48.0	121.9	48.0
NOMINAL ANGLE, $\theta$	77°		90°		110°		130°	

Figure 2.5-20. Pressure Transducer Positions SR-3 Propeller

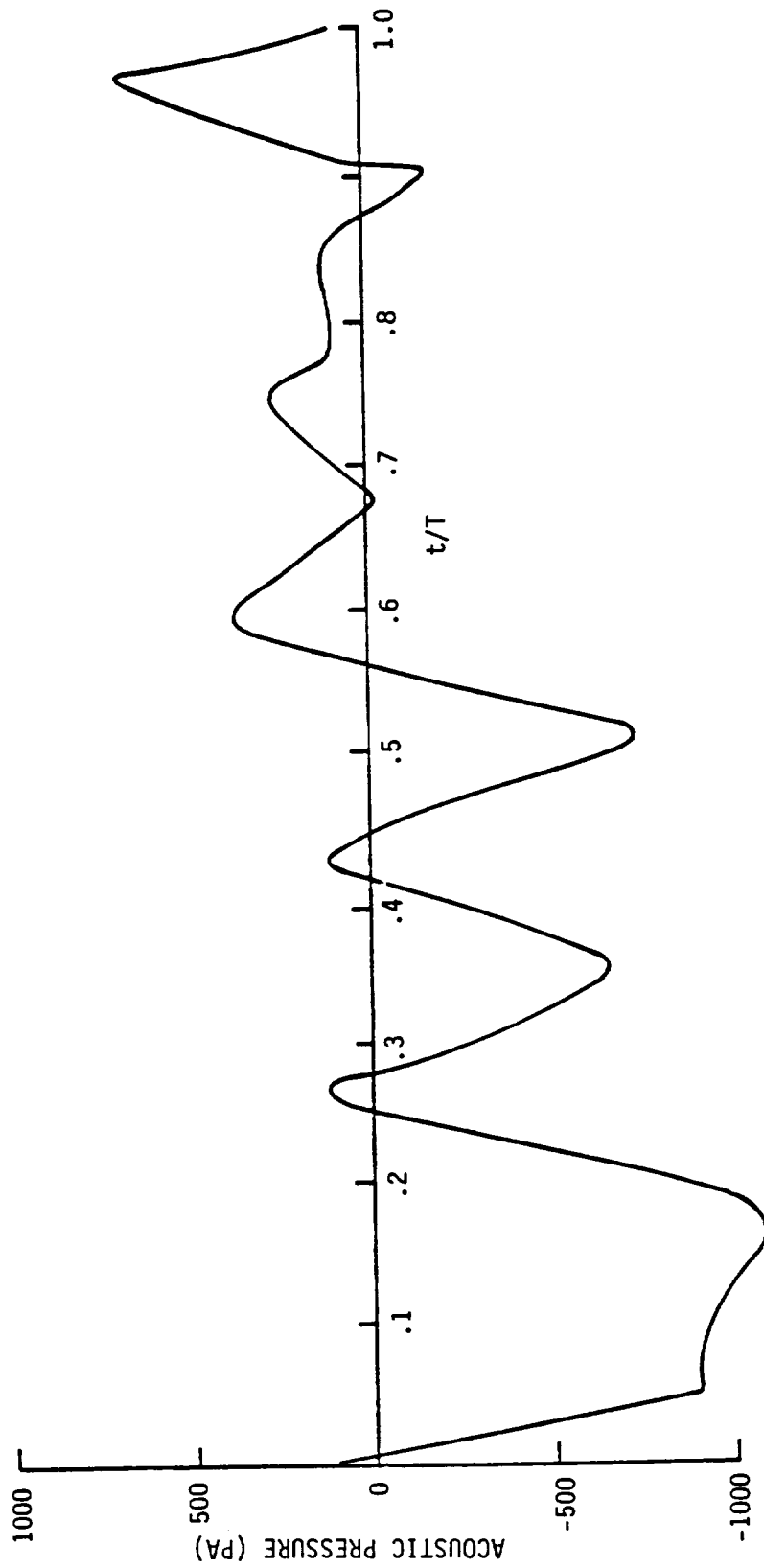


Figure 2.5-21. Acoustic Pressure Signature SR-3 Propeller Near Field  
 $M_0 = 0.80$ ,  $J = 3.06$ ,  $C_p = 1.71$ ,  $B = 8$ ,  $MTH = 1.14$   
 $x_1$ ,  $x_2$ ,  $x_3 = -1.219$ ,  $0.0254$ ,  $0.277$  (Station No. 1)  
 TPN code,  $a = 1.0$  blade loading  
 13 station blade description

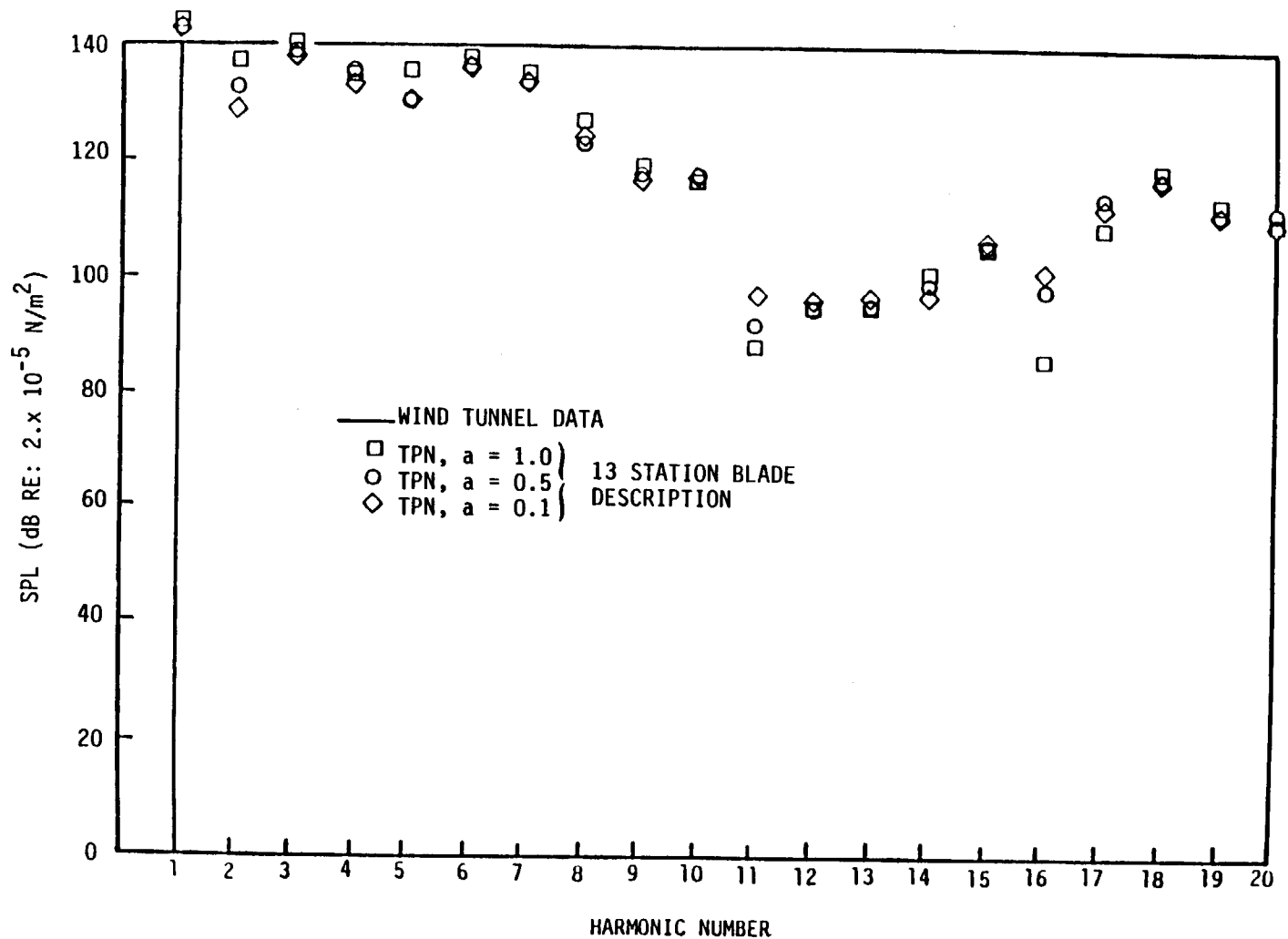


Figure 2.5-22. Overall Noise Spectra SR-3 Propeller Near Field  
 $M_0 = 0.8$ ,  $J = 3.06$ ,  $C_p = 1.71$ ,  $B = 8$ ,  $M_{TH} = 1.14$   
 Transducer Station No. 1

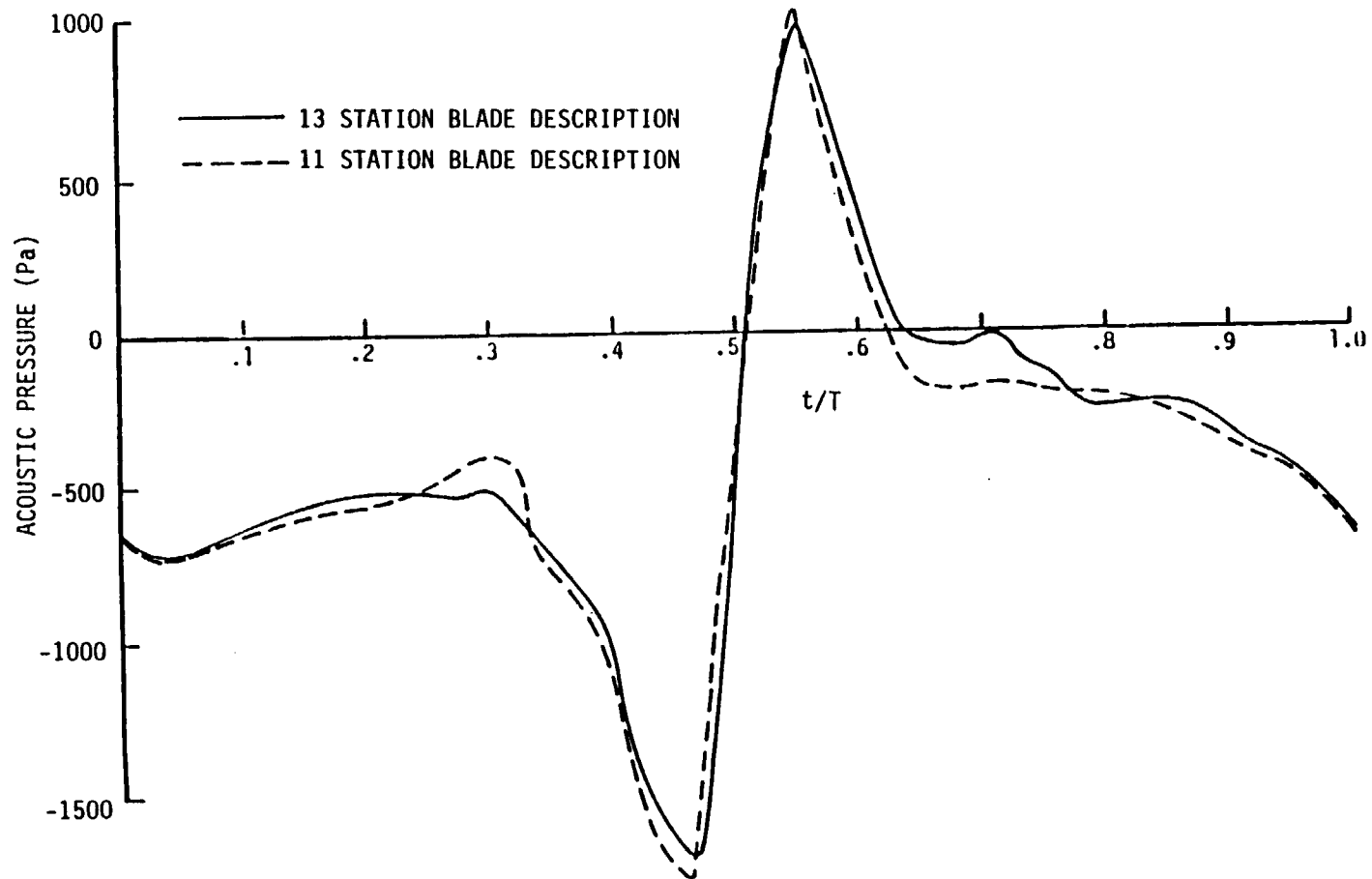


Figure 2.5-23. Acoustic Pressure Signature SR-3 Propeller Near Field  
 $M_0 = 0.80$ ,  $J = 3.06$ ,  $C_D = 1.71$ ,  $B = 8$ ,  $M_{TH} = 1.14$   
 $x_1, x_2, x_3 = -1.219, 0.102, -0.01$  m (Station No. 1)  
TPN code,  $a = 1.0$  blade loading

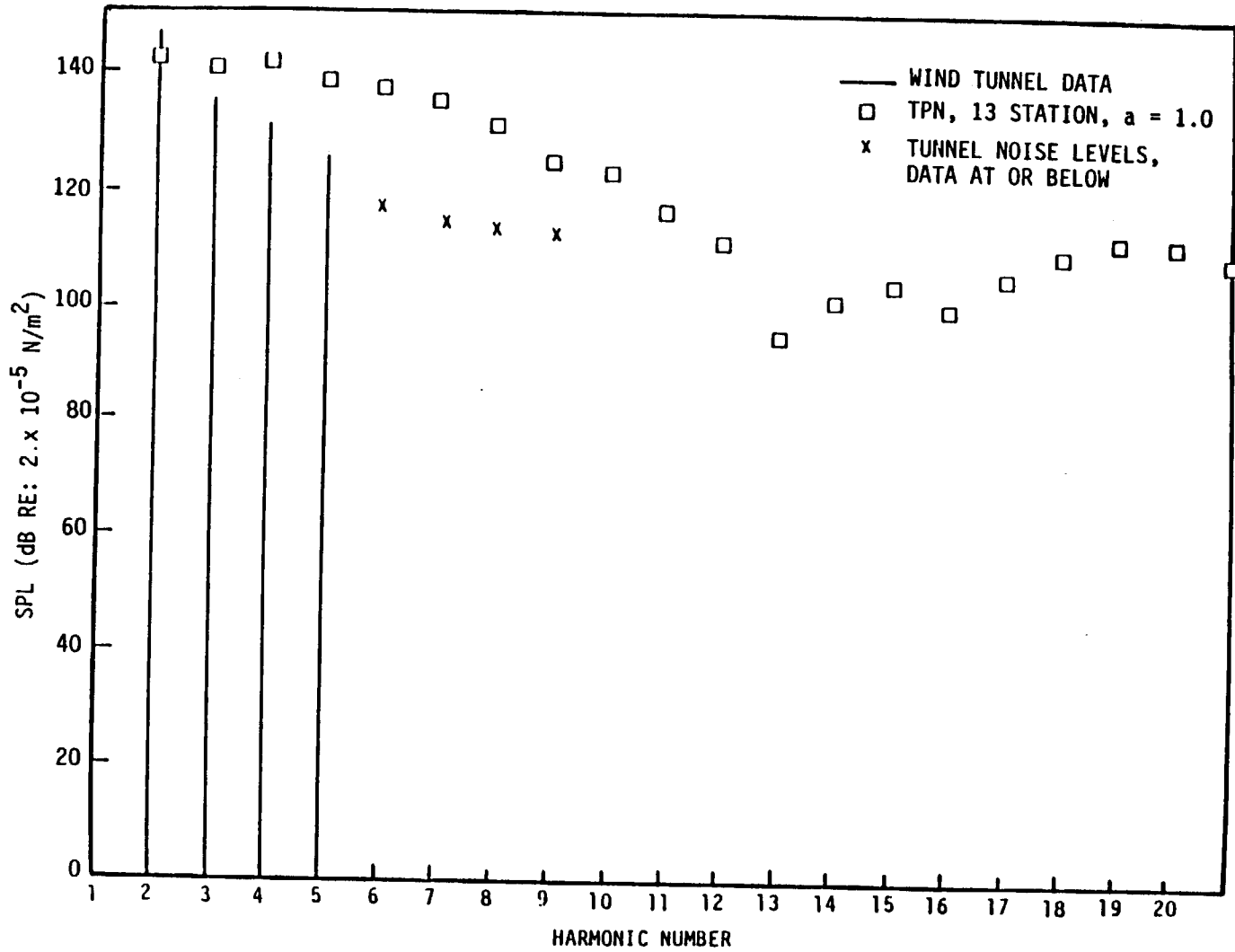
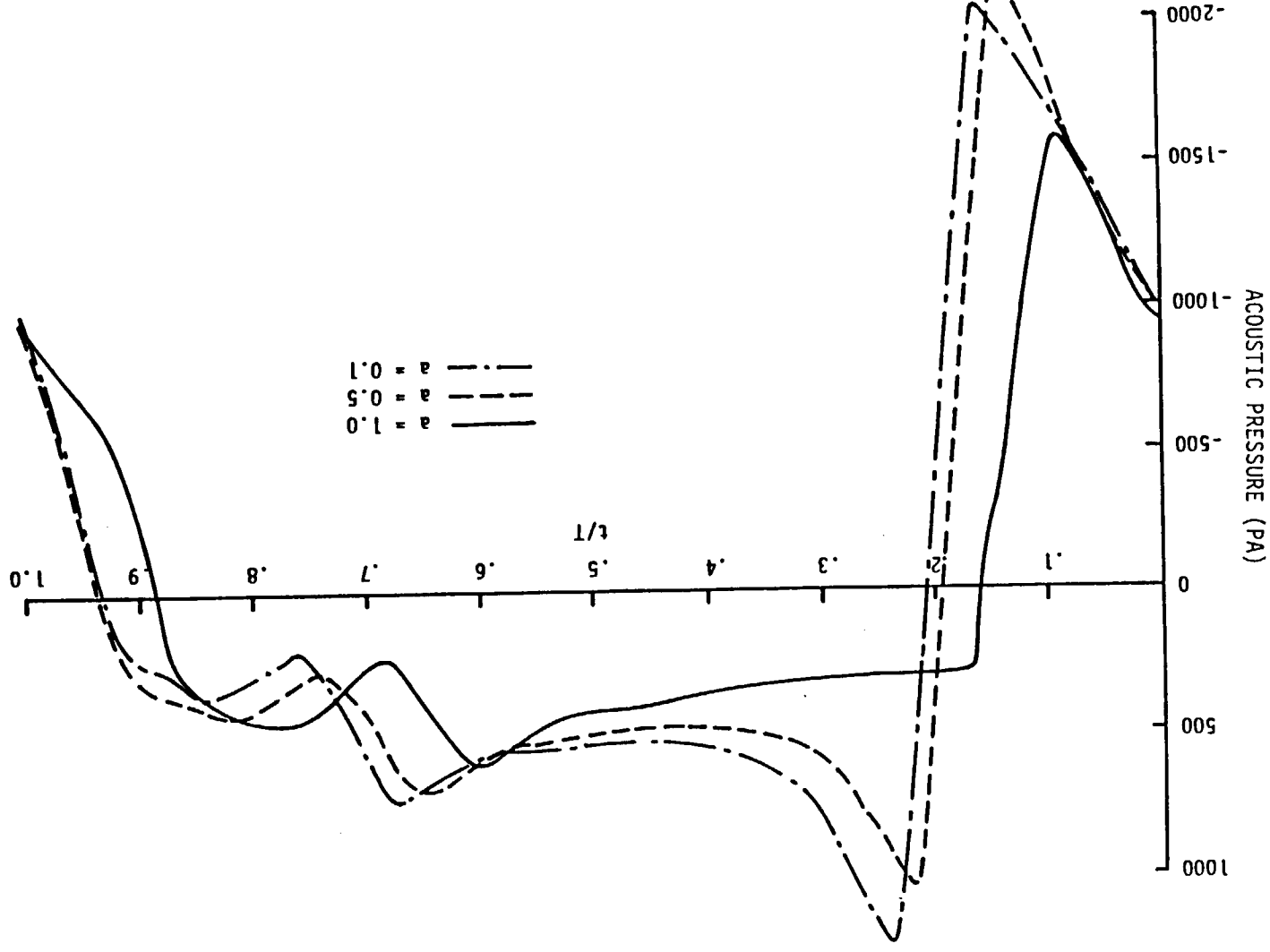


Figure 2.5-24. Overall Noise Spectra SR-3 Propeller Near Field  
 $M_0 = 0.80$ ,  $J = 3.06$ ,  $C_p = 1.71$ ,  $B = 8$ ,  $M_{TH} = 1.14$   
Transducer Station No. 2



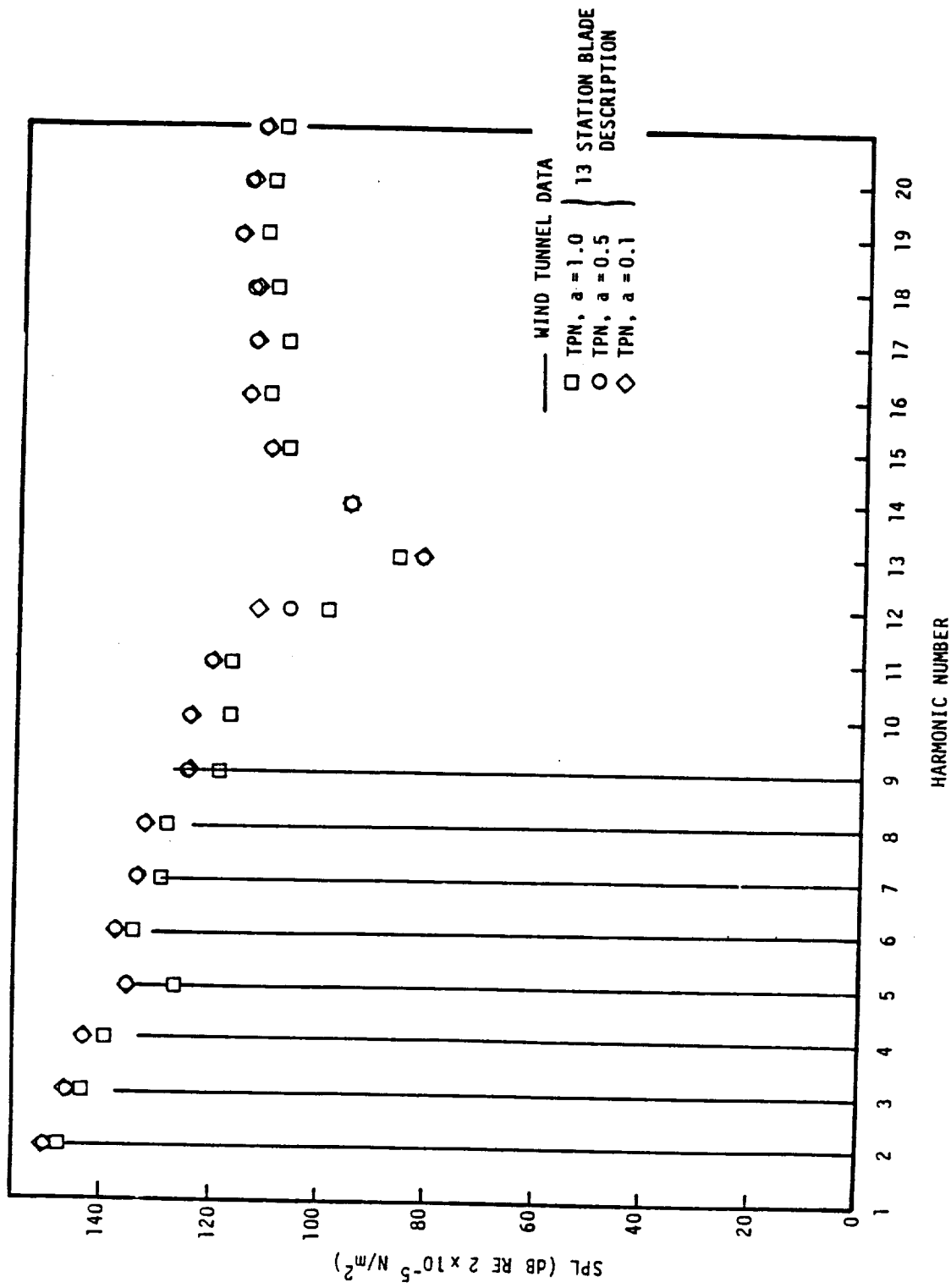


Figure 2.5-26. Overall Noise Spectra SR-3 Propeller Near Field  
 $M_0 = 0.80$ ,  $J = 3.06$ ,  $C_p = 1.71$ ,  $B = 8$ ,  $MTH = 1.14$   
 Transducer Station No. 3



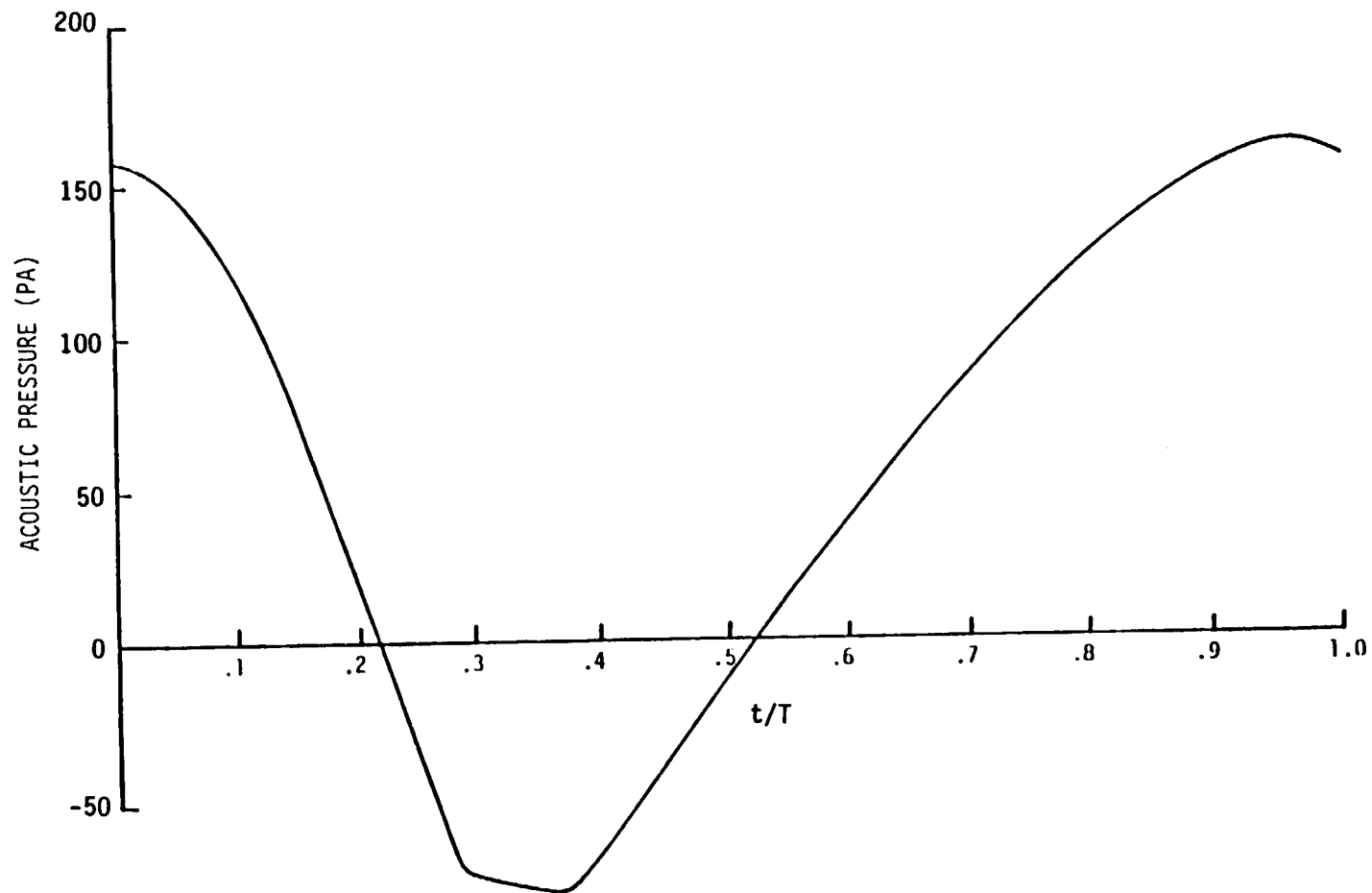


Figure 2.5-27. Acoustic Pressure Signature SR-3 Propeller Near Field  
 $M_0 = 0.80$ ,  $J = 3.06$ ,  $C_p = 1.71$ ,  $B = 8$ ,  $M_{TH} = 1.14$   
 $x_1, x_2, x_3 = -1.219, 0.315, -1.044$  m (Station No. 4)  
TPN code,  $a = 1.0$  blade loading  
13 and 11 station blade descriptions

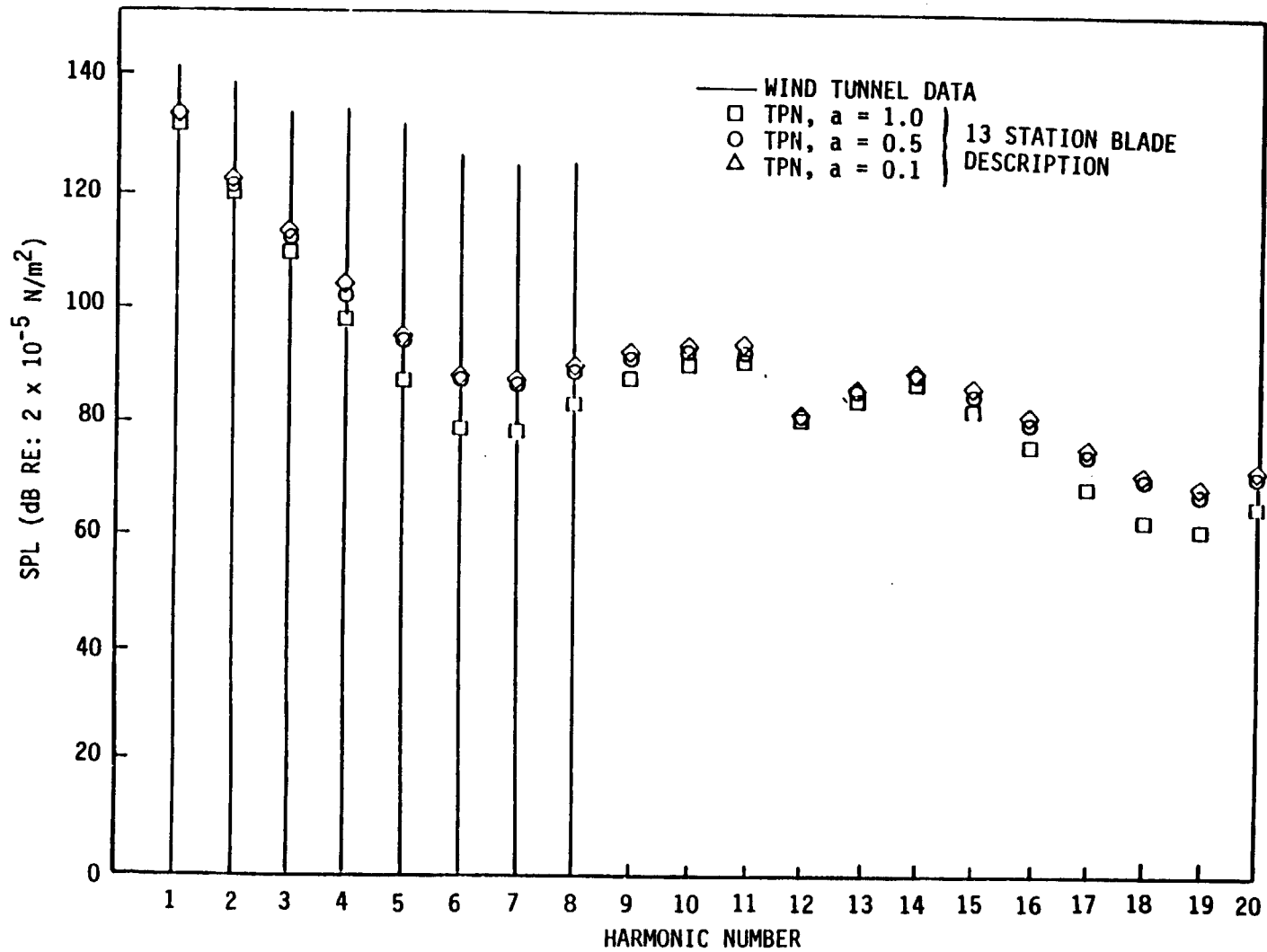


Figure 2.5-28. Overall Noise Spectra SR-3 Propeller Near Field  
 $M_0 = 0.80$ ,  $J = 3.06$ ,  $C_D = 1.71$ ,  $B = 8$ ,  $M_{TH} = 1.14$   
 Transducer Station No. 4

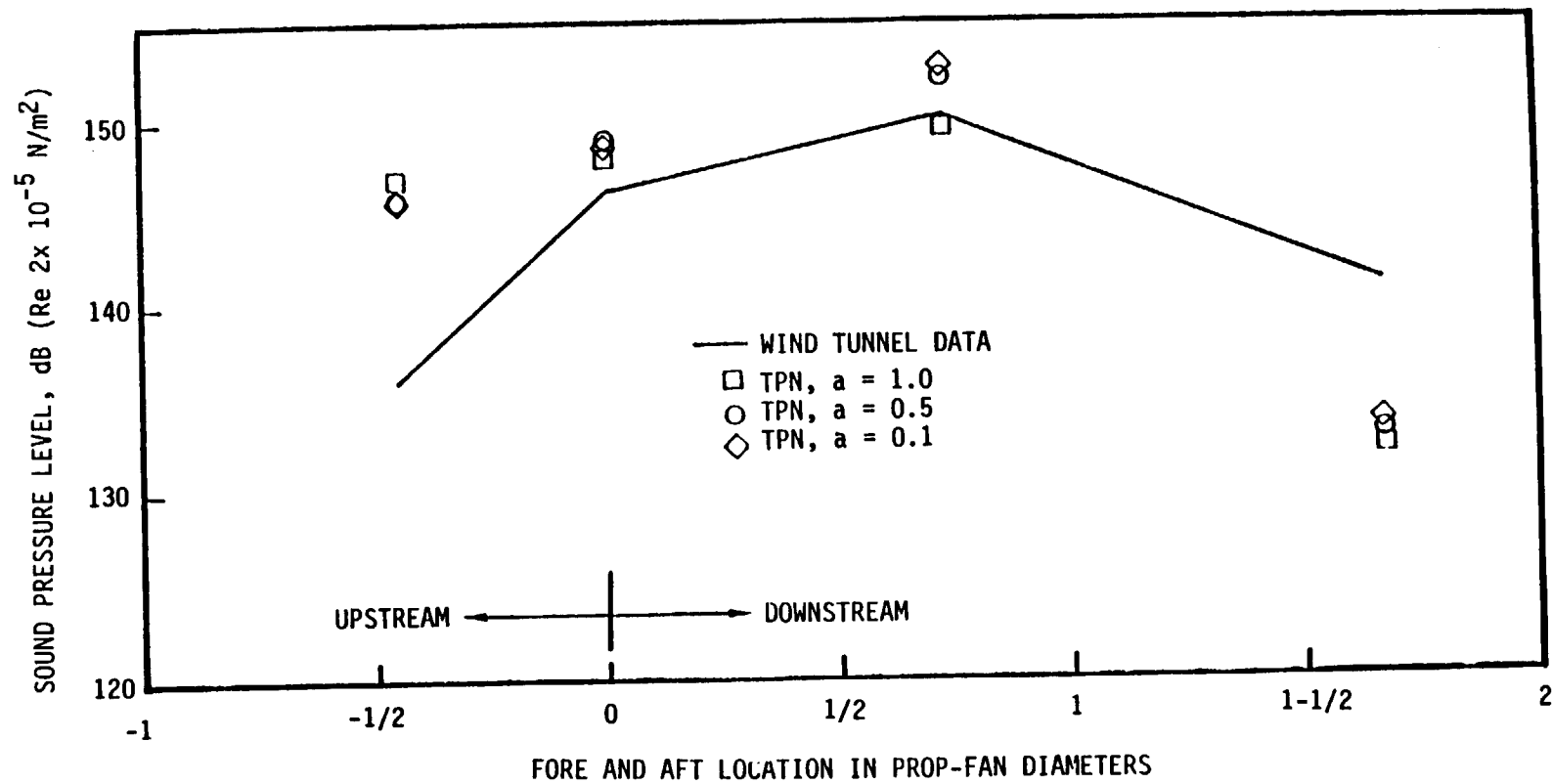


Figure 2.5-29. Overall Sound Pressure Level SR-3 Propeller Near Field  
 $M_0 = 0.80$ ,  $B = 8$ ,  $J = 3.06$ ,  $C_p = 1.71$ ,  $n = 8495$  rpm,  $M_{TH} = 1.14$

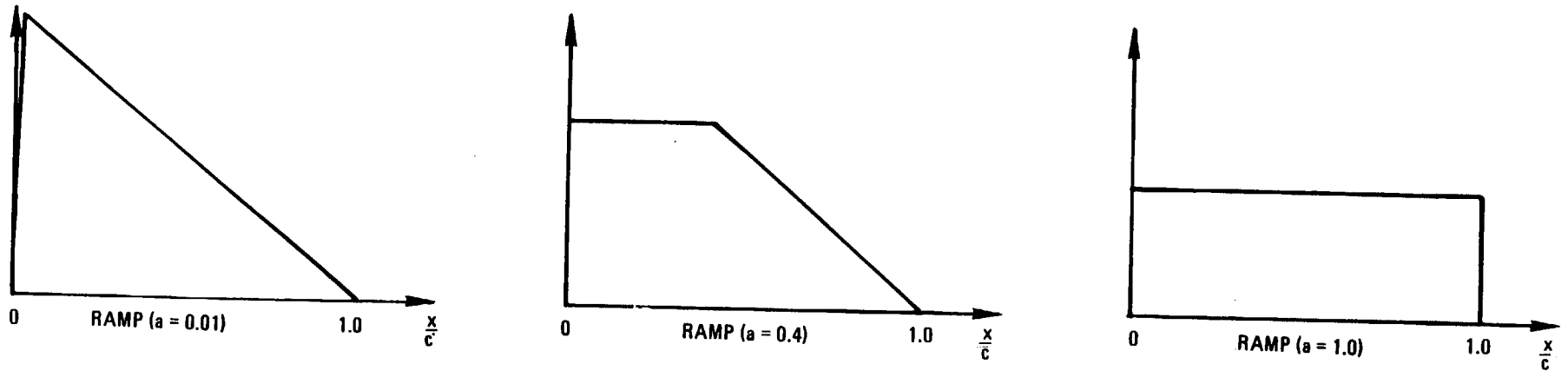


Figure 2.5-30. Definition of Loading Parameter, a

01-168-86

$$F(x) = Ax^2 + Bx + C + \cos(x+D) = 0$$

$$A = 0.787974$$

$$B = 0.253942$$

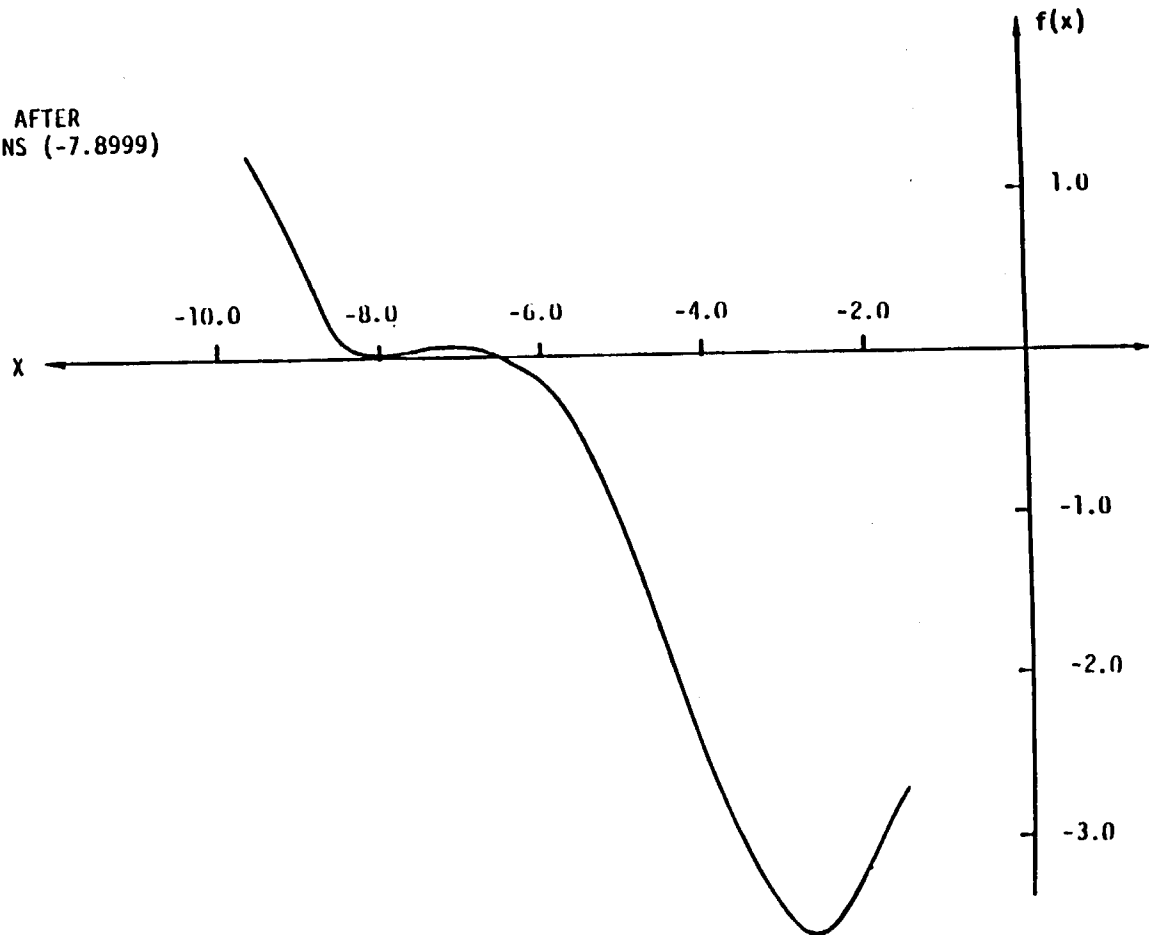
$$C = -2.492615$$

$$D = -.3788825$$

INITIAL GUESS = -3.40312

FINAL SOLUTION BY TPN = NO SOLUTION AFTER  
20 ITERATIONS (-7.8999)

GRAPHIC SOLUTION = -6.40



2-219

Figure 2.5-31. Retarded Time Equation

$$f(x) = Ax^2 + Bx + C + \cos(x+D) = 0$$

$$A = .070389$$

$$B = 0.26577$$

$$C = -2.28989$$

$$D = .398626$$

$$\text{INITIAL GUESS} = -3.7717$$

$$\text{FINAL SOLUTION} = 4.2245 \text{ BY TPN}$$

$$\text{GRAPHIC SOLUTION} = -5.55$$

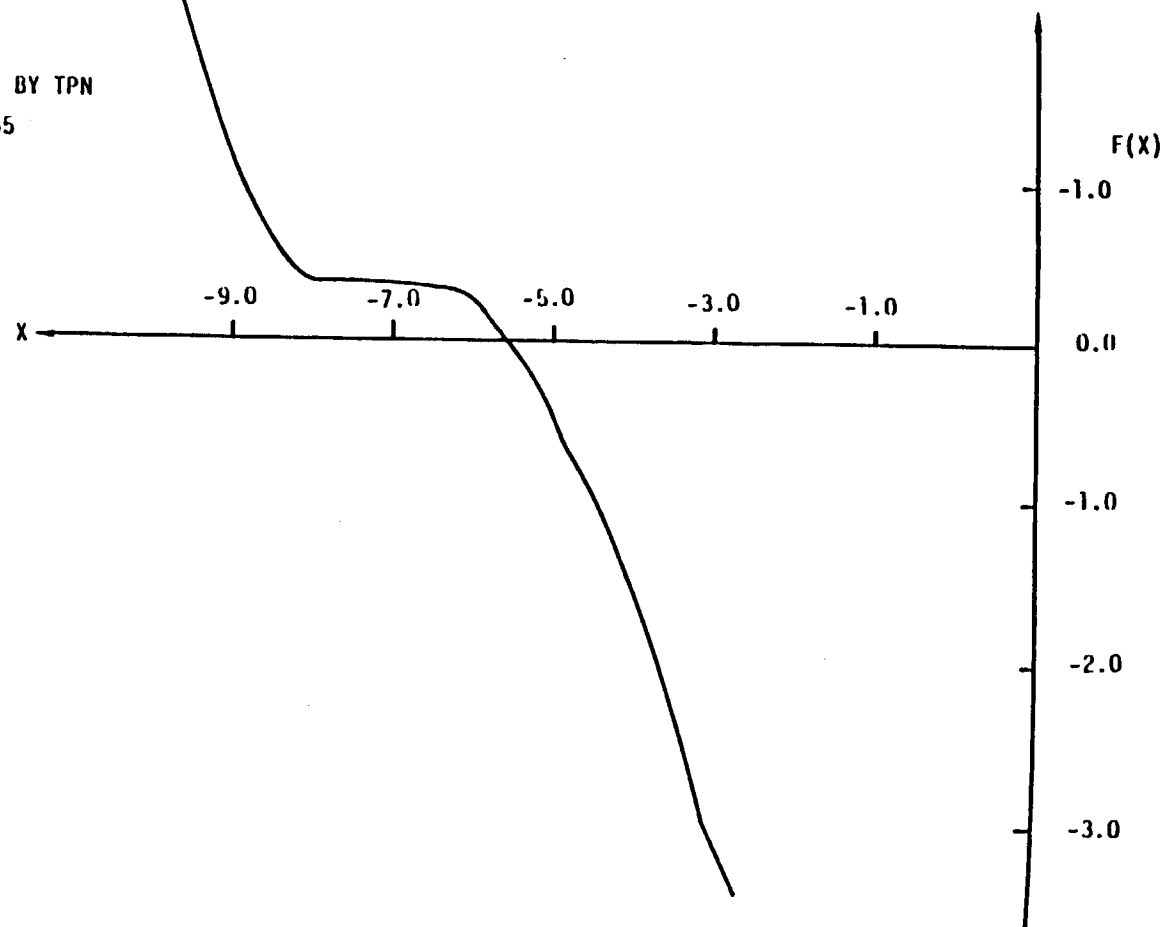


Figure 2.5-32. Retarded Time Equation

In addition, the overall sound-pressure levels predicted by TPN (Figure 2.5-29) show good-to-fair agreement with data depending on station location, and are relatively insensitive to blade-loading parameter. Errors of 1% - 2% may be noted at the first two stations downstream of the propeller plane (Stations 2 and 3, Figure 2.5-20), whereas errors on the order of 7% are noted at the upstream and far downstream station (Stations 1 and 4, respectively).

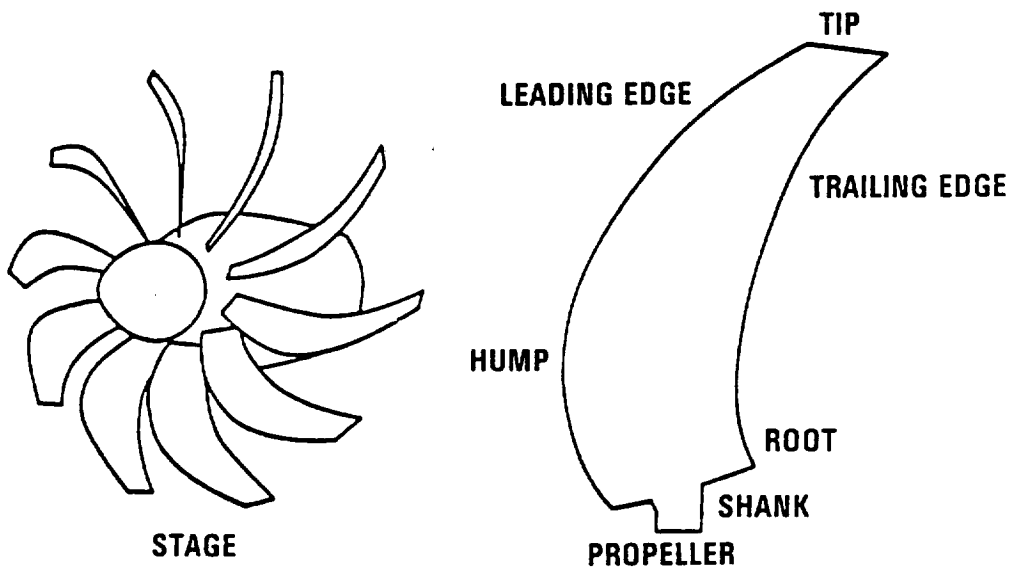
## 2.6 STRUCTURAL BEAM MODULE

The free vibration analysis of a generally curved and pretwisted rotating blade is a complex problem that requires the development of a nonlinear analytical model. For conventional blades, and advanced blades (advanced prop-fan, Figure 2.6-1), this model must be general enough to account for an arbitrary amount of blade curvature and pretwist, as well as blade cross sections with general prismatic shape. Allowances must also be included to account for changes in the blade orientation at the hub. These changes, which will alter the centrifugal forces of the blade, include a pitch setting for conventional propellers and a general pitch setting about an arbitrary vector for advanced prop-fans (i.e., pitch setting is a linear combination of pitch, precone, and sweep). These blades which are quite flexible require a nonlinear theory for deriving the equations of motion in order to properly account for the stiffening effects from the centrifugal forces.

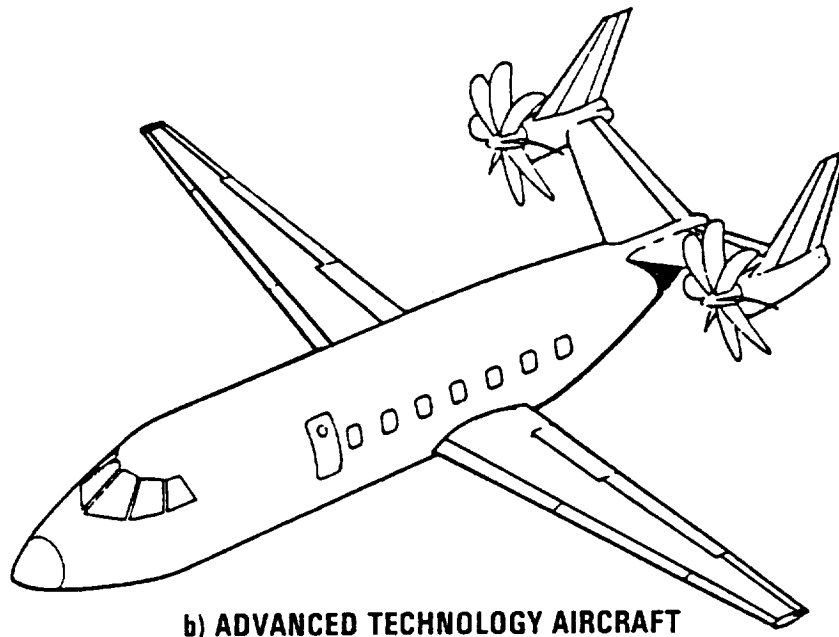
Due to the complexities of this problem, an exact solution cannot be determined so approximate methods (for example, Ritz, Galerkin, finite element) must be used. The finite element method was chosen because it readily lends itself to the modeling of a curved and twisted blade. The other methods were unattractive because they either required knowing approximate deflection functions (Ritz method) or required deriving the full nonlinear partial differential equations of motion and natural boundaries of the (Galerkin's method).

The finite element method, which is based on variational principles, is applied by dividing the blade up into subregions, calculating the total dynamic potential ( $U - T - W_e$ ) for each subregion, and then deriving the finite elements by taking the variation of each subregion (Hamilton's principle). In this development, each subregion is defined with straight beam type finite elements which are located along the curved line of shear centers of the blade, Figure 2.6-2. The curved line of shear centers of the blade must first be calculated (for example, using the method in Section 2.6.8) before this analysis can be done. The finite elements for each straight beam are derived with allowances for arbitrary cross section shape, beam pretwist, cross section warping (St. Venant type warping





**a) TURBOPROP STAGE AND PROPELLER**



**b) ADVANCED TECHNOLOGY AIRCRAFT**

Figure 2.6-1. Advanced Propeller (Prop-Fan)

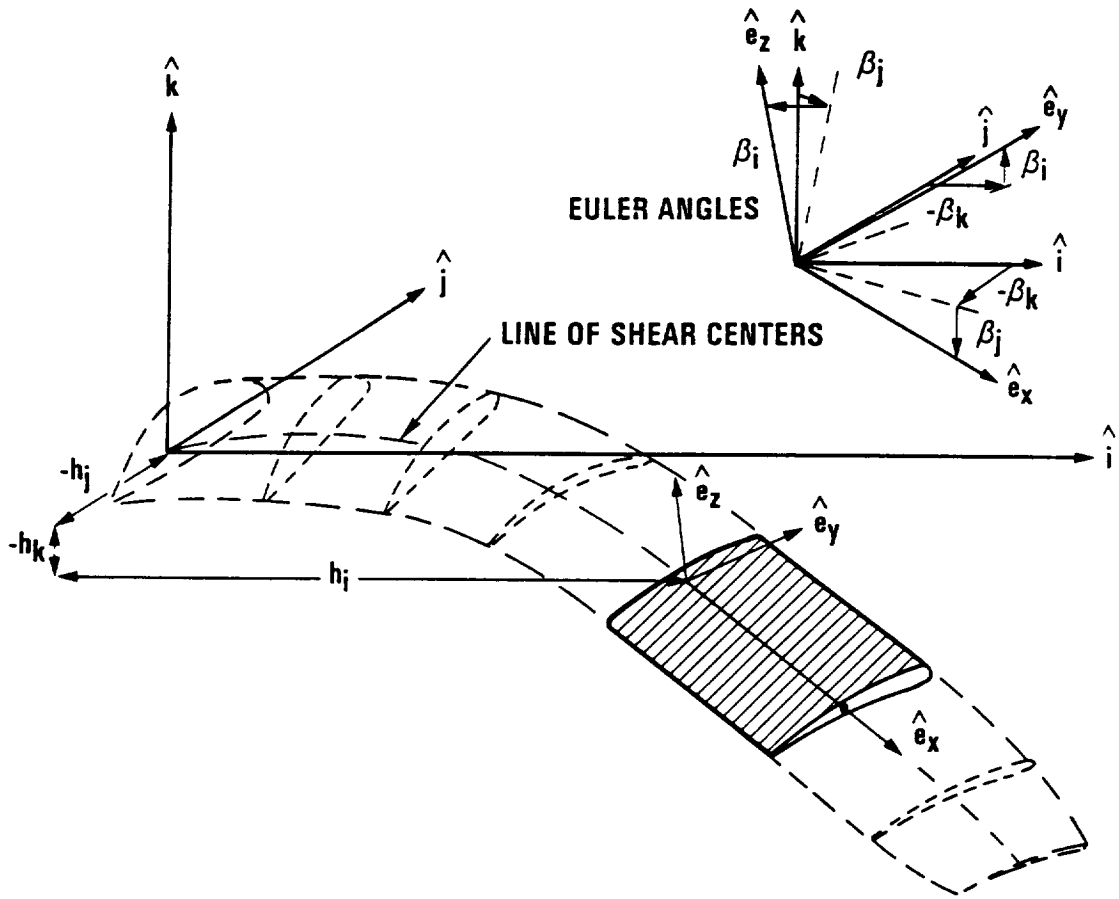


Figure 2.6-2. Propeller Fixed and Beam Element Coordinate Systems

functions) and nonlinear behavior based on the moderate deflection theory (small strains and finite rotations). All of the effects of blade rotation are properly included by allowing the beam element to be arbitrarily positioned within the blade coordinate system (Figure 2.6-2). An ordering scheme is developed to identify and neglect higher order terms which are produced during this development.

A model for the free vibration analysis of a generally curved and pretwisted rotating blade is developed by the following steps: (1) The curved blade is divided into a summation of straight beams which are laid end-to-end along the line of shear centers of the blade. (2) All of the transformations matrices associated with the blade and the straight beam element are derived. (3) The strain energy, kinetic energy, and work of external forces for a beam element are derived. (4) The finite elements associated with a single straight beam are derived by taking a variation of the total dynamic potential (Hamilton's principle). (5) The beam elements are assembled to form a structural model of the blade. (6) The free vibration analysis is calculated using the nonlinear finite element model of the blade.

#### List of Symbols

$b(U), b(T), b(W_e)$  = Strain energy, kinetic energy, and external work boundary terms of the beam, Equations (2.6-54), (2.6-67), and (2.6-71)

$E$  = Young's Modulus

$EA$  = Modulus weighted axial stiffness of the beam

$EA\eta_A, EA\zeta_A$  = Modulus weighted first moments of the beam cross section, Equations (2.6-48c) and (2.6-48e)

$\left. \begin{matrix} EAC_0, EAC_1, EAC_2 \\ EAC_3 \end{matrix} \right\}$  = Cross section integrals of the beam, Equations (2.6-48g) through (2.6-48j)

$EAD_0, EAD_1, EAD_2$   
 $EAD_3, EAD_4, EAD_5$   
 $EAD'_0, EAD'_1, EAD'_2$   
 $EAD'_3, EAD'_4$

= Modulus weighted warping constants of the beam cross section, Equations (2.6-49a) through (2.6-49k)

$EI_{\eta\eta}, EI_{SS}, EI_{\eta S}$  = Modulus weighted bending stiffness of the beam about the curvilinear coordinate axes,  $(\hat{\eta}, \hat{S})$ , Equations (2.6-48b), (2.6-48d), and (2.6-48f)

$\overline{EA}_{\eta A}, \overline{EA}_{SA},$   
 $\overline{EAD}_1, \overline{EAD}_2,$   
 $\overline{EI}_{\eta\eta}, \overline{EI}_{SS}, \overline{EI}_{\eta S}$

= Constants defined in Appendix A

$\hat{e}_x, \hat{e}_y, \hat{e}_z$  = Undeformed element coordinate axes of the beam model, Figure 2.6-2

$\hat{e}_x, \hat{e}_\eta, \hat{e}_S$  = Curvilinear coordinate axes, which rotate with the beam pretwist, before deformation, Figure 2.6-6

$\hat{e}'_x, \hat{e}'_\eta, \hat{e}'_S$  = Curvilinear coordinate axes of the beam, after deformation, Figure 2.6-8

$\hat{e}''_x, \hat{e}''_\eta, \hat{e}''_S$  = Deformed curvilinear coordinate axes of the beam,  $(\hat{e}'_x, \hat{e}'_\eta, \hat{e}'_S)$ , after a virtual motion

$\hat{E}_x, \hat{E}_\eta, \hat{E}_S$  = Base vectors on the deformed line of shear centers of the beam

$\{F^I\}, \{FCF\}, \{F^T\}$  = Element force arrays of the beam model, Equations (2.6-87) through (2.6-89)

$\{F^I\}_i, \{FCF\}_i$  = Element force subarrays of the beam model, Appendix A

$GJ$  = Torsional stiffness of the beam, Equation (2.6-48k)

$\hat{g}_x, \hat{g}_\eta, \hat{g}_S$  = Base vectors of the undeformed beam

- $\hat{G}_x, \hat{G}_\eta, \hat{G}_S$  = Base vectors of the deformed beam  
 $h_i, h_j, h_k$  = Translational offsets, used to position the element axes  $(\hat{e}_x, \hat{e}_y, \hat{e}_z)$  with the inertial reference frame  $(\hat{i}, \hat{j}, \hat{k})$ , Figure 2.6-2  
 $h_x, h_y, h_z$  = Translational offsets in the element coordinate axes  $(\hat{e}_x, \hat{e}_y, \hat{e}_z)$ , Equation (2.6-59a)  
 $\bar{h}\bar{\Omega}_x, \bar{h}\bar{\Omega}_y, \bar{h}\bar{\Omega}_z$  = Constants defined in Appendix A  
 $Im_{\eta\eta}, Im_{SS}, Im_{\eta S}$  = Mass weighted second moments of the beam cross section, Equations (2.6-65b), (2.6-65d), and (2.6-65f)  
 $\bar{I}m_{\eta\eta}, \bar{I}m_{SS}, \bar{I}m_{\eta S}$  = Constants defined in Appendix A  
 $\hat{i}, \hat{j}, \hat{k}$  = Blade coordinate system, Figure 2.6-4  
 $\hat{i}_0, \hat{j}_0, \hat{k}_0$  = Hub coordinate system, Figure 2.6-3  
 $\hat{I}, \hat{J}, \hat{K}$  = Inertial reference frame, Figure 2.6-3  
 $[J^L], [J^{NL}]$  = Linear and nonlinear Jacobian matrices, Equations (2.6-85) and (2.6-86)  
 $[J^{NL}]_{ij}$  = Submatrices of the nonlinear Jacobian matrix, Appendix A  
 $[K^L], [K^{NL}]$  } = Element stiffness matrices of the beam model,  
 $[K^{CF}], [K^I]$  } Equations (2.6-82), (2.6-87), (2.6-88), and  
 $[K^T]$  } (2.6-89)  
 $[K^L_{ij}], [K^{NL}_{ij}]$  } = Element stiffness submatrices of the beam model,  
 $[K^{CF}_{ij}], [K^I_{ij}]$  } Appendix A  
 $[K^T_{ij}]$  }

- $L$  = Length of the beam
- $m, \overline{m}, \overline{m}_m, \overline{m}_s$  = Mass weighted area and first moments of the beam cross section, Equations (2.6-65a), (2.6-65c), and (2.6-65e)
- $\overline{m}, \overline{m}_m, \overline{m}_s$  = Constants defined in Appendix A
- $\overline{M}_y, \overline{M}_z$  = Moment resultants
- $[M], [M^C]$  = Element mass and Coriolis damping matrices of the beam model, Equation (2.6-87)
- $[M_{ij}], [M_{ij}^C]$  = Element mass and Coriolis damping submatrices of the beam model, Appendix A
- $P_x, P_y, P_z$  = Beam distributed loads
- $\overline{P}_x, \overline{P}'_x$  = Stress resultants of the beam
- $q_x, q_y, q_z$  = Beam distributed moments
- $\{q\}$  = Nodal deformation array of the beam, Equation (2.6-80)
- ${}_o\hat{r}$  = Position vector of a point on the line of shear centers, before deformation
- $\hat{r}_o$  = Position vector of a point on the rotating blade after deformation, Equation (2.6-56)
- $\hat{r}_p$  = Position vector of a point on the beam, before deformation, Equation (2.6-6) and Figure 2.6-7
- $\hat{R}$  = Position vector of a point on the beam, after deformation, Equation (2.6-10) and Figure 2.6-7
- $\hat{R}_D$  = Displacement vector of any point on the beam, Equation (2.6-11) and Figure 2.6-7

$\hat{o}\hat{R}$	= Position vector of a point on the line of shear centers, after deformation
$\bar{S}_x, \bar{S}'_x$	= Moment resultants, Equations (2.6-47a) and (2.6-47b)
t	= Time
T	= Kinetic energy
$\bar{T}_x$	= Moment resultant of the beam
$\left. \begin{array}{l} [T_{HI}], [T_{BH}] \\ [T_{EB}], [T_{CE}] \\ [T_{DC}], [T_{DE}] \end{array} \right\}$	= Transformation matrices used in the beam model development, Equations (2.6-1) through (2.6-4), (2.6-25), and (2.6-59b)
u, v, w	= Displacement components
$u_c, v_c, w_c$	= Components of displacement in the $\hat{e}_x$ , $\hat{e}_\eta$ , and $\hat{e}_\zeta$ directions, respectively
U	= Strain energy
{U}, {V}, {W}	= Element deflection arrays, Equation (2.6-78)
$\bar{V}_x$	= Stress resultant of the beam
$\hat{V}$	= Velocity vector of a point on the rotating beam, Equation (2.6-56)
$W_e$	= Work of the nonconservative forces
x, y, z	= Element coordinates
x, $\eta$ , $\zeta$	= Curvilinear coordinate system of the beam, Figure 2.6-5
$x_i, y_i$	= Components of the element coordinate arrays
$\left. \begin{array}{l} Y_u, Y_v, Y_w, Y_\phi \\ Z_u, Z_v, Z_w, Z_\phi \\ Z'_v, Z'_w \end{array} \right\}$	= Notation used for writing the beam energy expressions in concise form, Equations (2.6-53a-d) and (2.6-64a-f)

## Greek Symbols

$\beta(x)$	= Pretwist angle of the beam, Equation (2.6-4)
$\beta_1, \beta_j, \beta_k$	= Euler angles, Equation (2.6-3) and Figure 2.6-2
$\beta_p, \beta_c, \beta_s$	= Pitch, precone, and sweep settings of the blade, Equation (2.6-2) and Figure 2.6-4
$\epsilon$	= Small parameter. Defined in Section 2.6.3.3
$\epsilon_{xx}, \epsilon_{\eta\eta}, \epsilon_{\zeta\zeta}$	= Normal strain components
$\epsilon_{x\zeta}, \epsilon_{\eta\zeta}, \epsilon_{x\eta}$	= Shear strain components
$\theta_x, \theta_\eta, \theta_\zeta$	= Rotation components of the curvilinear axes, $(\hat{e}_x, \hat{e}_\eta, \hat{e}_\zeta)$ , during deformation, Figure 2.6-8 and Equation (2.6-25)
$\kappa_\eta, \kappa_\zeta$	= Deformed curvatures of the beam, Equations (2.6-30a-b)
$\kappa_{\eta 0}, \kappa_{\zeta 0}$	= Initial curvatures of the beam, Equation (2.6-8)
$\Pi$	= Total potential energy of the system
$\rho$	= Mass density
$\left. \begin{array}{l} \sigma_{xx}, \sigma_{yy}, \sigma_{zz} \\ \sigma_{xx}, \sigma_{\eta\eta}, \sigma_{\zeta\zeta} \end{array} \right\}$	= Normal stress components
$\tau_0$	= Initial pretwist of the beam, Equation (2.6-8)
$\tau$	= Deformed twist of the beam, Equation (2.6-30c)
$\tau'$	= Elastic twist of the beam ( $\tau' = \tau - \tau_0$ )
$\tau_c, \tau_c'$	= Second order elastic twist effects of the beam, Equations (2.6-28) and (2.6-59b)
$\tau_{xz}, \tau_{yz}, \tau_{xy}$	= Shear stress components
$\phi$	= Elastic twist angle of the beam, Equation (2.6-27)



- $\phi_0$  = Second order elastic twist effects of the beam, Equation (2.6-31d)
- $\{\phi\}$  = Array of beam element twist rotations, Equation (2.6-78)
- $\{\phi_c(x)\}, \{\phi_q(x)\}$  = Hermite cubic and quadratic interpolation polynomials for the beam element
- $\{\phi'_c\}, \{\phi'_q\}$  = First derivative of the Hermite cubic and quadratic interpolation polynomials, Appendix A
- $\{\phi''_c\}, \{\phi''_q\}$  = Second derivative of the Hermite cubic and quadratic interpolation polynomials, Appendix A
- $\psi(\eta, \zeta)$  = Beam cross section warping function, Equation (2.6-36)
- $\omega$  = Natural frequency of vibration
- $\Omega$  = Rotational speed
- $\Omega_x, \Omega_y, \Omega_z$  = Rotational speed defined in the element coordinate  $(\hat{e}_x, \hat{e}_y, \hat{e}_z)$

### Special Symbols

- $\left. \begin{array}{l} ( )_{,x} \quad ( )_{,y} \\ ( )_{,z} \quad ( )_{,\eta} \\ ( )_{,\zeta} \end{array} \right\}$  = Derivatives of the argument with respect to the coordinates  $x, y, z, \eta,$  and  $\zeta,$  respectively
- $d( )$  = Differential of ( )
- $\delta( )$  = Variation of ( )
- $( \dot{ } )$  = Derivative with respect to time,  $t$
- $( )$  Higher order terms
- 
-

- { } = Array
- [ ] = Matrix
- [ ]<sup>-1</sup> = Inverse of a matrix

### 2.6.1 Basic Assumptions Used in the Analysis

The following assumptions are used in the development of a beam finite element model capable of a nonlinear structural dynamic analysis of rotating blades. The geometry of the blade and the beam type finite element are presented in Figures 2.6-3 through 2.6-6.

- (1) The speed of rotation of the blade is constant.
- (2) The blade is modeled by a series of straight beam finite elements which are located along the line of shear centers of the blade. The cross section for each beam finite element is defined as the section perpendicular to the line of shear centers.
- (3) The shape of the blade cross section is arbitrary (i.e. no simplifying assumptions based on symmetry are made). There are three distinct points on the cross section.
  - (a) Shear center of the cross section. The beam finite element is defined using the shear centers of two adjacent cross sections of the blade. All cross sectional properties of the beam are defined relative to this point.
  - (b) Area center of the cross section.
  - (c) Center of mass.
- (4) The beam element has an arbitrary amount of built in pretwist which is defined about the line of shear centers. Pretwist is defined as the change in the angular orientation of the principal axes of the cross section along the beam length.
- (5) The stresses within the cross section ( $\sigma_{yy}, \sigma_{zz}, \sigma_{yz}$ ) are equal to zero. The strains within the cross section ( $\epsilon_{yy}, \epsilon_{zz}, \gamma_{yz}$ ) are allowed to be nonzero, but small (negligible compared to 1).
- (6) The beam can bend in two mutually perpendicular directions normal to the line of shear centers of the beam, and can also twist about this line of shear centers.
- (7) The beam is subject to undergo moderate deflections, which implies small strains and finite rotations.
- (8) An ordering scheme, based on technical beam theory, is used to identify and neglect higher order terms.

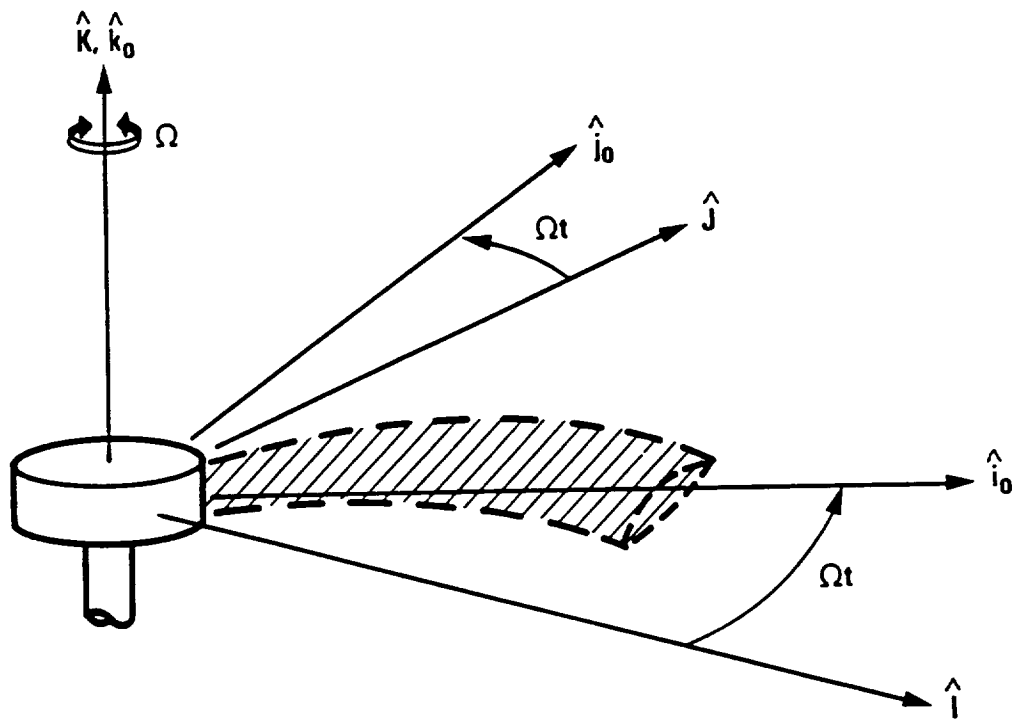


Figure 2.6-3. Inertial and Hub-Fixed Coordinate Systems

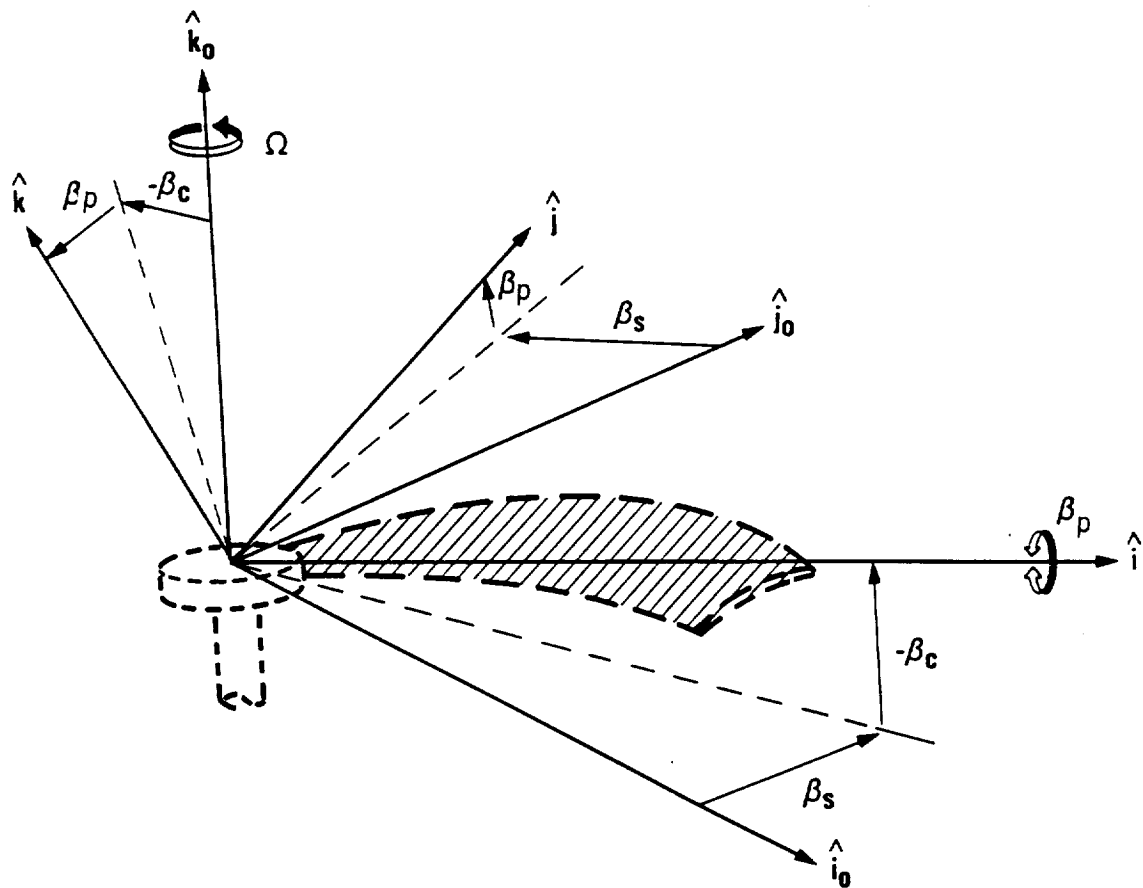


Figure 2.6-4. Hub-Fixed and Propeller-Fixed Coordinate Systems Including Sweep ( $\beta_s$ ), Precone ( $\beta_c$ ), and Pitch ( $\beta_p$ ) Adjustments

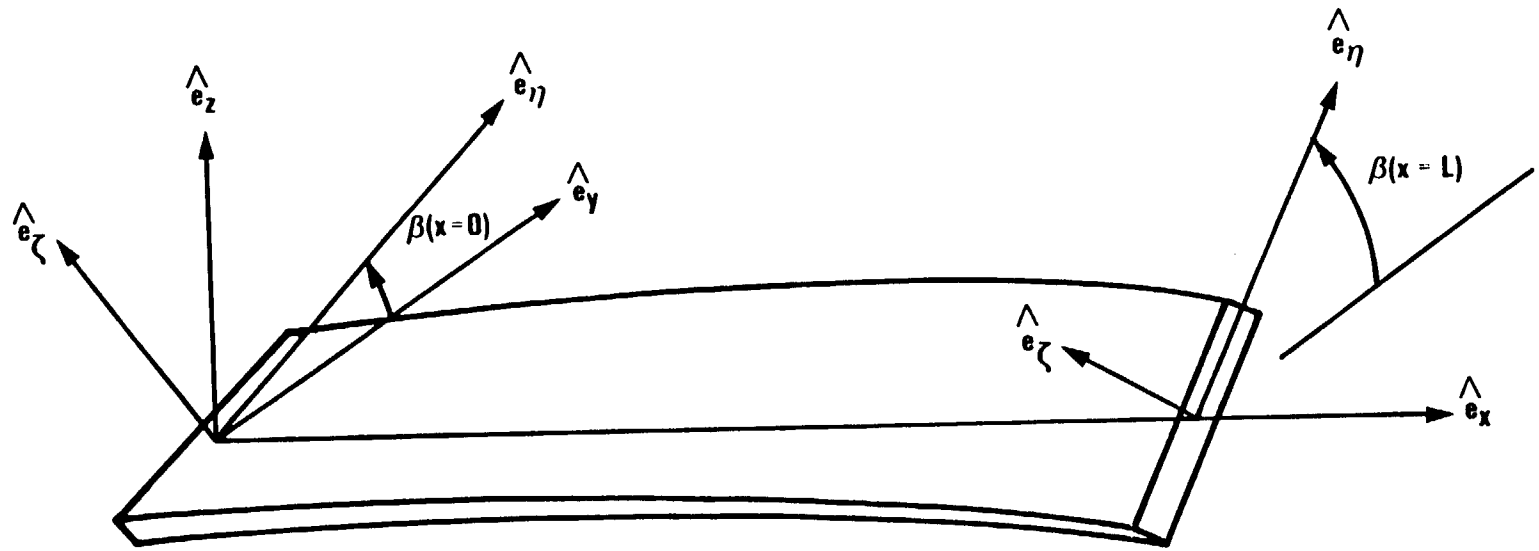
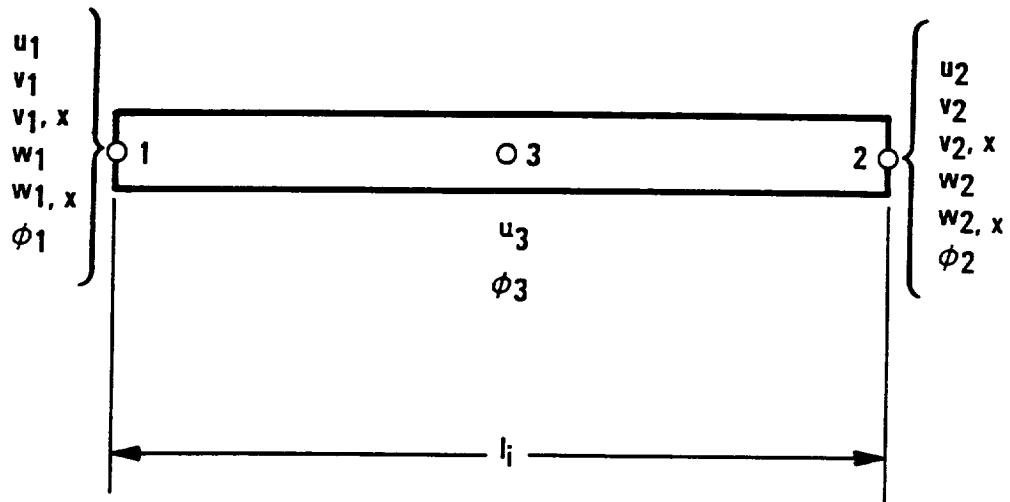
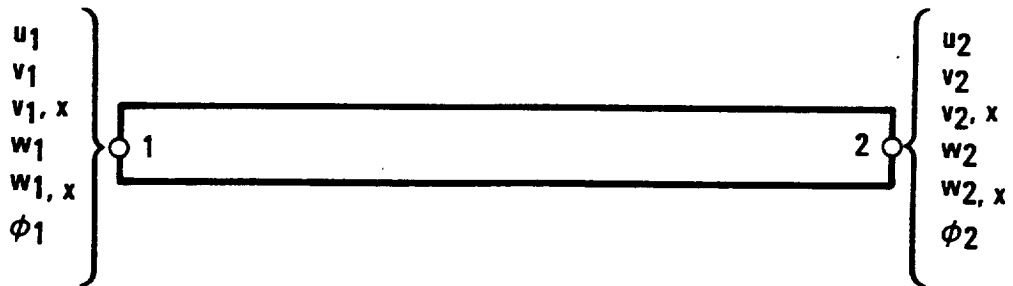


Figure 2.6-5. Element and Curvilinear Coordinate Systems of Pretwisted Beam Element



a) VIRGIN FORM—14 NODAL DEGREES OF FREEDOM



b) AFTER CONDENSATION—12 NODAL DEGREES OF FREEDOM

Figure 2.6-6. Beam-Type Finite Element

- (9) A warping function is used to model the axial deformation of the beam due to torsional twist.
- (10) The effects of damping (i.e., structural or viscous) are not included.

### 2.6.2 Geometric Preliminaries

The development of a nonlinear beam finite element model for the analysis of a rotating, pretwisted blade undergoing moderate deflections requires the use of five coordinate vector sets. The coordinate vector sets are: the element coordinate system,  $(\hat{e}_x, \hat{e}_y, \hat{e}_z)$ , the curvilinear coordinate system,  $(\hat{e}_x, \hat{e}_\eta, \hat{e}_\zeta)$ , a blade coordinate system,  $(\hat{i}, \hat{j}, \hat{k})$ , a hub coordinate system,  $(\hat{i}_0, \hat{j}_0, \hat{k}_0)$ , and an inertial reference frame  $(\hat{I}, \hat{J}, \hat{K})$ .

The beam finite element and the equations of motion are derived in the element coordinate system,  $(\hat{e}_x, \hat{e}_y, \hat{e}_z)$ . This coordinate system is an orthonormal vector set with one vector,  $\hat{e}_x$ , defined along the undeformed line of shear centers of the beam, and the other two vectors,  $\hat{e}_y$  and  $\hat{e}_z$ , defined in the cross section of the beam, Figure 2.6-5. The applied forces of the beam are defined in this system.

The curvilinear coordinate system,  $(\hat{e}_x, \hat{e}_\eta, \hat{e}_\zeta)$ , is also associated with the undeformed beam and it is used to account for the axial-bending-torsion coupling effects of a pretwisted beam. The vector,  $\hat{e}_x$ , is defined along the undeformed line of shear centers of the beam element. The two remaining vectors,  $\hat{e}_\eta$  and  $\hat{e}_\zeta$ , are defined in the beam cross section and they rotate about the line of shear centers,  $\hat{e}_x$ , with the pretwist of the beam. These two vectors remain parallel to the principal axes of the beam cross section at any point over the beam length, Figure 2.6-5. The strain components, the material properties, and the cross section warping function are all derived in this system.

The blade coordinate system  $(\hat{i}, \hat{j}, \hat{k})$  rotates with the hub and it is used to define the coordinates and the displacements of the blade. The vector  $\hat{i}$  aligns with the pitch axis of the blade and the vectors  $\hat{j}$  and  $\hat{k}$  are associated with in-plane and out-of-plane displacements of the blade, respectively. All three vectors are orthogonal to each other. The beam finite elements, which are defined in the element coordinate system, must

first be transformed to this coordinate system before they are assembled along the line of shear centers of the blade.

The hub coordinate system  $(\hat{i}_0, \hat{j}_0, \hat{k}_0)$  is fixed to the rotating hub, where the  $\hat{k}_0$  axis aligns with the spin axis,  $\hat{i}_0$  axis aligns with the pitch axis, and  $\hat{j}_0$  is perpendicular to both axes and is located in the spin plane. Changes in the pitch setting or precone of the blade are accomplished by changing the orientation of the blade coordinate system with respect to the hub coordinate system.

The inertial reference frame  $(\hat{I}, \hat{J}, \hat{K})$  is fixed in space and it is used to locate the rotating blade, where the  $\hat{K}$  axis aligns with the spin axis of the hub.

In this section, the transformations between the five coordinate vector sets are defined. The initial pretwist and curvatures of the undeformed beam are calculated and an ordering scheme is defined.

#### 2.6.2.1 Transformation Relations Between the Coordinate Systems

The four transformation matrices that relate the five different coordinate systems are defined in this section. The first transformation,  $[T_{HI}]$ , is used to relate the hub coordinate system to the inertial reference frame. The second transformation,  $[T_{BH}]$ , relates the blade coordinate system to the hub coordinate system. The third transformation,  $[T_{EB}]$ , relates the beam element coordinate system to the blade coordinate system. The fourth transformation,  $[T_{CE}]$ , relates the curvilinear coordinate system to the beam element coordinate system.

The transformation matrix,  $[T_{HI}]$ , is defined to relate the hub coordinate system to the inertial reference frame and to locate the rotating blade in space. The  $\hat{K}$  axis of the inertial reference frame and the  $\hat{k}_0$  axis of the hub are both aligned with the spin axis (Figure 2.6-3).

The transformation  $[T_{HI}]$  is written as follows:

$$\begin{Bmatrix} \hat{i}_0 \\ \hat{j}_0 \\ \hat{k}_0 \end{Bmatrix} = [T_{HI}] \begin{Bmatrix} \hat{I} \\ \hat{J} \\ \hat{K} \end{Bmatrix} \quad (2.6-1)$$



where  $[T_{HI}]$  is expressed as

$$[T_{HI}] = \begin{bmatrix} \cos(\Omega t) & \sin(\Omega t) & 0 \\ -\sin(\Omega t) & \cos(\Omega t) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where  $\Omega$  is the rotational speed of the blade and  $t$  is time.

The transformation matrix,  $[T_{BH}]$ , is defined to relate the blade coordinate system to the hub coordinate system. This transformation is used to account for changes in the orientation of the blade with respect to the hub due to pitch ( $\beta_p$ ), precone ( $\beta_c$ ), and sweep ( $\beta_s$ ) which may all be operator controlled (Figure 2.6-4). Pitch is the rotation of the blade about the pitch axis,  $i$ , precone is the rotation of the blade out of the plane of rotation, and sweep is rotation of the blade within the plane of rotation. For standard aviation propeller blades, only the pitch setting can be changed. For advanced prop-fan propellers, all three angles are included.

The transformation  $[T_{BH}]$  is written as follows:

$$\begin{bmatrix} \hat{i} \\ \hat{j} \\ \hat{k} \end{bmatrix} = [T_{BH}] \begin{bmatrix} \hat{i}_0 \\ \hat{j}_0 \\ \hat{k}_0 \end{bmatrix} \quad (2.6-2)$$

where  $[T_{BH}]$  is expressed as

$$[T_{BH}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\beta_p & \sin\beta_p \\ 0 & -\sin\beta_p & \cos\beta_p \end{bmatrix} \begin{bmatrix} \cos\beta_c & 0 & -\sin\beta_c \\ 0 & 1 & 0 \\ \sin\beta_c & 0 & \cos\beta_c \end{bmatrix} \begin{bmatrix} \cos\beta_s & \sin\beta_s & 0 \\ -\sin\beta_s & \cos\beta_s & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where the angles  $\beta_p$ ,  $\beta_c$ , and  $\beta_s$  are defined in Figure 2.6-4.

The transformation matrix,  $[T_{EB}]$ , is defined to relate the coordinate axes of the beam finite element to the blade coordinate system. Three Euler angles ( $\beta_i, \beta_j, \beta_k$ ) are used to orientate the beam finite element in the blade coordinate system (Figure 2.6-2). This transformation matrix is also used to transform the applied loads, which are defined in the blade coordinates, to the beam coordinates system.

Writing the transformation relation:

$$\begin{Bmatrix} \hat{e}_x \\ \hat{e}_y \\ \hat{e}_z \end{Bmatrix} = [T_{EB}] \begin{Bmatrix} \hat{i} \\ \hat{j} \\ \hat{k} \end{Bmatrix} \quad (2.6-3)$$

where  $[T_{EB}]$  is expressed as

$$[T_{EB}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\beta_i & \sin\beta_i \\ 0 & -\sin\beta_i & \cos\beta_i \end{bmatrix} \begin{bmatrix} \cos\beta_j & 0 & -\sin\beta_j \\ 0 & 1 & 0 \\ \sin\beta_j & 0 & \cos\beta_j \end{bmatrix} \begin{bmatrix} \cos\beta_k & \sin\beta_k & 0 \\ -\sin\beta_k & \cos\beta_k & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where the Euler angles are defined in Figure 2.6-2.

The transformation,  $[T_{CE}]$ , is used to relate the element coordinate system and the curvilinear coordinate system of the beam. This transformation relation is defined using the pretwist angle,  $\beta$ , of the beam. The pretwist angle,  $\beta$ , is defined as the angle between the element axes of the cross section,  $(\hat{e}_y, \hat{e}_z)$ , and the principal axes of the cross section at any location on the beam, Figure 2.6-5. It is used to account for variations in the cross section properties (material or shape of the cross section) over the beam length. As the shape of the cross section or the material properties vary over the beam length, the orientation of the principal axes of the cross section will also vary along the beam length.

The relation between the curvilinear coordinate system and the element coordinate system is given by:

$$\begin{Bmatrix} \hat{e}_x \\ \hat{e}_\eta \\ \hat{e}_\zeta \end{Bmatrix} = [T_{CE}] \begin{Bmatrix} \hat{e}_x \\ \hat{e}_y \\ \hat{e}_z \end{Bmatrix} \quad (2.6-4)$$

where  $[T_{CE}]$  is expressed as

$$[T_{CE}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\beta & \sin\beta \\ 0 & -\sin\beta & \cos\beta \end{bmatrix}$$

The inverse relations of Equations (2.6-1) through (2.6-4) are given by the transpose of the matrices, because all of the coordinate systems are orthonormal.

$$\begin{Bmatrix} \hat{i} \\ \hat{j} \\ \hat{k} \end{Bmatrix} = [T_{HI}]^T \begin{Bmatrix} \hat{i}_0 \\ \hat{j}_0 \\ \hat{k}_0 \end{Bmatrix} \quad (2.6-5a)$$

$$\begin{Bmatrix} \hat{i}_0 \\ \hat{j}_0 \\ \hat{k}_0 \end{Bmatrix} = [T_{BH}]^T \begin{Bmatrix} \hat{i} \\ \hat{j} \\ \hat{k} \end{Bmatrix} \quad (2.6-5b)$$

$$\begin{Bmatrix} \hat{i} \\ \hat{j} \\ \hat{k} \end{Bmatrix} = [T_{EB}]^T \begin{Bmatrix} \hat{e}_x \\ \hat{e}_y \\ \hat{e}_z \end{Bmatrix} \quad (2.6-5c)$$

$$\begin{Bmatrix} \hat{e}_x \\ \hat{e}_y \\ \hat{e}_z \end{Bmatrix} = [T_{CE}]^T \begin{Bmatrix} \hat{e}_x \\ \hat{e}_\eta \\ \hat{e}_\zeta \end{Bmatrix} \quad (2.6-5d)$$

From Figure 2.6-5, a position vector,  $\hat{r}_p$ , from the origin of the element coordinate system to a point in the cross section of the undeformed beam can be defined in either of the two coordinate systems:

$$\hat{r}_p = x \hat{e}_x + y \hat{e}_y + z \hat{e}_z = x \hat{e}_x + \eta \hat{e}_\eta + \zeta \hat{e}_\zeta \quad (2.6-6)$$

Using relation 2.6-4:

$$\begin{Bmatrix} x \\ \eta \\ \zeta \end{Bmatrix} = [T_{CE}] \begin{Bmatrix} x \\ y \\ z \end{Bmatrix} \quad (2.6-7)$$

### 2.6.2.2 Initial Curvatures and Twist of Beam

The initial curvatures ( $\kappa_{\eta 0}, \kappa_{\zeta 0}$ ) and the initial pretwist ( $\tau_0$ ) of the undeformed beam, are calculated using the transformation given in Equation (2.6-4), along with the derivatives of the curvilinear coordinate axes.

These relations are defined in the theory of space curves (for example, Wempner, Ref. 2.6-1, p. 354, Eqs. 8.8-8.10)

$$\left. \begin{aligned} \kappa_{\eta 0} &= \hat{e}_{\eta} \cdot \hat{e}_{x,x} = -\hat{e}_x \cdot \hat{e}_{\eta,x} = 0 \\ \kappa_{\zeta 0} &= \hat{e}_{\zeta} \cdot \hat{e}_{x,x} = -\hat{e}_x \cdot \hat{e}_{\zeta,x} = 0 \\ \tau_0 &= \hat{e}_{\zeta} \cdot \hat{e}_{\eta,x} = -\hat{e}_{\eta} \cdot \hat{e}_{\zeta,x} = \beta_{,x} \end{aligned} \right\} \quad (2.6-8)$$

### 2.6.2.3 Ordering Scheme for a Long Slender Beam

An ordering scheme is defined which is used to identify and delete higher order terms which are produced during the derivation of the non-linear equations of motion for a straight rotating beam. This is accomplished by assigning relative magnitudes to all terms, and then neglecting terms which are considered small in comparison to the largest terms. This approach is in agreement with the ordering scheme developed by other authors (Refs. 2.6-2 and 2.6-3).

The guidelines, which are used for assigning the relative magnitudes, were determined by careful study of the geometry and the deflection patterns of tip loaded straight cantilever beams. They are valid for most isotropic engineering materials (i.e., aluminum, steel, titanium, etc.) and most fiber-reinforced composite materials (i.e., orthotropic and anisotropic material behavior). The use of fiber-reinforced composite materials will alter the actual deflection patterns of the beam, but these patterns will not exceed the ordering scheme parameters for most engineering applications.

The geometric ratio of the cross section to the length ( $y/L$ ,  $z/L$ ) is taken as 0.2 for a long, slender beam. Defining this ratio as  $\epsilon$ , ( $\epsilon \approx 0.2$ ), quantities of  $\epsilon^2$  are neglected with respect to unity. The axial coordinate ratio of the beam is taken as unity ( $x/L \approx 1$ ). The pretwist of the beam,  $\beta$ , is defined as a large angle ( $\beta \approx 1$ ). The large angle assumption for pretwist angle,  $\beta$ , will guarantee that all the axial-bending-torsion coupling effects are included in the derivation.

The maximum allowed axial deflection of a cantilever beam is determined by applying an axial tip load that will produce a stress state equal to the material yield stress. The nondimensional axial deflection ( $u/L$ ) is on the order of 0.001-0.008 for most engineering materials.

The maximum torsional deflection that a tip-loaded cantilever beam can experience before yielding is calculated based on the octahedral shear stress theory. For a cantilever beam with an elliptical cross section, the maximum torsional deflection ( $\phi$ ) will be on the order of 0.1-0.2, ( $\epsilon$ ).

The maximum planar deflections are determined by applying a tip moment such that the stresses in the beam are equal to the material yield stress. For typical engineering materials, the nondimensional deflections ( $v/L$ ,  $w/L$ ) and slopes ( $v_{,x}$ ,  $w_{,x}$ ) at the tip will be on the order of 0.06-0.15, ( $\epsilon$ ).

The warping deflection ratio ( $\psi/L^2$ ) is defined to be the same order as the axial deflection,  $u$ . This assumption agrees with the exact solutions developed by other authors (Refs. 2.6-4 through 2.6-8), when the cross section exhibits symmetry. Based on this work, the derivatives of the warping function ( $\psi_{,\eta}/L$ ,  $\psi_{,\zeta}/L$ ) are defined as  $\epsilon$ .

A summary of the ordering parameter,  $\epsilon$ :

Order 1:

$$\left. \begin{array}{l} \frac{x}{L} \\ \beta \end{array} \right\} L ( )_{,x} \quad (2.6-9a)$$

Order  $\epsilon$ :

$$\left. \begin{array}{l} \frac{y}{L}, \frac{\eta}{L} \\ \frac{v}{L}, \frac{w}{L} \\ \phi \end{array} \right\} \begin{array}{l} \frac{z}{L}, \frac{\zeta}{L} \\ \frac{\psi_{,x}}{L}, \frac{\psi_{,y}}{L} \\ v_{,x}, w_{,x} \end{array} \quad (2.6-9b)$$

Order  $\epsilon^2$ :

$$\left. \begin{array}{l} \frac{u}{L}, u_{,x} \\ \frac{v}{L^2} \end{array} \right\} (2.6-9c)$$

The ordering scheme is applied by determining the relative magnitude of all terms of an energy expression. If the largest terms are of order ( $\epsilon^i$ ), then all of the terms of order ( $\epsilon^i$ ) are retained (first-order terms), all of the terms of order ( $\epsilon^{i+1}$ ) are retained (second-order terms), and all of the terms of order ( $\epsilon^{i+2}$ ) or higher are neglected (higher order terms). It is important to apply this ordering scheme to energy-type expressions only because, if one equation is ordered differently than another, unbalanced matrices are produced.

### 2.6.3 Development of Strain Components

A detailed development of the elastic strain components by either a linear or nonlinear approach can be found in many books which cover the theory of elasticity. In this development, the elastic strain components are derived using a full nonlinear moderate deflection theory based upon the work of Wempner (Ref. 2.6-1). They are derived in a curvilinear coordinate system ( $\hat{e}_x, \hat{e}_\eta, \hat{e}_\zeta$ ) so that the effects of beam pretwist are properly accounted for.

#### 2.6.3.1 Outline of the Calculation of the Deformations and Strains

Defining a point on an undeformed body as P, a position vector,  $\hat{r}_p$ , can be defined from the origin to the undeformed point, P [Equation (2.6-6), Figure 2.6-7].

After deformation, the point P on the body moves to a point P'. This point is defined by the vector  $\hat{R}$ , Figure 2.6-7. The vector R is a function of the initial coordinates ( $x, \eta, \zeta$ ) and the time, t.

$$\hat{R} = \hat{R}(x, \eta, \zeta, t) \quad (2.6-10)$$

The actual displacement,  $\hat{R}_D$ , of the point P due to the deformation can be defined as the difference in the vectors R (Equation (2.6-10)) and  $\hat{r}_p$  (Equation (2.6-6)), Figure 2.6-7.

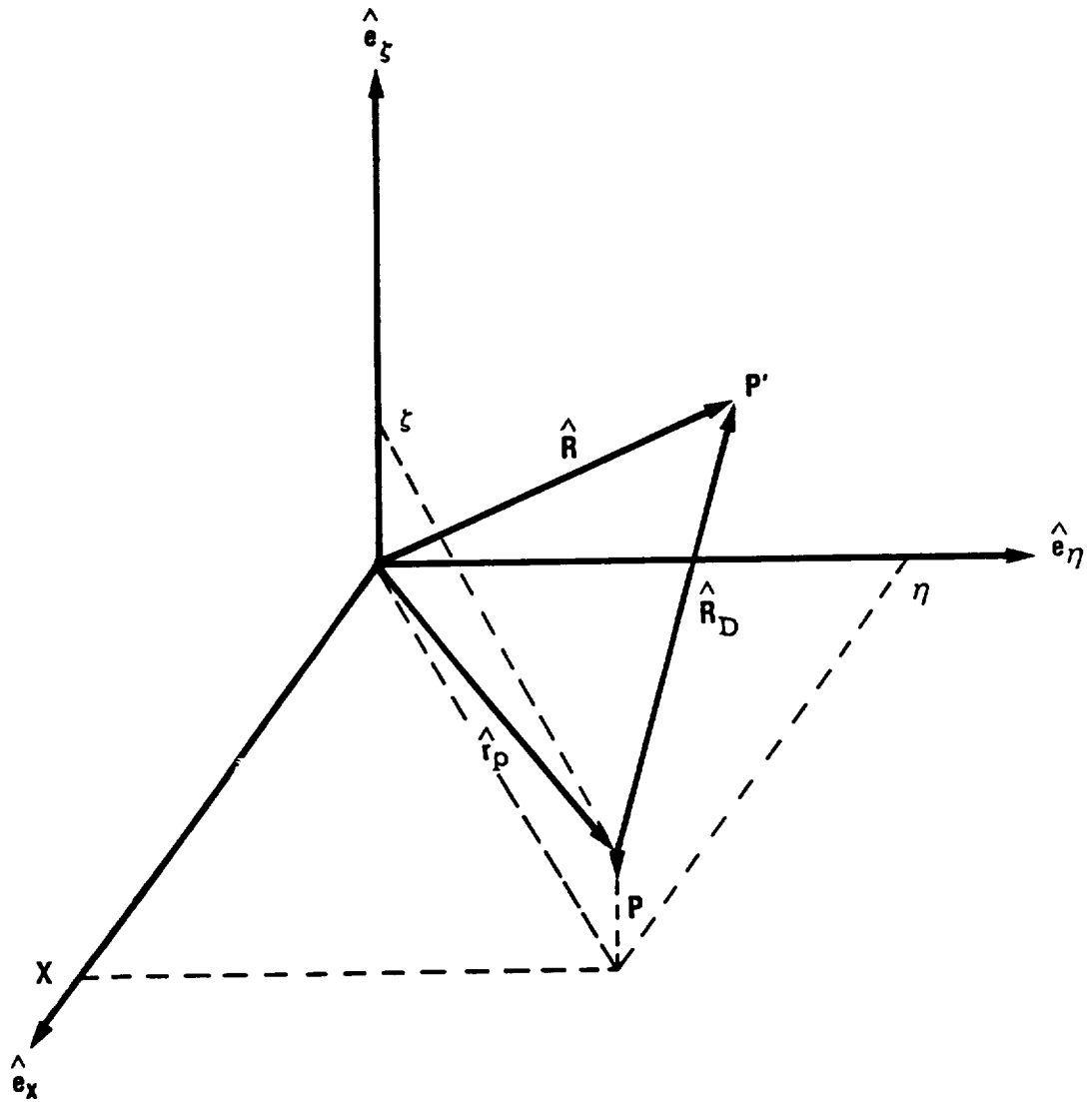


Figure 2.6-7. Position of a Material Particle Before and After the Deformation

$$\hat{R}_D = \hat{R} - r_p \quad (2.6-11)$$

The curvilinear strain components are defined in Wempner (Ref. 2.6-1, p. 355, Eq. 8.14) as follows:

$$\epsilon_{xx} = \frac{1}{2} \left\{ \frac{\hat{G}_x \cdot \hat{G}_x}{\hat{g}_x \cdot \hat{g}_x} - 1 \right\}, \quad \epsilon_{\eta\eta} = \frac{1}{2} \left\{ \hat{G}_\eta \cdot \hat{G}_\eta - 1 \right\} \quad (2.6-12a,b)$$

$$\epsilon_{x\eta} = \frac{1}{2} \left\{ \frac{\hat{G}_x \cdot \hat{G}_\eta - \hat{g}_x \cdot \hat{g}_\eta}{\hat{g}_x \cdot \hat{g}_x} \right\}, \quad \epsilon_{SS} = \frac{1}{2} \left\{ \hat{G}_S \cdot \hat{G}_S - 1 \right\} \quad (2.6-12c,d)$$

$$\epsilon_{xS} = \frac{1}{2} \left\{ \frac{\hat{G}_x \cdot \hat{G}_S - \hat{g}_x \cdot \hat{g}_S}{\hat{g}_x \cdot \hat{g}_x} \right\}, \quad \epsilon_{\eta S} = \frac{1}{2} \left\{ \hat{G}_\eta \cdot \hat{G}_S \right\} \quad (2.6-12e,f)$$

where the base vectors of a point, P, in the undeformed body are defined as:

$$\hat{g}_x = \hat{r}_{p,x}, \quad \hat{g}_\eta = \hat{r}_{p,\eta}, \quad \hat{g}_S = \hat{r}_{p,S} \quad (2.6-13)$$

and the base vectors of a point, P', in the deformed body are defined as:

$$\hat{G}_x = \hat{R}_{,x}, \quad \hat{G}_\eta = \hat{R}_{,\eta}, \quad \hat{G}_S = \hat{R}_{,S} \quad (2.6-14)$$

### 2.6.3.2 Undeformed Base Vector Development

The calculation of the undeformed base vectors ( $\hat{g}_x, \hat{g}_\eta, \hat{g}_S$ ) for any point on the undeformed beam are calculated by taking the derivatives of a position vector  $\hat{r}_p$ , with respect to the curvilinear coordinates (Equation 2.6-13). This position vector is written from the origin of the beam coordinate axes to an arbitrary point, P, on the undeformed beam (Equation 2.6-6).

Substituting the position vector into the definition of the undeformed base vectors and taking the appropriate derivatives:

$$\hat{g}_x = \hat{r}_{p,x} = \hat{e}_x, \quad \hat{g}_\eta = \hat{r}_{p,\eta} = \hat{e}_\eta, \quad \hat{g}_S = \hat{r}_{p,S} = \hat{e}_S \quad (2.6-15)$$



The position vector of a point on the line of shear centers before deformation is ( $\eta = \zeta = 0$ ):

$${}_0\hat{r} = \hat{r}_p(x,0,0,t) = x \hat{e}_x \quad (2.6-16)$$

### 2.6.3.3 Deformed Base Vector Development

The deformed base vectors ( $\hat{G}_x, \hat{G}_\eta, \hat{G}_\zeta$ ) are calculated by taking the derivatives of a position vector,  $\hat{R}$  (Figure 2.6-7), to a point on the deformed beam, P', with respect to the curvilinear coordinates (Equation 2.6-12)). The derivation of the vector  $\hat{R}$  requires the introduction of two assumptions on the deformation of the beam. The first assumption is used to define the motion of a point on the line of shear centers during deformation (Section 2.6.3.2.1). The second assumption is an approximation of the strains within the cross section (Section 2.6.3.2.2).

The position vector for an arbitrary point P' on the beam after deformation at time t, is:

$$\hat{R} = \hat{R}(x,\eta,\zeta,t) \quad (2.6-17)$$

it can also be written, using Equation (2.6-11), as:

$$\hat{R} = \hat{r}_p(x,\eta,\zeta,t) + \hat{R}_D(x,\eta,\zeta,t) \quad (2.6-18)$$

The deformed base vectors for any point on the deformed beam are calculated using the definition of the deformed base vectors from Equation (2.6-13).

$$\left. \begin{aligned} \hat{G}_x &= \hat{R}_{,x} = \hat{r}_{p,x} + \hat{R}_{D,x} = \hat{e}_x + \hat{R}_{D,x} \\ \hat{G}_\eta &= \hat{R}_{,\eta} = \hat{r}_{p,\eta} + \hat{R}_{D,\eta} = \hat{e}_\eta + \hat{R}_{D,\eta} \\ \hat{G}_\zeta &= \hat{R}_{,\zeta} = \hat{r}_{p,\zeta} + \hat{R}_{D,\zeta} = \hat{e}_\zeta + \hat{R}_{D,\zeta} \end{aligned} \right\} \quad (2.6-19)$$

The position vector for a point P on the line of shear centers, after deformation ( $\eta = \zeta = 0$ ) is:

$${}_0\hat{R} = \hat{R}(x,0,0,t) \quad (2.6-20)$$

The deformation vector ( $\hat{R}_D$ ) of a point on the line of shear centers is defined as:

$$\hat{R}_D(x,0,0,t) = u_c \hat{e}_x + v_c \hat{e}_\eta + w_c \hat{e}_\zeta \quad (2.6-21)$$

The position vector describing the line of shear centers after deformation is obtained by substituting Equations (2.6-16), (2.6-20), and (2.6-21) into Equation (2.6-18):

$$\hat{o}R = (x + u_c) \hat{e}_x + v_c \hat{e}_\eta + w_c \hat{e}_\zeta \quad (2.6-22)$$

The deformed base vectors are calculated for points which lie on along the deformed line of shear centers ( $\eta = \zeta = 0$ ), by making use of Equations (2.6-19) and (2.6-21):

$$\left. \begin{aligned} \hat{E}_x &= (1 + u_{c,x}) \hat{e}_x + (v_{c,x} - w_c \beta_{,x}) \hat{e}_\eta + (w_{c,x} + v_c \beta_{,x}) \hat{e}_\zeta \\ \hat{E}_\eta &= \hat{e}_\eta + \hat{R}_D(x_0,0,0,t)_{,\eta} \\ \hat{E}_\zeta &= \hat{e}_\zeta + \hat{R}_D(x_0,0,0,t)_{,\zeta} \end{aligned} \right\} (2.6-23)$$

The motion which carries the curvilinear lines of the undeformed beam into the curved lines of the deformed beam, carries the initial undeformed base vectors ( $\hat{g}_x, \hat{g}_\eta, \hat{g}_\zeta$ ) of a point P, to the deformed base vector set ( $\hat{G}_x, \hat{G}_\eta, \hat{G}_\zeta$ ). This motion can be looked upon as two successive motions. First, the undeformed base vector set ( $\hat{g}_x, \hat{g}_\eta, \hat{g}_\zeta$ ) is rigidly transformed and rotated to the orientation of a deformed curvilinear orthonormal vector set ( $\hat{e}_x, \hat{e}_\eta, \hat{e}_\zeta$ ). Secondly, the deformed curvilinear vector set is deformed to the deformed base vector set ( $\hat{G}_x, \hat{G}_\eta, \hat{G}_\zeta$ ), which means changing the angles of the vectors as well as the lengths. An assumption, which is based on the magnitude of the strains within the cross section ( $\epsilon_{\eta\eta}, \epsilon_{\zeta\zeta}, \epsilon_{\eta\zeta}$ ), is needed in order to complete the second step.

#### 2.6.3.3.1 Deformed Curvilinear Vector Set ( $\hat{e}_x, \hat{e}_\eta, \hat{e}_\zeta$ )

Consider an undeformed base vector set ( $\hat{g}_x, \hat{g}_\eta, \hat{g}_\zeta$ ) located on the undeformed line of shear centers of the beam before deformation. During

deformation of the line of shear centers of the beam, this vector set is carried to a deformed curvilinear vector set  $(\hat{e}_x, \hat{e}_\eta, \hat{e}_\zeta)$  through rigid body translation and rigid body rotation. Without any loss of generality, assume that  $\hat{e}_x$  is carried to the direction  $\hat{E}_x$  (i.e., tangent to the deformed line of shear centers). At this point, no assumption is made about the orientation of the remaining vectors  $(\hat{e}_\eta, \hat{e}_\zeta)$  with respect to the vectors  $(\hat{E}_\eta, \hat{E}_\zeta)$ . The transformation is illustrated in Figure 2.6-8.

$$\hat{e}_x = \hat{E}_x = (1 + u_{c,x})\hat{e}_x + (v_{c,x} - w_c \tau_0)\hat{e}_\eta + (w_{c,x} + v_c \tau_0)\hat{e}_\zeta \quad (2.6-24)$$

If the rotation of the undeformed vector set to the deformed curvilinear vector set is finite (i.e., not infinitesimal), it has to be described by Euler angles. The sequence of rotations for the transformation is  $\theta_\zeta$ ,  $\theta_\eta$ , and  $\theta_x$  and is in agreement with the work of previous authors (Refs. 2.6-2, 2.6-3, and 2.6-9). It should be noted that other sequences are possible and the form of the final set of equations is dependent on this sequence. This motion is described in Figure 2.6-8.

$$\begin{Bmatrix} \hat{e}'_x \\ \hat{e}'_\eta \\ \hat{e}'_\zeta \end{Bmatrix} = [T_{DC}] \begin{Bmatrix} \hat{e}_x \\ \hat{e}_\eta \\ \hat{e}_\zeta \end{Bmatrix} \quad (2.6-25)$$

where  $[T_{DC}]$  is expressed as

$$[T_{DC}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & \sin\theta_x \\ 0 & -\sin\theta_x & \cos\theta_x \end{bmatrix} \begin{bmatrix} \cos\theta_\eta & 0 & -\sin\theta_\eta \\ 0 & 1 & 0 \\ \sin\theta_\eta & 0 & \cos\theta_\eta \end{bmatrix} \begin{bmatrix} \cos\theta_\zeta & \sin\theta_\zeta & 0 \\ -\sin\theta_\zeta & \cos\theta_\zeta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The inverse of this transformation relation is given by the transpose of the matrix because both vector sets are orthonormal.

Consider the deformation of an element,  $dx$ , on the line of shear centers of the beam, as described in Figure 2.6-8. The deformation procedure is as follows: First the element is carried in a rigid body translational that does not appear in Figure 2.6-8. Then, the element is stretched an

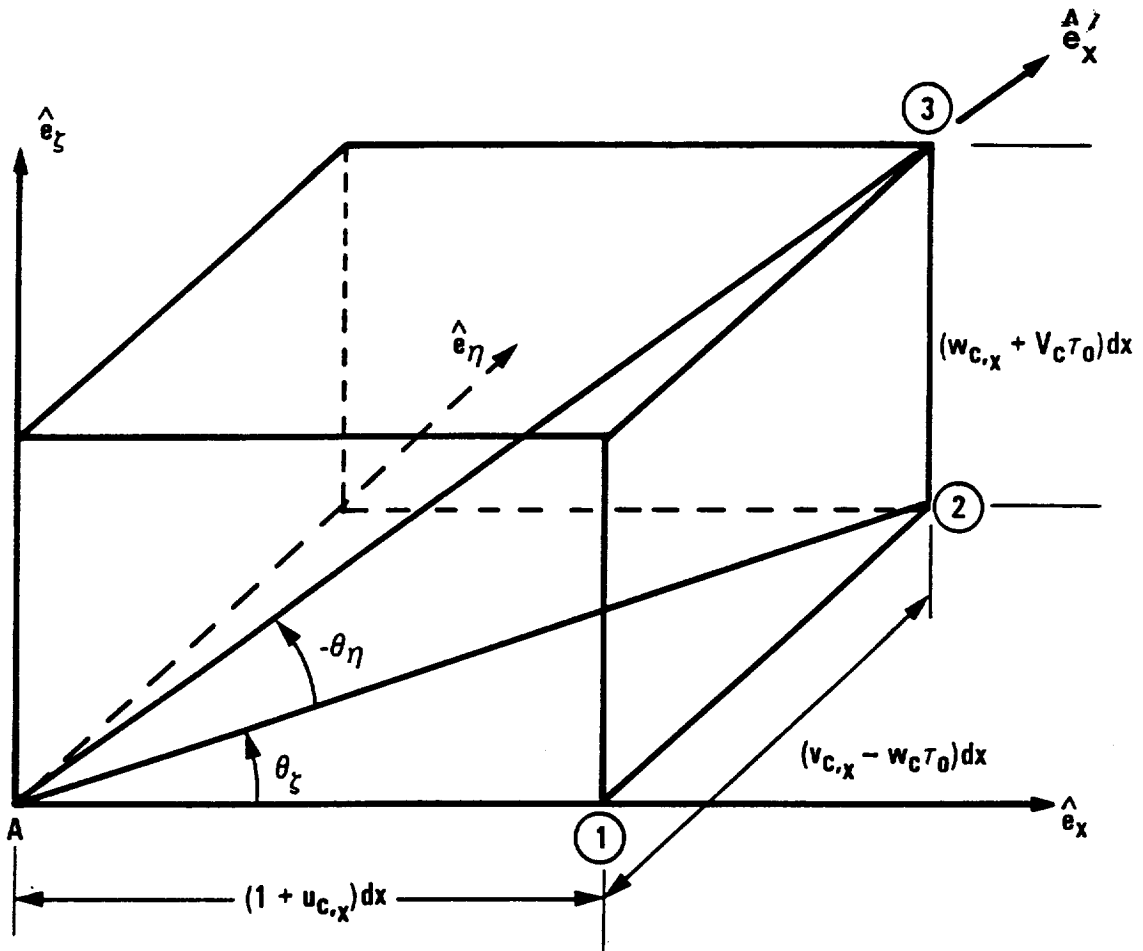


Figure 2.6-8. Curvilinear Coordinate System After Deformation and Euler Angles

amount  $u_{c,x} dx$ . Next, the element is rotated by  $\theta_S$ , about  $\hat{e}_S$ , so that the tip of the element moves a distance  $(v_{c,x} - w_c \tau_0) dx$ . Next, the element is rotated an amount  $-\theta_\eta$ , so that the tip of the element moves a distance  $(w_{c,x} + v_c \tau_0) dx$ . Finally, the element is rotated an amount  $\theta_x$  about its axial coordinate.

The following trigonometric relations can be obtained from Figure 2.6-8.

$$\cos \theta_S = \frac{1 + u_{c,x}}{\sqrt{(1 + u_{c,x})^2 + (v_{c,x} - w_c \tau_0)^2}} \quad (2.6-26a)$$

$$\sin \theta_S = \frac{v_{c,x} - w_c \tau_0}{\sqrt{(1 + u_{c,x})^2 + (v_{c,x} - w_c \tau_0)^2}} \quad (2.6-26b)$$

$$\cos \theta_\eta = \frac{\sqrt{(1 + u_{c,x})^2 + (v_{c,x} - w_c \tau_0)^2}}{\sqrt{(1 + u_{c,x})^2 + (v_{c,x} - w_c \tau_0)^2 + (w_{c,x} + v_c \tau_0)^2}} \quad (2.6-26c)$$

$$\sin \theta_\eta = \frac{-(w_{c,x} + v_c \tau_0)}{\sqrt{(1 + u_{c,x})^2 + (v_{c,x} - w_c \tau_0)^2 + (w_{c,x} + v_c \tau_0)^2}} \quad (2.6-26d)$$

The above expressions can be simplified using the ordering scheme described in Section 2.6.2.3. The torsion twist angle,  $\theta_x$ , is replaced by  $\phi$ , in order to be consistent with the literature (Ref. 2.6-4).

$$\left. \begin{aligned} \cos \theta_S &\approx 1, & \sin \theta_S &\approx v_{c,x} - w_c \tau_0 \\ \cos \theta_\eta &\approx 1, & \sin \theta_\eta &\approx -(w_{c,x} + v_c \tau_0) \\ \cos \theta_x &\approx 1, & \sin \theta_x &\approx \theta_x = \phi \end{aligned} \right\} (2.6-27)$$

These six trigonometric relations are substituted into the Euler angle transformation relation (Equation (2.6-26)) to produce the following transformation.

$$[ T_{DC} ] = \begin{bmatrix} 1 & (v_{c,x} - w_c \tau_0) & (w_{c,x} + v_c \tau_0) \\ -(v_{c,x} - w_c \tau_0) - \phi (w_{c,x} + v_c \tau_0) & 1 & \phi \\ -(w_{c,x} + v_c \tau_0) + \phi (v_{c,x} - w_c \tau_0) & -\phi - \tau_c & 1 \end{bmatrix} \quad (2.6-28)$$

where,  $\tau_c = (v_{c,x} - w_c \tau_0) (w_{c,x} + v_c \tau_0)$

The inverse of this transformation matrix is equal to the transpose, within the confines of the ordering scheme, Section 2.6.2.3.

The derivatives of the deformed curvilinear vector set  $(\hat{e}_x, \hat{e}_\eta, \hat{e}_\zeta)$  with respect to the axial coordinate of the beam,  $x$ , are calculated using the transformation relation provided in Equation (2.6-25) and Equation (2.6-28).

$$\hat{e}'_{x,x} = \left. \begin{aligned} & \left[ (v_{c,x} - w_c \tau_0)_{,x} - \tau_0 (w_{c,x} + v_c \tau_0) \right] \hat{e}_\eta \\ & + \left[ (w_{c,x} + v_c \tau_0)_{,x} + \tau_0 (v_{c,x} - w_c \tau_0) \right] \hat{e}_\zeta \end{aligned} \right\} \quad (2.6-29a)$$

$$\hat{e}'_{\eta,x} = \left. \begin{aligned} & \left[ -(v_{c,x} - w_c \tau_0) - \phi (w_{c,x} + v_c \tau_0) \right]_{,x} \hat{e}_x \\ & - \left[ \phi \tau_0 \right] \hat{e}_\eta + \left[ \phi_{,x} + \tau_0 \right] \hat{e}_\zeta \end{aligned} \right\} \quad (2.6-29b)$$

$$\hat{e}'_{\zeta,x} = \left. \begin{aligned} & \left[ -(w_{c,x} + v_c \tau_0) + \phi (v_{c,x} - w_c \tau_0) \right]_{,x} \hat{e}_x \\ & + \left[ -(\phi + \tau_c)_{,x} - \tau_0 \right] \hat{e}_\eta + \left[ -\tau_0 (\phi + \tau_c) \right] \hat{e}_\zeta \end{aligned} \right\} \quad (2.6-29c)$$

The deformed curvatures ( $\kappa_\eta, \kappa_\zeta$ ) and the deformed twist ( $\tau$ ) of the line of shear centers of the beam are defined in the theory of space curves (for example, Wempner, Ref. 2.6-1, p. 357, Eq. 8-19).

$$\kappa_\eta = \hat{e}'_\eta \cdot \hat{e}'_{x,x} = -\hat{e}'_x \cdot \hat{e}'_{\eta,x} \quad (2.6-30a)$$

$$\kappa_\zeta = \hat{e}'_\zeta \cdot \hat{e}'_{x,x} = -\hat{e}'_x \cdot \hat{e}'_{\zeta,x} \quad (2.6-30b)$$

$$\tau = \hat{e}'_\zeta \cdot \hat{e}'_{\eta,x} = -\hat{e}'_\eta \cdot \hat{e}'_{\zeta,x} \quad (2.6-30c)$$

The deformed curvatures and the deformed twist of the line of shear centers are calculated by taking the dot product of the deformed curvilinear vector set (Equations 2.6-25 and 2.6-28) with the derivatives of the deformed curvilinear vector set (Equations 2.6-29a-c). Using the ordering scheme, neglecting higher order terms, and substituting in for the element deformations ( $u, v, w$ ):

$$\kappa_\eta = v_{,xx} \cos(\beta + \phi) + w_{,xx} \sin(\beta + \phi) \quad (2.6-31a)$$

$$\kappa_\zeta = -v_{,xx} \sin(\beta + \phi) + w_{,xx} \cos(\beta + \phi) \quad (2.6-31b)$$

$$\tau = \tau_0 + \phi_{,x} + \phi_0 \quad (2.6-31c)$$

where

$$\phi_0 = \left\{ w_{,x} \cos(\beta + \phi) - v_{,x} \sin(\beta + \phi) \right\} \left\{ v_{,xx} \cos(\beta + \phi) + w_{,xx} \sin(\beta + \phi) \right\} \quad (2.6-31d)$$

and the small angle assumption is used for  $\phi$ :

$$\left. \begin{aligned} \cos(\beta + \phi) &\simeq \cos\beta - \phi \sin\beta \\ \sin(\beta + \phi) &\simeq \sin\beta + \phi \cos\beta \end{aligned} \right\} \quad (2.6-31e)$$

From Equations (2.6-31a) through (2.6-31c) and the orthonormality conditions for  $(\hat{e}_x, \hat{e}_\eta, \hat{e}_\zeta)$ , it can be shown that:

$$\left. \begin{aligned} \hat{e}'_{x,x} &= \kappa_\eta \hat{e}'_\eta + \kappa_\zeta \hat{e}'_\zeta \\ \hat{e}'_{\eta,x} &= -\kappa_\eta \hat{e}'_x + \tau \hat{e}'_\zeta \\ \hat{e}'_{\zeta,x} &= -\kappa_\zeta \hat{e}'_x - \tau \hat{e}'_\eta \end{aligned} \right\} \quad (2.6-32)$$

Up to this point, the only assumption that has been made about the orientation of the deformed curvilinear vector set  $(\hat{e}_x, \hat{e}_\eta, \hat{e}_\zeta)$  with respect to the base vectors for the deformed line of shear centers  $(\hat{E}_x, \hat{E}_\eta, \hat{E}_\zeta)$ , is that  $\hat{e}_x$  is parallel to  $\hat{E}_x$  (Equation 2.6-23). This assumption has also been made in previous developments (Ref. 2.6-2 and 2.6-3). The axes  $\hat{e}_\eta$  and  $\hat{e}_\zeta$  will have a slightly different orientation than  $\hat{E}_\eta$  and  $\hat{E}_\zeta$  due to the presence of strain within the cross section. The vectors,  $\hat{e}_\eta$  and  $\hat{e}_\zeta$ , can be expressed in terms of  $\hat{E}_\eta$  and  $\hat{E}_\zeta$  using the definitions presented in Wempner (Ref. 2.6-1, p. 356, Equation 8.16):

$$\hat{e}'_\eta \cdot \hat{E}_\eta = 1 + \epsilon_{\eta\eta} \quad (2.6-33a)$$

$$\hat{e}'_\zeta \cdot \hat{E}_\zeta = 1 + \epsilon_{\zeta\zeta} \quad (2.6-33b)$$

$$\hat{e}'_\eta \cdot \hat{E}_\zeta = \hat{e}'_\zeta \cdot \hat{E}_\eta = 1 + \epsilon_{\eta\zeta} \quad (2.6-33c)$$

For beams, composed of isotropic materials, the Bernoulli-Euler hypothesis is used (Ref. 2.6-3, 2.6-7, and 2.6-8) to relate the deformed curvilinear vector set  $(\hat{e}_x, \hat{e}_\eta, \hat{e}_\zeta)$  to the base vectors of the deformed line of shear centers  $(\hat{E}_x, \hat{E}_\eta, \hat{E}_\zeta)$ . This hypothesis states:



"During bending, plane cross sections which are normal to the elastic axis (line of shear centers) before deformation remain plane after deformation, and normal to the deformed elastic axis (line of shear centers)."

This hypothesis is combined with the implicitly stated assumption that strains within the cross section are equal to zero.

$$\epsilon_{\eta\eta} = \epsilon_{\zeta\zeta} = \epsilon_{\eta\zeta} = 0 \quad (2.6-34)$$

Combining Equation (2.6-34) with Equations (2.6-33a-c) will yield that the deformed curvilinear vector set  $(\hat{e}_x, \hat{e}_\eta, \hat{e}_\zeta)$  which is parallel to the base vectors of the deformed line of shear centers  $(\hat{E}_x, \hat{E}_\eta, \hat{E}_\zeta)$ :

$$\hat{e}'_x = \hat{E}_x, \quad \hat{e}'_\eta = \hat{E}_\eta, \quad \hat{e}'_\zeta = \hat{E}_\zeta \quad (2.6-35)$$

#### 2.6.3.3.2 Deformation Approximation

The approximation for the deformation of a beam can be written by combining the solutions for simple extension, flexure, and torsion. A modification is required to accommodate the warping of the cross section which is produced from the bending and/or torsion of the composite beam. The vector  $\hat{R}$ , which is a position vector of a point in the cross section of the beam after deformation, consists of the sum of the deformation of a point on the line of shear centers combined with the warping of the cross section due to torsion twist of the beam and the rigid rotation of the cross section.

$$\hat{R} = {}_o\hat{R} + \psi \tau' \hat{E}_x + \eta \hat{E}_\eta + \zeta \hat{E}_\zeta \quad (2.6-36)$$

The axial deformation due to elastic torsional twist of the beam is analyzed using the product of a warping type function,  $\psi$ , and the elastic twist,  $\tau'$ , of the beam at the cross section location. This approach was first used for the torsional analysis of straight, untwisted, isotropic rods. It has been shown that the warping function will have antisymmetric properties for doubly symmetric cross sections (Refs. 2.6-1, 2.6-5, 2.6-6 and 2.6-10). This same approach is also used for the analysis of the axial

deformation of straight, pretwisted, isotropic rods (Refs. 2.6-11 through 2.6-18), where the assumption is made that the warping function is an antisymmetric type function that is only dependent on the cross section variables and not on the pretwist.

In this study, a general warping function is used to model the axial deformation due to elastic torsional twist. It is assumed that the function is dependent on all three coordinates  $(x, \eta, \zeta)$  for a general pretwisted beam, but will only be dependent on the cross section coordinates  $(\eta, \zeta)$  for a uniform pretwisted beam (i.e., a beam that has a constant cross section along the beam length, but the pretwist varies along the length). No assumption regarding the form of the function is made (for example, antisymmetric function). The order of magnitude of the axial deformation due to elastic twist, is assumed to be no larger than the allowed axial deformation due to axial loads (from Section 2.6.2.3,  $o(\epsilon^2)$ ).

Substituting Equations (2.6-22) and (2.6-35) into Equations (2.6-36), yields:

$$\hat{R} = \{x + u_c, v_c, w_c\} \begin{Bmatrix} \hat{e}_x \\ \hat{e}_\eta \\ \hat{e}_\zeta \end{Bmatrix} + \{\psi, \tau', \eta, \zeta\} \begin{Bmatrix} \hat{e}'_x \\ \hat{e}'_\eta \\ \hat{e}'_\zeta \end{Bmatrix} \quad (2.6-37)$$

Substituting Equation (2.6-24) into Equation (2.6-36):

$$\hat{R} = \{x + u_c, v_c, w_c\} + \{\psi, \tau', \eta, \zeta\} [T_{DC}] \begin{Bmatrix} \hat{e}_x \\ \hat{e}_\eta \\ \hat{e}_\zeta \end{Bmatrix} \quad (2.6-38)$$

The deformed base vectors  $(\hat{G}_x, \hat{G}_\eta, \hat{G}_\zeta)$  are calculated using Equations (2.6-14), (2.6-32) and (2.6-37):

$$\left. \begin{aligned} \hat{G}_x = \hat{R}_{,x} &= (1 + u_{c,x}) \hat{e}_x + \{(\psi \tau')_{,x} - \eta \kappa_\eta - \zeta \kappa_\zeta\} \hat{e}'_x \\ &+ (v_{c,x} - w_c \tau_0) \hat{e}_\eta + \{(\psi \tau') \kappa_\eta - \zeta \tau'\} \hat{e}'_\eta \\ &+ (w_{c,x} + v_c \tau_0) \hat{e}_\zeta + \{(\psi \tau') \kappa_\zeta + \eta \tau'\} \hat{e}'_\zeta \end{aligned} \right\} \quad (2.6-39a)$$

$$\hat{G}_\eta = \hat{R}_{,\eta} = (\psi_{,\eta} \tau') \hat{e}'_x + \hat{e}'_\eta \quad (2.6-39b)$$

$$\hat{G}_\zeta = \hat{R}_{,\zeta} = (\psi_{,\zeta} \tau') \hat{e}'_x + \hat{e}'_\zeta \quad (2.6-39c)$$

#### 2.6.3.4 Strain Components

The strain components (Equations 2.6-12a-f) are calculated using the undeformed base vectors ( $\hat{g}_x, \hat{g}_\eta, \hat{g}_\zeta$ ) from Equation (2.6-15), along with deformed base vectors ( $\hat{G}_x, \hat{G}_\eta, \hat{G}_\zeta$ ) from Equations (2.6-39a-c). They are expressed in the curvilinear coordinate system ( $\hat{e}_x, \hat{e}_\eta, \hat{e}_\zeta$ ) in terms of the element displacements ( $u, v, w$ ). The ordering scheme is used to neglect higher order terms which are generated during the formation of the strain components.

$$\left. \begin{aligned} \epsilon_{xx} &= u_{,x} + (\psi \phi_{,x})_{,x} + \frac{1}{2} (\eta^2 + \zeta^2) \frac{(\phi_{,x})^2}{\phantom{(\phi_{,x})^2}} \\ &+ \frac{1}{2} (v_{,x})^2 - v_{,xx} \{\eta \cos(\beta + \phi) - \zeta \sin(\beta + \phi)\} \\ &+ \frac{1}{2} (w_{,x})^2 - w_{,xx} \{\zeta \cos(\beta + \phi) + \eta \sin(\beta + \phi)\} \end{aligned} \right\} \quad (2.6-40a)$$

$$\epsilon_{x\eta} = \frac{1}{2} (\psi_{,\eta} - \zeta) (\phi_{,x} + \phi_0) \quad (2.6-40b)$$

$$\epsilon_{x\zeta} = \frac{1}{2} (\psi_{,\zeta} + \eta) (\phi_{,x} + \phi_0) \quad (2.6-40c)$$

$$\epsilon_{\eta\eta} \approx \epsilon_{\zeta\zeta} \approx \epsilon_{\eta\zeta} \approx 0 \quad (2.6-40d-f)$$

where  $\phi_0$  represents a second-order effect that couples elastic torsion twist with the beam bending (Equation 2.6-31d).

These six strain components can be directly compared with the strain components derived by Hodges and Dowell (Ref. 2.6-2), in their development of the nonlinear equations of a twisted rotor blade. The strain components agree with those calculated by Hodges and Dowell except for three terms.

The first term,  $\phi_0$ , is the second order effects in the shear strains,  $\epsilon_{x\eta}$  and  $\epsilon_{x\zeta}$  that appears in this model. This term, Equation (2.6-31d), represents the bending-torsion coupling of the pretwisted beam.

A second term,  $(\eta^2 + \zeta^2) \phi_{,x} \beta_{,x}$ , representing additional axial strain due to beam pretwist is not present in this derivation and was shown to be incorrect by other authors (Ref. 2.6-3, 2.6-11 through 2.6-17, and 2.6-19). This term, which is used to model the axial-torsion coupling effects due to pretwist, is inappropriate because it does not reduce to zero for the simple case of a uniform beam with a circular cross section.

A third term is the axial strain due to cross section warping. In this development the term has the form,  $(\psi \phi_{,x})_{,x}$ , while in development of Hodges and Dowell the term has the form,  $\psi \phi_{,xx}$ . For pretwisted beams with a uniform cross section, the warping function is only a function of the cross section coordinates,  $\eta$  and  $\zeta$ , so the chain rule must be applied to calculate the derivative.

$$\begin{aligned} (\psi \phi_{,x})_{,x} &= \psi \phi_{,xx} + \{ \psi_{,x} \} \phi_{,x} \\ &= \psi \phi_{,xx} + \{ \psi_{,\eta} \eta_{,x} + \psi_{,\zeta} \zeta_{,x} \} \phi_{,x} \\ &= \psi \phi_{,xx} + \{ \psi_{,\eta} \zeta - \psi_{,\zeta} \eta \} \tau_0 \phi_{,x} \end{aligned} \quad (2.6-41)$$

This term has been shown to represent the correct model for the axial deformation and the decrease in torsional stiffness of a pretwisted uniform beam subjected to axial loads (Refs. 2.6-11 through 2.6-13, 2.6-16, and

2.6-18). Furthermore, it will reduce to zero for a uniform beam with a circular cross section.

#### 2.6.4 Derivation of the Equations of Motion by Hamilton's Principle

The equations of motion and boundary conditions are derived for a beam rotating about an arbitrary axis using Hamilton's principle. These equations are valid for a slender beam which is composed of a homogenous material. The beam is defined so that it can be arbitrarily offset and oriented with respect to the spin axis. This representation includes lengthwise variation in the following properties: beam pretwist, mass and stiffness properties, and mass centroid offset and area centroid offset from the line of shear centers. The external loads are represented by a set of generalized distributed loads and moments, which are defined in the element coordinate axes ( $\hat{e}_x, \hat{e}_y, \hat{e}_z$ ).

The strain energy,  $U$ , is obtained using the strain components and the constitutive relations defined in the curvilinear coordinate system ( $\hat{e}_x, \hat{e}_\eta, \hat{e}_\zeta$ ). The curvilinear coordinate system is chosen because, for most applications where the beam has a uniform cross section with varying pretwist, the cross section properties need only be defined once. If the element coordinate system was chosen, the cross section properties would need to be specified at each node location.

The kinetic energy,  $T$ , is developed by calculating the velocity vector of a point on the deformed beam and using the mass distribution of the cross section. The velocity vector is calculated by taking the time derivative of a position vector from the origin of the inertial reference to a point in the cross section on the deformed rotating beam. This position vector is capable of modeling a beam with an arbitrary offset and orientation from the inertial reference frame.

The work of external forces,  $W_e$ , is derived by including the effects of the nonconservative forces (i.e., distributed loads and moments) acting through a displacement about the deformed equilibrium position.

Hamilton's principle represents a generalization of the principle of minimum potential energy to include dynamic effects. It is stated as an integral equation in which the total dynamic potential ( $U - T - W_e$ ) is an

extremum over the time integral;  $t_1 \leq t \leq t_2$ . A discussion of the principle can be found in Refs. 2.6-20 and 2.6-21. Hamilton's principle can be stated in the following mathematical form:

$$\int_{t_1}^{t_2} (\delta U - \delta T - \delta W_e) dt = 0 \quad (2.6-42)$$

#### 2.6.4.1 Strain Energy Contributions

The total strain energy over the entire volume of a structure is written in terms of the curvilinear stresses and strains.

$$U = \frac{1}{2} \int_0^L \iint_A (\sigma_{xx} \epsilon_{xx} + \sigma_{x\zeta} \gamma_{x\zeta} + \sigma_{x\eta} \gamma_{x\eta} + \sigma_{\eta\eta} \epsilon_{\eta\eta} + \sigma_{\zeta\zeta} \epsilon_{\zeta\zeta} + \sigma_{\eta\zeta} \gamma_{\eta\zeta}) d\eta d\zeta dx \quad (2.6-43)$$

where  $\gamma_{x\eta} = 2 \epsilon_{x\eta}$ ,  $\gamma_{x\zeta} = 2 \epsilon_{x\zeta}$ ,  $\gamma_{\eta\zeta} = 2 \epsilon_{\eta\zeta}$

The variation of the strain energy,  $\delta U$ , is calculated using the assumption that the stresses within the cross section are equal to zero ( $\sigma_{\eta\eta} = \sigma_{\zeta\zeta} = \sigma_{\eta\zeta} = 0$ ) and by substituting the elastic isotropic stress-strain relations into Equation (2.6-43):

$$\delta U = \int_0^L \iint_A \begin{Bmatrix} \delta \epsilon_{xx} \\ \delta \gamma_{x\zeta} \\ \delta \gamma_{x\eta} \end{Bmatrix}^T \begin{bmatrix} E & 0 & 0 \\ 0 & G & 0 \\ 0 & 0 & G \end{bmatrix} \begin{Bmatrix} \epsilon_{xx} \\ \gamma_{x\zeta} \\ \gamma_{x\eta} \end{Bmatrix} d\zeta d\eta dx \quad (2.6-44)$$

where the strains have been defined in Equations (2.6-32a-f) and the variations of the strain components can be written as:

$$\begin{aligned}
\delta\epsilon_{xx} = & \delta u_{,x} + v_{,x} \delta v_{,x} + w_{,x} \delta w_{,x} + \delta(\psi \phi_{,x})_{,x} \\
& - (\delta v_{,xx} + w_{,xx} \delta\phi) \{ \eta \cos(\beta + \phi) - \zeta \sin(\beta + \phi) \} \\
& - (\delta w_{,xx} - v_{,xx} \delta\phi) \{ \zeta \cos(\beta + \phi) + \eta \sin(\beta + \phi) \} \\
& + \frac{(\eta^2 + \zeta^2) \phi_{,x} \delta\phi_{,x}}{\phantom{}}
\end{aligned}
\quad \left. \vphantom{\begin{aligned} \delta\epsilon_{xx} = \\ - (\delta v_{,xx} + w_{,xx} \delta\phi) \\ - (\delta w_{,xx} - v_{,xx} \delta\phi) \\ + \frac{(\eta^2 + \zeta^2) \phi_{,x} \delta\phi_{,x}}{\phantom{}} \end{aligned}} \right\} (2.6-45a)$$

$$\delta\gamma_{x\eta} = (\psi_{,\eta} - \zeta)(\delta\phi_{,x} + \delta\phi_0) \quad (2.6-45b)$$

$$\delta\gamma_{x\zeta} = (\psi_{,\zeta} + \eta)(\delta\phi_{,x} + \delta\phi_0) \quad (2.6-45c)$$

where

$$\begin{aligned}
\delta\phi_0 = & \{ \delta w_{,x} \cos(\beta + \phi) - \delta v_{,x} \sin(\beta + \phi) \} \{ v_{,xx} \cos(\beta + \phi) + w_{,xx} \sin(\beta + \phi) \} \\
& + \{ w_{,x} \cos(\beta + \phi) - v_{,x} \sin(\beta + \phi) \} \{ \delta v_{,xx} \cos(\beta + \phi) + \delta w_{,xx} \sin(\beta + \phi) \}
\end{aligned}
\quad (2.6-45d)$$

Since, the strain components are of order  $\epsilon^2$  and  $\epsilon^3$  based on an ordering scheme, the variation in the strain components are also defined as order  $\epsilon^2$  and  $\epsilon^3$ . Expanding the strain energy (Equation 2.6-44) will yield terms of order  $\epsilon^4$ ,  $\epsilon^5$ , and  $\epsilon^6$ . Consistent with the assumptions of the ordering scheme, terms of order  $\epsilon^6$  are neglected compared to terms of order  $\epsilon^4$ . Once the triple product is carried out, the integration is performed over the cross section.

The variation of the strain energy,  $\delta U$ , is written in terms of stress and moment resultants:

$$\begin{aligned}
\delta U = \int_0^L \bar{V}_x (\delta u_{,x} + v_{,x} \delta v_{,x} + w_{,x} \delta w_{,x}) + \bar{P}_x \delta \phi_{,xx} \\
+ (\bar{M}_z \cos(\beta + \phi) + \bar{M}_y \sin(\beta + \phi)) (\delta v_{,xx} + w_{,xx} \delta \phi) \\
+ (\bar{M}_z \sin(\beta + \phi) - \bar{M}_y \cos(\beta + \phi)) (\delta w_{,xx} - v_{,xx} \delta \phi) \\
+ (\bar{P}'_x + \bar{S}'_x + \underline{\underline{\bar{T}'_x \phi_{,x}}}) \delta \phi_{,x} + \bar{S}'_x \delta \phi_0 \quad dx
\end{aligned} \quad (2.6-46)$$

The definition of the stress resultants is described below.

Terms which are double or triple underlined should be dropped when using the ordering scheme in a strict manner, (order  $\epsilon^6$ ,  $\epsilon^7$ ), however, these terms are retained because they represent important axial-torsion coupling effects for pretwisted beams (Ref. 2.6-11 and 2.6-13).

$$\bar{V}_x = \iint_A E \epsilon_{xx} d\eta d\zeta \quad (2.6-47a)$$

$$\begin{aligned}
&= EA \left\{ u_{,x} + \frac{1}{2} (v_{,x})^2 + \frac{1}{2} (w_{,x})^2 \right\} + EAD_0 \phi_{,xx} + EAD'_0 \phi_{,x} \\
&\quad - EA\eta_A (v_{,xx} \cos(\beta + \phi) + w_{,xx} \sin(\beta + \phi)) \\
&\quad - EA\zeta_A (w_{,xx} \cos(\beta + \phi) - v_{,xx} \sin(\beta + \phi)) + \underline{\underline{\frac{1}{2} EAC_0 (\phi_{,x})^2}}
\end{aligned}$$

$$\bar{P}_x = \iint_A \psi E \epsilon_{xx} d\eta d\zeta \quad (2.6-47b)$$

$$\begin{aligned}
&= EAD_0 \left\{ u_{,x} + \frac{1}{2} (v_{,x})^2 + \frac{1}{2} (w_{,x})^2 \right\} + \underline{\underline{EAD_3 \phi_{,xx}}} \\
&\quad - EAD_1 (v_{,xx} \cos(\beta + \phi) + w_{,xx} \sin(\beta + \phi)) + \underline{\underline{EAD_5 \phi_{,x}}}
\end{aligned}$$



$$- \underline{\underline{EAD_2}} (w_{,xx} \cos(\beta + \phi) - v_{,xx} \sin(\beta + \phi)) + \underline{\underline{\frac{1}{2} EAD_4 (\phi_{,x})^2}}$$

$$\bar{P}'_x = \iint_A \phi_{,x} E \epsilon_{xx} d\eta d\zeta \quad (2.6-47c)$$

$$= EAD'_0 \left\{ u_{,x} + \frac{1}{2} (v_{,x})^2 + \frac{1}{2} (w_{,x})^2 \right\} + \underline{\underline{EAD_5 \phi_{,xx}}}$$

$$- \underline{\underline{EAD'_1}} (v_{,xx} \cos(\beta + \phi) + w_{,xx} \sin(\beta + \phi)) + \underline{\underline{EAD'_3 \phi_{,x}}}$$

$$- \underline{\underline{EAD'_2}} (w_{,xx} \cos(\beta + \phi) - v_{,xx} \sin(\beta + \phi)) + \underline{\underline{\frac{1}{2} EAD'_4 (\phi_{,x})^2}}$$

$$\underline{\underline{\bar{T}'_x}} = \iint_A (\eta^2 + \zeta^2) E \epsilon_{xx} d\eta d\zeta \quad (2.6-47d)$$

$$= \underline{\underline{EAC_0}} \left\{ u_{,x} + \frac{1}{2} (v_{,x})^2 + \frac{1}{2} (w_{,x})^2 \right\} + \underline{\underline{EAD_4 \phi_{,xx}}}$$

$$- \underline{\underline{EAC_1}} (v_{,xx} \cos(\beta + \phi) + w_{,xx} \sin(\beta + \phi)) + \underline{\underline{EAD'_4 \phi_{,x}}}$$

$$- \underline{\underline{EAC_2}} (w_{,xx} \cos(\beta + \phi) - v_{,xx} \sin(\beta + \phi)) + \underline{\underline{\frac{1}{2} EAC_3 (\phi_{,x})^2}}$$

$$\bar{M}'_y = \iint_A \zeta E \epsilon_{xx} d\eta d\zeta \quad (2.6-47e)$$

$$= EAS_A \left\{ u_{,x} + \frac{1}{2} (v_{,x})^2 + \frac{1}{2} (w_{,x})^2 \right\} + EAD_2 \phi_{,xx} + EAD'_2 \phi_{,x}$$

$$- EI_{\eta\zeta} (v_{,xx} \cos(\beta + \phi) + w_{,xx} \sin(\beta + \phi))$$

$$- EI_{\eta\eta} (w_{,xx} \cos(\beta + \phi) - v_{,xx} \sin(\beta + \phi)) + \frac{1}{2} \underline{\underline{EAC_2}} (\phi_{,x})^2$$

$$\bar{M}_z = \iint_A -\eta E \epsilon_{xx} d\eta d\zeta \quad (2.6-47f)$$

$$\begin{aligned} &= -EA\eta_A \left\{ u_{,x} + \frac{1}{2} (v_{,x})^2 + \frac{1}{2} (w_{,x})^2 \right\} - EAD_1 \phi_{,xx} - EAD_1' \phi_{,x} \\ &+ EI_{\zeta\zeta} (v_{,xx} \cos(\beta + \phi) + w_{,xx} \sin(\beta + \phi)) \\ &+ EI_{\eta\zeta} (w_{,xx} \cos(\beta + \phi) - v_{,xx} \sin(\beta + \phi)) - \frac{1}{2} \underline{\underline{EAC_1}} (\phi_{,x})^2 \end{aligned}$$

The torque resultants,  $\bar{S}_x$  and  $\bar{S}_x'$ , are used to couple the first- and second-order elastic twist effects.  $\bar{S}_x$  includes terms of order  $\epsilon^3$  and  $\epsilon^4$  because it couples the first order of elastic twist.  $\bar{S}_x'$  includes terms of  $\epsilon^3$  only, because it couples the second order of elastic twist.

$$\begin{aligned} \bar{S}_x &= \iint_A \left\{ \psi_{,\zeta} + \eta \right\} G \gamma_{x\zeta} + \left\{ \psi_{,\eta} - \zeta \right\} G \gamma_{x\eta} d\eta d\zeta \quad \left. \right\} (2.6-47g) \\ &= GJ (\phi_{,x} + \phi_0) \end{aligned}$$

$$\begin{aligned} \bar{S}_x' &= \iint_A \left\{ \psi_{,\zeta} + \eta \right\} G \gamma_{x\zeta} + \left\{ \psi_{,\eta} - \zeta \right\} G \gamma_{x\eta} d\eta d\zeta \quad \left. \right\} (2.6-47h) \\ &= GJ \phi_{,x} \end{aligned}$$

The section integrals in Equations (2.6-47a-h) are defined as:

Modulus weighted area, first and second moments of inertia, and torsion integrals.

$$EA = \iint_A E d\eta d\zeta ; \quad EI_{\eta\eta} = \iint_A E \zeta^2 d\eta d\zeta \quad (2.6-48a,b)$$

$$EA\eta_A = \iint_A E \eta \, d\eta \, d\zeta ; \quad EI_{\zeta\zeta} = \iint_A E \eta^2 \, d\eta \, d\zeta \quad (2.6-48c,d)$$

$$EA\zeta_A = \iint_A E \zeta \, d\eta \, d\zeta ; \quad EI_{\eta\eta} = \iint_A E \eta \zeta \, d\eta \, d\zeta \quad (2.6-48e,f)$$

$$EAC_0 = \iint_A E (\eta^2 + \zeta^2) \, d\eta \, d\zeta ; \quad EAC_1 = \iint_A E \eta (\eta^2 + \zeta^2) \, d\eta \, d\zeta \quad (2.6-48g,h)$$

$$EAC_2 = \iint_A E \zeta (\eta^2 + \zeta^2) \, d\eta \, d\zeta ; \quad EAC_3 = \iint_A E (\eta^2 + \zeta^2)^2 \, d\eta \, d\zeta \quad (2.6-48i,j)$$

$$GJ = \iint_A G (\psi_{,\zeta} + \eta)^2 + G (\psi_{,\eta} - \zeta)^2 \, d\eta \, d\zeta \quad (2.6-48k)$$

Modulus weighted area and first moment warping integrals:

$$EAD_0 = \iint_A E \psi \, d\eta \, d\zeta ; \quad EAD'_0 = \iint_A E \psi_{,x} \, d\eta \, d\zeta \quad (2.6-49a,b)$$

$$EAD_1 = \iint_A E \psi \eta \, d\eta \, d\zeta ; \quad EAD'_1 = \iint_A E \psi_{,x} \eta \, d\eta \, d\zeta \quad (2.6-49c,d)$$

$$EAD_2 = \iint_A E \psi \zeta \, d\eta \, d\zeta ; \quad EAD'_2 = \iint_A E \psi_{,x} \zeta \, d\eta \, d\zeta \quad (2.6-49e,f)$$

$$EAD_3 = \iint_A E \psi^2 \, d\eta \, d\zeta ; \quad EAD'_3 = \iint_A E \psi_{,x}^2 \, d\eta \, d\zeta \quad (2.6-49g,h)$$

$$EAD_4 = \iint_A E \psi (\eta^2 + \zeta^2) \, d\eta \, d\zeta ; \quad EAD'_4 = \iint_A E \psi_{,x} (\eta^2 + \zeta^2) \, d\eta \, d\zeta \quad (2.6-49i,j)$$

$$EAD_5 = \iint_A E \psi \psi_{,x} \, d\eta \, d\zeta \quad (2.6-49k)$$

The section integrals, which are evaluated over the beam cross section, are defined as follows. The modulus weighted area is defined as  $EA$ . The first moments of the modulus weighted area are defined as  $EA\eta_A$  and  $EA\zeta_A$ , where  $\eta_A$  and  $\zeta_A$  are the distances from the shear center to the modulus weighted area center along  $\hat{e}_\eta$  and  $\hat{e}_\zeta$ , respectively. The second moments of the modulus weighted area about the shear center are defined as  $EI_{\eta\eta}$ ,  $EI_{\zeta\zeta}$ , and  $EI_{\eta\zeta}$ . Higher order constants that are necessary for modeling the axial-torsion coupling of the beam are defined as  $EAC_0$ ,  $EAC_1$ ,  $EAC_2$ , and  $EAC_3$ .  $EAC_0$  represents a higher order coupling of the axial and planar bending deformations with the nonlinear torsional twist.  $EAC_1$  and  $EAC_2$  are coupling terms between the torsional twist and the axial deflection produced by planar bending deformations.  $EAC_3$  is a higher order term that alters the nonlinear torsional stiffness of the beam. As previously mentioned, these constants are negligible when applying the ordering scheme in a strict fashion, however they are retained in order to capture these terms which are present in previous studies (Ref. 2.6-2, 2.6-11, and 2.6-13). The quantity  $GJ$  is the modulus weighted torsion constant.

The modulus weighted warping integrals are defined in Equations (2.6-49a-k). The quantity  $EAD_0$  represents the linear and nonlinear coupling of the axial deflections and the warping deflections. The term,  $EAD_0'$  is used to couple the axial deflections with the variation of the warping function over the beam length. This warping function variation is produced by either a change in the cross section properties or pretwist along the beam length. The constants,  $EAD_1$  and  $EAD_2$ , are terms which couple the axial strain due to warping with the axial strain produced by planar bending.  $EAD_1'$  and  $EAD_2'$  represent coupling between the axial strain produced by planar deformation and the axial strain produced by the variation of the warping function along the beam length (i.e., beam pretwist). The quantities  $EAD_3$ ,  $EAD_3'$ , and  $EAD_5$  represent the change of the torsional stiffness of the beam due to cross section warping and/or varying cross section properties (i.e., beam pretwist). The terms  $EAD_4$  and  $EAD_4'$  are higher order torsion coupling terms. Some of these terms have been defined by other authors for either isotropic materials or symmetric cross sections (Ref. 2.6-2).

For the case of a uniform pretwisted beam, some of the warping

For the case of a uniform pretwisted beam, some of the warping integrals can be written in an alternate form by using the identity in Equation (2.6-41):

$$\psi_{,x} = \left\{ \psi_{,\eta} s - \psi_{,s} \eta \right\} \tau_0 \quad (2.6-50)$$

substituting in:

$$EAD'_0 = \tau_0 \iint_A E \left\{ \psi_{,\eta} s - \psi_{,s} \eta \right\} d\eta ds \quad (2.6-51a)$$

$$EAD'_1 = \tau_0 \iint_A E \left\{ \psi_{,\eta} s - \psi_{,s} \eta \right\} \eta d\eta ds \quad (2.6-51b)$$

$$EAD'_2 = \tau_0 \iint_A E \left\{ \psi_{,\eta} s - \psi_{,s} \eta \right\} s d\eta ds \quad (2.6-51c)$$

$$EAD'_3 = \tau_0^2 \iint_A E \left\{ \psi_{,\eta} s - \psi_{,s} \eta \right\}^2 d\eta ds \quad (2.6-51d)$$

$$EAD'_4 = \tau_0 \iint_A E \left\{ \psi_{,\eta} s - \psi_{,s} \eta \right\} (\eta^2 + s^2) d\eta ds \quad (2.6-51e)$$

$$EAD'_5 = \tau_0 \iint_A E \left\{ \psi_{,\eta} s - \psi_{,s} \eta \right\} \psi d\eta ds \quad (2.6-51f)$$

Integrating the strain energy by parts:

$$\delta U = \int_0^L (Y_u) \delta u + (Y_v) \delta v + (Y_w) \delta w + (Y_\phi) \delta \phi dx + b(U) \quad (2.6-52)$$

where

$$Y_u = - \left\{ \bar{V}_x \right\}_{,x} \quad (2.6-53a)$$

$$Y_v = \left\{ \bar{M}_y \sin(\beta + \phi) + \bar{M}_z \cos(\beta + \phi) - \bar{S}'_x \cos \beta (v_{,x} \sin \beta - w_{,x} \cos \beta) \right\}_{,xx}$$

$$+ \left\{ -\bar{V}_x v_{,x} + \bar{S}_x^i \sin\theta (v_{,xx} \cos\theta + w_{,xx} \sin\theta) \right\}_{,x} \quad (2.6-53b)$$

$$Y_w = - \left\{ \bar{M}_y \cos(\theta + \phi) - \bar{M}_z \sin(\theta + \phi) + \bar{S}_x^i \sin\theta (v_{,x} \sin\theta - w_{,x} \cos\theta) \right\}_{,xx} \\ + \left\{ -\bar{V}_x w_{,x} - \bar{S}_x^i \cos\theta (v_{,xx} \cos\theta + w_{,xx} \sin\theta) \right\}_{,x} \quad (2.6-53c)$$

$$Y_\phi = v_{,xx} (\bar{M}_y \cos(\theta + \phi) - \bar{M}_z \sin(\theta + \phi)) - \left\{ \bar{S}_x + \bar{P}_x^i + \frac{\bar{T}_x \phi_{,x}}{\underline{\underline{\phi_{,x}}}} \right\}_{,x} \\ + w_{,xx} (\bar{M}_y \sin(\theta + \phi) + \bar{M}_z \cos(\theta + \phi)) + \left\{ \bar{P}_x \right\}_{,xx} \quad (2.6-53d)$$

writing the equilibrium conditions at the boundary (b(U)):

$$b(U) = \left\{ \bar{V}_x \right\}_{,x} \delta u \Big|_0^L + \left\{ \bar{S}_x + \bar{P}_x^i - \bar{P}_{x,x} + \frac{\bar{T}_x \phi_{,x}}{\underline{\underline{\phi_{,x}}}} \right\} \delta \phi \Big|_0^L + \left\{ \bar{P}_x \right\} \delta \phi_{,x} \Big|_0^L \\ + \left\{ \bar{V}_x v_{,x} - \bar{S}_x^i \sin\theta (v_{,xx} \cos\theta + w_{,xx} \sin\theta) \right\} \delta v \Big|_0^L \\ - \left\{ \bar{M}_y \sin(\theta + \phi) + \bar{M}_z \cos(\theta + \phi) - \bar{S}_x^i \cos\theta (v_{,x} \sin\theta - w_{,x} \cos\theta) \right\}_{,x} \delta v \Big|_0^L \\ + \left\{ \bar{M}_y \sin(\theta + \phi) + \bar{M}_z \cos(\theta + \phi) - \bar{S}_x^i \cos\theta (v_{,x} \sin\theta - w_{,x} \cos\theta) \right\} \delta v_{,x} \Big|_0^L \\ + \left\{ \bar{V}_x w_{,x} + \bar{S}_x^i \cos\theta (v_{,xx} \cos\theta + w_{,xx} \sin\theta) \right\} \delta w \Big|_0^L \\ + \left\{ \bar{M}_y \cos(\theta + \phi) - \bar{M}_z \sin(\theta + \phi) + \bar{S}_x^i \sin\theta (v_{,x} \sin\theta - w_{,x} \cos\theta) \right\}_{,x} \delta w \Big|_0^L$$

$$- \left\{ \bar{M}_y \cos(\beta + \phi) - \bar{M}_z \sin(\beta + \phi) + \bar{S}'_x \sin\beta (v_{,x} \sin\beta - w_{,x} \cos\beta) \right\} \delta w_{,x} \Big|_0^L$$

(2.6-54)

The strain energy variation,  $\delta U$  (Equations 2.6-52 and 2.6-53), and the associated conditions at the boundary,  $b(W_e)$  (Equation 2.6-54), contain many of the same basic terms as those derived by other authors (Ref. 2.6-2 and 2.6-3). The additional terms in this development are due to the following assumptions: (1) the cross section is not symmetric, (2) the effects of beam pretwist and cross section warping are fully included, and (3) the strain energy is derived using a complete moderate deflection theory and the second-order effects of the shear strain are included.

Three new stress resultants are developed in this study which do not appear in previous derivations. The first term,  $T_x$ , appears in the torsion variation,  $Y_\phi$ , and is used to account for the torsion twist of the beam due to axial loads. This term is defined as higher order based on the ordering scheme and should be neglected, however for rods which experience considerable axial loads (i.e., centrifugal forces) this term should be retained. The second term,  $P_x$ , also appears in the torsion variation  $Y_\phi$  and is used to account for the variation of the cross section warping function along the length of the beam. The third term,  $S_x$ , which appears in the beam bending variations represents the coupling between the beam bending deformations ( $v, w$ ) and the elastic torsion twist ( $\phi$ ).

#### 2.6.4.2 Kinetic Energy Contributions

The total kinetic energy of the beam is defined as:

$$T = \frac{1}{2} \int_0^L \iiint_A \rho \hat{V} \cdot \hat{V} \, d\eta \, d\zeta \, dx$$

(2.6-55)

where  $V$  is the velocity vector of an arbitrary point on the beam with respect to the inertial reference frame (I,J,K).

Since it is assumed that the beam rotates about the spin axis,  $K$ , with a constant velocity,  $\Omega$ , the velocity vector,  $V$ , will have the form:

$$\hat{V} = \dot{\hat{r}}_0 + \Omega \hat{K} \times \hat{r}_0 \quad (2.6-56)$$

where  $\hat{r}_0$  is the position vector of a point on the deformed cross section of the beam in the blade coordinate system  $(\hat{i}, \hat{j}, \hat{k})$ . The blade coordinate system rotates with the hub about the spin axis with a constant angular velocity,  $\Omega$ .

Writing the vector  $\hat{r}_0$ :

$$\hat{r}_0 = \{ h_i \hat{i} + h_j \hat{j} + h_k \hat{k} \} + \hat{R} \quad (2.6-57)$$

where  $h_i$ ,  $h_j$ , and  $h_k$  are the translational offsets of the beam defined in the blade coordinate system and  $\hat{R}$  is defined in Equation (2.6-37). The position vector  $\hat{r}_0$  can also be expressed in terms of the element coordinate system  $(\hat{e}_x, \hat{e}_y, \hat{e}_z)$  using the transformation matrices defined in Equations (2.6-4) and (2.6-5c).

$$\hat{r}_0 = \{ (h_x + x + u, h_y + v, h_z + w) + (\psi, \tau', \eta, \zeta) [ T_{DE} ] \} \begin{Bmatrix} \hat{e}_x \\ \hat{e}_y \\ \hat{e}_z \end{Bmatrix} \quad (2.6-58)$$

where  $h_x$ ,  $h_y$ , and  $h_z$  are the translational offsets of the beam from the origin of the blade coordinate system to the origin of the beam coordinate system. Expressing the offsets in terms of the element coordinate system using Equation (2.6-5c):

$$(h_x, h_y, h_z) = (h_i, h_j, h_k) [ T_{EB} ]^T \quad (2.6-59a)$$

and the transformation from the element coordinate system to the vector set on the deformed beam,  $[T_{DE}]$ , is defined using Equations (2.6-4) and (2.6-25):

$$[ T_{DE} ] = [ T_{DC} ] [ T_{CE} ] \quad (2.6-59b)$$



This matrix multiplication is carried out using Equations (2.6-4) and (2.6-28) and substituting for the element deformations (u,v,w).

$$[ T_{DE} ] =$$

$$\begin{bmatrix} 1 & v_{,x} & w_{,x} \\ -v_{,x} \cos(\beta + \phi) - w_{,x} \sin(\beta + \phi) & \cos(\beta + \phi) & \sin(\beta + \phi) \\ -w_{,x} \cos(\beta + \phi) + v_{,x} \sin(\beta + \phi) & -\sin(\beta + \phi + \tau'_c) & \cos(\beta + \phi) \end{bmatrix}$$

where

$$\tau'_c = \{v_{,x} \cos\beta + w_{,x} \sin\beta\} \{w_{,x} \cos\beta - v_{,x} \sin\beta\}$$

The spin vector is transformed to the element coordinate system using Equations (2.6-1) through (2.6-3):

$$\Omega \hat{k}_o = \Omega_x \hat{e}_x + \Omega_y \hat{e}_y + \Omega_z \hat{e}_z \quad (2.6-60a)$$

where  $\Omega_x$ ,  $\Omega_y$ , and  $\Omega_z$  are defined as:

$$(\Omega_x, \Omega_y, \Omega_z) = (0, 0, \Omega) [ T_{HI} ]^T [ T_{BH} ]^T [ T_{EB} ]^T \quad (2.6-60b)$$

The pitch, precone, and sweep settings of the blade, as well as orientation of the beam element with respect to the blade coordinate system, are implicitly defined within the rotational constants;  $\Omega_x$ ,  $\Omega_y$ , and  $\Omega_z$ .

The total velocity vector,  $\hat{V}$ , is calculated by substituting in the time derivative of Equation (2.6-58) along with the cross product of the spin vector (Equation (2.6-60a)) and the vector  $r_o$  (Equation (2.6-58)). The ordering scheme is used and terms of order  $\epsilon^1$  and  $\epsilon^2$  are retained.

$$\hat{V} = \{ \dot{u} + \Omega_y \{ h_z + w \} + (\Omega_y - \dot{w}_{,x}) \{ \zeta \cos(\beta + \underline{\phi}) + \eta \sin(\beta + \underline{\phi}) \} \}$$

$$\begin{aligned}
& -\Omega_z\{h_y + v\} - (\Omega_z + \dot{v}_{,x})\{\eta \cos(\beta + \underline{\phi}) - \zeta \sin(\beta + \underline{\phi})\} \hat{e}_x \\
& + \{\dot{v} + \Omega_z\{h_x + x + u\} - (\Omega_x + \dot{\phi})\{\zeta \cos(\beta + \underline{\phi}) + \eta \sin(\beta + \underline{\phi})\} \\
& -\Omega_x\{h_z + w\} - \Omega_z\{(\eta v_{,x} + \zeta w_{,x})\cos\beta + (\eta w_{,x} - \zeta v_{,x})\sin\beta\} \hat{e}_y \\
& + \{\dot{w} - \Omega_y\{h_x + x + u\} + (\Omega_x + \dot{\phi})\{\eta \cos(\beta + \underline{\phi}) - \zeta \sin(\beta + \underline{\phi})\} \\
& +\Omega_x\{h_y + v\} + \Omega_y\{(\eta v_{,x} + \zeta w_{,x})\cos\beta + (\eta w_{,x} - \zeta v_{,x})\sin\beta\} \hat{e}_z
\end{aligned} \tag{2.6-61}$$

The variation of the kinetic energy,  $\delta T$ , for the beam is defined as:

$$\delta T = \int_0^L \iint_A \rho \hat{V} \cdot \delta \hat{V} \, d\eta \, d\zeta \, dx \tag{2.6-62}$$

where  $\hat{V}$ , is the total velocity vector (Equation (2.6-61)),  $\delta \hat{V}$  is the variation of the total velocity vector, and  $\rho$  is the mass density of the beam.

The variation of the kinetic energy,  $\delta T$ , is calculated by taking the dot product of the velocity vector,  $\hat{V}$ , with the variation of the velocity vector,  $\delta \hat{V}$ , integrating by parts with respect to time, and finally integrating over the cross section. From Hamilton's principle, the variation of the kinetic energy,  $\delta T$ , is an extremum over the time integral from  $t_1$  to  $t_2$ .

Since the velocity vector and the variation of the velocity vector are both of order  $\epsilon^1$  and  $\epsilon^2$ , the resulting variation of the kinetic energy,  $\delta T$ , will be of order  $\epsilon^2$ ,  $\epsilon^3$ , and  $\epsilon^4$ . Consistent with the ordering scheme, terms of order  $\epsilon^4$  are neglected due to the presence of  $\epsilon^2$  terms.

Integrating  $\delta T$  by parts with respect to time between two arbitrary points,  $t_1$  and  $t_2$ , and then over the cross section, one obtains:

$$\delta T = \int_0^L (\bar{Z}_u \delta u + \bar{Z}_v \delta v + \bar{Z}'_v \delta v_{,x} + \bar{Z}_w \delta w + \bar{Z}'_w \delta w_{,x} + \bar{Z}_\phi \delta \phi) dx \quad (2.6-63)$$

where,

$$\begin{aligned} \bar{Z}_u = m \left\{ \ddot{u} - 2 (\Omega_y \dot{w} - \Omega_z \dot{v}) + (\Omega_y^2 + \Omega_z^2) (h_x + x + u) - \Omega_x \Omega_y (h_y + v) \right. \\ \left. - \Omega_x \Omega_z (h_z + w) \right\} - (m\eta_m \cos\beta - m\zeta_m \sin\beta) \{ \Omega_x \Omega_y \} \\ - (m\zeta_m \cos\beta + m\eta_m \sin\beta) \{ \Omega_x \Omega_z \} \end{aligned} \quad (2.6-64a)$$

$$\begin{aligned} \bar{Z}_v = m \left\{ \ddot{v} + 2 (\Omega_x \dot{w} - \Omega_z \dot{u}) - \Omega_x \Omega_y (h_x + x + u) + (\Omega_x^2 + \Omega_z^2) (h_y + v) \right. \\ \left. - \Omega_y \Omega_z (h_z + w) \right\} + (m\eta_m \cos\beta - m\zeta_m \sin\beta) \{ 2 (\Omega_x \dot{\phi} + \Omega_z \dot{v}_{,x}) \\ + \Omega_x \Omega_y v_{,x} + (\Omega_x^2 + \Omega_z^2) - \phi \Omega_y \Omega_z \} + (m\zeta_m \cos\beta + m\eta_m \sin\beta) \{ \dot{\phi} \\ + 2 \Omega_z \dot{w}_{,x} + \Omega_x \Omega_y w_{,x} - \phi (\Omega_x^2 + \Omega_z^2) - \Omega_y \Omega_z \} \end{aligned} \quad (2.6-64b)$$

$$\begin{aligned} \bar{Z}'_v = (m\eta_m \cos\beta - m\zeta_m \sin\beta) \left\{ 2 (\Omega_y \dot{w} - \Omega_z \dot{v}) - (\Omega_y^2 + \Omega_z^2) (h_x + x) \right. \\ \left. + \Omega_x \Omega_y (h_y + v) + \Omega_x \Omega_z (h_z + w) \right\} - (m\zeta_m \cos\beta + m\eta_m \sin\beta) \{ \Omega_y h_z \dot{\phi} \\ - \Omega_z h_y \dot{\phi} - \phi \{ (\Omega_y^2 + \Omega_z^2) (h_x + x) - \Omega_x \Omega_y h_y - \Omega_x \Omega_z h_z \} \} \\ + \Omega_x \Omega_y \{ \text{Im}_{\zeta\zeta} \cos^2\beta + \text{Im}_{\eta\eta} \sin^2\beta - 2 \text{Im}_{\eta\zeta} \cos\beta \sin\beta \} \end{aligned}$$

$$+ \Omega_x \Omega_z \{ \text{Im} \eta_{\zeta} (\cos^2 \theta - \sin^2 \theta) + (\text{Im} \zeta_{\zeta} - \text{Im} \eta_{\eta}) \cos \theta \sin \theta \} \quad (2.6-64c)$$

$$\begin{aligned} \bar{Z}_w^i = m \{ & -\dot{w} + 2 (\Omega_y \dot{u} - \Omega_x \dot{v}) - \Omega_x \Omega_z (h_x + x + u) - \Omega_y \Omega_z (h_y + v) \\ & + (\Omega_x^2 + \Omega_y^2)(h_z + w) \} + (m \zeta_m \cos \theta + m \eta_m \sin \theta) \{ 2 (\Omega_x \dot{\phi} - \Omega_y \dot{w}_{,x}) \\ & + \Omega_x \Omega_z w_{,x} + (\Omega_x^2 + \Omega_y^2) + \phi \Omega_y \Omega_z \} - (m \eta_m \cos \theta - m \zeta_m \sin \theta) \{ \dot{\phi} \\ & + 2 \Omega_y \dot{v}_{,x} - \Omega_x \Omega_z v_{,x} - \phi (\Omega_x^2 + \Omega_y^2) + \Omega_y \Omega_z \} \end{aligned} \quad (2.6-64d)$$

$$\begin{aligned} \bar{Z}_w^i = (m \zeta_m \cos \theta + m \eta_m \sin \theta) \{ & 2 (\Omega_y \dot{w} - \Omega_z \dot{v}) - (\Omega_y^2 + \Omega_z^2)(h_x + x) \\ & + \Omega_x \Omega_y (h_y + v) + \Omega_x \Omega_z (h_z + w) \} + (m \eta_m \cos \theta - m \zeta_m \sin \theta) \{ \Omega_y h_z \dot{\phi} \\ & - \Omega_z h_y \dot{\phi} - \phi \{ (\Omega_y^2 + \Omega_z^2)(h_x + x) - \Omega_x \Omega_y h_y + \Omega_x \Omega_z h_z \} \} \\ & + \Omega_x \Omega_z \{ \text{Im} \eta_{\eta} \cos^2 \theta + \text{Im} \zeta_{\zeta} \sin^2 \theta + 2 \text{Im} \eta_{\zeta} \cos \theta \sin \theta \} \\ & + \Omega_x \Omega_y \{ \text{Im} \eta_{\zeta} (\cos^2 \theta - \sin^2 \theta) + (\text{Im} \zeta_{\zeta} - \text{Im} \eta_{\eta}) \cos \theta \sin \theta \} \end{aligned} \quad (2.6-64e)$$

$$\begin{aligned} \bar{Z}_{\phi} = & -(m \eta_m \cos \theta - m \zeta_m \sin \theta) \{ \dot{w} + 2 \Omega_x \dot{v} + (\Omega_y h_z - \Omega_z h_y) \dot{w}_{,x} \\ & + \Omega_x \Omega_z (h_x + x) + \Omega_y \Omega_z (h_y + v) - (\Omega_x^2 + \Omega_y^2)(h_z + w) \\ & + w_{,x} \{ (\Omega_y^2 + \Omega_z^2)(h_x + x) - \Omega_x \Omega_y h_y - \Omega_x \Omega_z h_z \} \\ & - \phi \{ \Omega_x \Omega_y (h_x + x) - (\Omega_x^2 + \Omega_z^2) h_y + \Omega_y \Omega_z h_z \} \} \end{aligned}$$

$$\begin{aligned}
& + (m\zeta_m \cos\beta + m\eta_m \sin\beta) \left\{ \dot{v} - 2 \Omega_x \dot{w} + (\Omega_y h_z - \Omega_z h_y) \dot{v}_{,x} \right. \\
& + \Omega_x \Omega_y (h_x + x) - (\Omega_x^2 + \Omega_z^2)(h_y + v) + \Omega_y \Omega_z (h_z + w) \\
& + v_{,x} \{ (\Omega_y^2 + \Omega_z^2)(h_x + x) - \Omega_x \Omega_y h_y - \Omega_x \Omega_z h_z \} \\
& \left. + \phi \{ \Omega_x \Omega_z (h_x + x) + \Omega_y \Omega_z h_y - (\Omega_x^2 + \Omega_y^2) h_z \} \right\} - \underline{\underline{(\text{Im}_{\eta\eta} + \text{Im}_{\zeta\zeta}) \dot{\phi}}} \\
& - (\Omega_x^2 + \Omega_z^2) \{ \text{Im}_{\eta\zeta} (\cos^2\beta - \sin^2\beta) + (\text{Im}_{\zeta\zeta} - \text{Im}_{\eta\eta}) \cos\beta \sin\beta \} \\
& + (\Omega_x^2 + \Omega_y^2) \{ \text{Im}_{\eta\zeta} (\cos^2\beta - \sin^2\beta) + (\text{Im}_{\zeta\zeta} - \text{Im}_{\eta\eta}) \cos\beta \sin\beta \} \\
& + \Omega_y \Omega_z \{ \text{Im}_{\eta\eta} \cos^2\beta + \text{Im}_{\zeta\zeta} \sin^2\beta + 2 \text{Im}_{\eta\zeta} \cos\beta \sin\beta \} \\
& - \Omega_y \Omega_z \{ \text{Im}_{\zeta\zeta} \cos^2\beta + \text{Im}_{\eta\eta} \sin^2\beta - 2 \text{Im}_{\eta\zeta} \cos\beta \sin\beta \} \quad (2.6-64f)
\end{aligned}$$

where the double underlined terms are of order  $\epsilon^4$  and, based on the ordering scheme, should be neglected. These terms are retained in this development to correctly model the axial inertia, axial centrifugal forces, and the torsion inertia.

The section integrals in Equations (5.64a-f) are defined as:

$$m = \iint_A \rho \, d\eta \, d\zeta ; \quad \text{Im}_{\eta\eta} = \iint_A \rho \, \zeta^2 \, d\eta \, d\zeta \quad (2.6-65a,b)$$

$$m\eta_m = \iint_A \rho \, \eta \, d\eta \, d\zeta ; \quad \text{Im}_{\zeta\zeta} = \iint_A \rho \, \eta^2 \, d\eta \, d\zeta \quad (2.6-65c,d)$$

$$m\zeta_m = \iint_A \rho \, \zeta \, d\eta \, d\zeta ; \quad \text{Im}_{\eta\zeta} = \iint_A \rho \, \eta \, \zeta \, d\eta \, d\zeta \quad (2.6-65e,f)$$

where  $m$  is equal to the mass of the cross section. The first mass moments of the cross section about the shear center are defined as  $m\eta_m$  and  $m\zeta_m$ . The second mass moments of the cross section about the shear center are defined as  $Im_{\eta\eta}$ ,  $Im_{\zeta\zeta}$ , and  $Im_{\eta\zeta}$ .

Integrating the kinetic energy (Equation 2.6-63) by parts:

$$\delta T = \int_0^L \left\{ \bar{Z}_u \delta u + (\bar{Z}_v - \bar{Z}'_{v,x}) \delta v + (\bar{Z}_w - \bar{Z}'_{w,x}) \delta w + \bar{Z}_\phi \delta \phi \right\} dx + b(T) \quad (2.6-66)$$

writing the equilibrium conditions at the boundary,  $b(T)$ :

$$b(T) = \bar{Z}'_v \delta v \Big|_0^L + \bar{Z}'_w \delta w \Big|_0^L \quad (2.6-67)$$

#### 2.6.4.3 External Work Contributions

The principal of virtual work is used to include the effects of non-conservative distributed loads and moments that act along the line of shear centers. These distributed loads and moments are defined in the element coordinate system  $(\hat{e}_x, \hat{e}_y, \hat{e}_z)$ . Virtual work is the work done by the distributed loads and moments acting through virtual displacements and rotations about the deformed equilibrium position.

The virtual work due to the distributed loads along the line of shear centers, is calculated by taking the dot product of the real distributed forces and the virtual displacements and integrating over the beam length.

$$\delta W_{e_p} = \int_0^L (p_x \delta u + p_y \delta v + p_z \delta w) dx \quad (2.6-68)$$

The virtual work, due to the distributed moments along the line of shear centers, is calculated by taking the dot product of the real distributed moments with the virtual rotations and integrating over the beam length.

$$\delta W_{e_Q} = \int_0^L (q_y w_{,x} - q_z v_{,x}) \delta u_{,x} + (q_z - q_x w_{,x}) \delta v_{,x}$$

$$+ (q_x v_{,x} - q_y) \delta w_{,x} + (q_x + q_y v_{,x} + q_z w_{,x}) \delta \phi \, dx \quad (2.6-69)$$

The total virtual work is calculated by combining the virtual work due to distributed loads (Equation (2.6-68)) and the virtual work due to distributed moments (Equation (2.6-69)) and integrating by parts.

$$\begin{aligned} \delta W_e = & \int_0^L \{p_x - (q_y w_{,x} - q_z v_{,x})_{,x}\} \delta u + \{p_y - (q_z - q_x w_{,x})_{,x}\} \delta v \\ & + \{p_z + (q_y - q_x v_{,x})_{,x}\} \delta w + (q_x + q_y v_{,x} + q_z w_{,x}) \delta \phi \, dx \\ & + b(W_e) \end{aligned} \quad (2.6-70)$$

writing the equilibrium conditions at the boundary,  $b(W_e)$ :

$$b(W_e) = (q_y w_{,x} - q_z v_{,x}) \delta u \Big|_0^L + (q_z - q_x w_{,x}) \delta v \Big|_0^L + (q_x v_{,x} - q_y) \delta w \Big|_0^L \quad (2.6-71)$$

#### 2.6.4.4 Summary of the Partial Differential Equations of Motion and Boundary Conditions

The partial differential equations of motion and the associated boundary conditions are calculated by substituting the variation of the strain energy,  $\delta U$  (Equation 2.6-52), the variation of the kinetic energy,  $\delta T$  (Equation 2.6-66), and the virtual work of the external loads,  $\delta W_e$  (Equation 2.6-70), into Hamilton's principle. There are two requirements that must be met in order for Hamilton's principle to be identically equal to zero and thus, the total dynamic potential to be an extremum.

The first is that all the terms inside of the integral must be identically equal to zero. Since the variations of the variables ( $\delta u, \delta v, \delta w, \delta \phi$ ) are arbitrary over the length of the beam then, in order for all of the terms to be equal to zero, the functions multiplied by these variations must be zero. The four functions which are associated with the four variations are the equations of motion of the beam.

Second, the sum of the equilibrium conditions at the boundary due to the strain energy variation,  $b(U)$ , the kinetic energy variation,  $b(T)$ , and

the virtual work,  $b(W_e)$ , must also be zero at  $x = 0$  and  $x = L$ . These conditions are used to obtain the boundary condition of the beam element.

The four equations of motion are:

$\delta u$  equation:

$$\{\bar{V}_x - q_y w_{,x} + q_z v_{,x}\}_{,x} + \bar{Z}_u + p_x = 0 \quad (2.6-72)$$

$\delta v$  equation:

$$\begin{aligned} & \{\bar{M}_y \sin(\beta + \phi) + \bar{M}_z \cos(\beta + \phi) - \bar{S}'_x \cos\beta \{v_{,x} \sin\beta - w_{,x} \cos\beta\}\}_{,xx} \\ & + \{\bar{S}'_x \sin\beta \{v_{,xx} \cos\beta + w_{,xx} \sin\beta\} - \bar{V}_x v_{,x} + \bar{Z}'_v - q_x w_{,x} + q_z\}_{,x} \\ & - \bar{Z}'_v - p_y = 0 \end{aligned} \quad (2.6-73)$$

$\delta w$  equation:

$$\begin{aligned} & \{\bar{M}_z \sin(\beta + \phi) - \bar{M}_y \cos(\beta + \phi) - \bar{S}'_x \sin\beta \{v_{,x} \sin\beta - w_{,x} \cos\beta\}\}_{,xx} \\ & - \{\bar{S}'_x \cos\beta \{v_{,xx} \cos\beta + w_{,xx} \sin\beta\} + \bar{V}_x w_{,x} - \bar{Z}'_w - q_x v_{,x} + q_y\}_{,x} \\ & - \bar{Z}'_w - p_z = 0 \end{aligned} \quad (2.6-74)$$

$\delta\phi$  equation:

$$\begin{aligned} & \{\bar{P}_x\}_{,xx} - \{\bar{S}_x + \bar{P}'_x + \underline{\underline{\bar{T}_x \phi_{,x}}}\}_{,x} + \{\bar{M}_y \sin(\beta + \phi) + \bar{M}_z \cos(\beta + \phi)\}_{,xx} \\ & \{\bar{M}_y \cos(\beta + \phi) - \bar{M}_z \sin(\beta + \phi)\}_{,xx} - \bar{Z}'_\phi - q_y v_{,x} - q_z w_{,x} - q_x = 0 \end{aligned} \quad (2.6-75)$$



The equilibrium conditions at the boundary are:

$$\begin{aligned}
 b(U) - b(T) - b(W_e) &= \left\{ \bar{v}_{x,x} + q_z v_{,x} - q_y w_{,x} \right\} \delta u \Big|_0^L \\
 &+ \left\{ \bar{M}_y \sin(\beta + \phi) + \bar{M}_z \cos(\beta + \phi) - \bar{S}'_x \cos\beta \{v_{,x} \sin\beta - w_{,x} \cos\beta\} \right\} \delta v_{,x} \Big|_0^L \\
 &+ \left\{ \left\{ \bar{v}_x v_{,x} - \bar{S}'_x \sin\beta \{v_{,xx} \cos\beta + w_{,xx} \sin\beta\} - \bar{Z}'_v + q_x w_{,x} - q_z \right\} \right. \\
 &- \left. \left\{ \bar{M}_y \sin(\beta + \phi) + \bar{M}_z \cos(\beta + \phi) - \bar{S}'_x \cos\beta \{v_{,x} \sin\beta - w_{,x} \cos\beta\} \right\}_{,x} \right\} \delta v \Big|_0^L \\
 &+ \left\{ \bar{M}_z \sin(\beta + \phi) - \bar{M}_y \cos(\beta + \phi) - \bar{S}'_x \sin\beta \{v_{,x} \sin\beta - w_{,x} \cos\beta\} \right\} \delta w_{,x} \Big|_0^L \\
 &+ \left\{ \left\{ \bar{v}_x w_{,x} + \bar{S}'_x \cos\beta \{v_{,xx} \cos\beta + w_{,xx} \sin\beta\} - \bar{Z}'_w - q_x v_{,x} + q_y \right\} \right. \\
 &- \left. \left\{ \bar{M}_z \sin(\beta + \phi) - \bar{M}_y \cos(\beta + \phi) - \bar{S}'_x \sin\beta \{v_{,x} \sin\beta - w_{,x} \cos\beta\} \right\}_{,x} \right\} \delta w \Big|_0^L \\
 &+ \left\{ \bar{p}_x \right\} \delta \phi_{,x} \Big|_0^L + \left\{ \bar{S}_x + \bar{p}'_x - \bar{p}_{x,x} + \underline{\bar{T}_x \phi_{,x}} \right\} \delta \phi \Big|_0^L \tag{2.6-76}
 \end{aligned}$$

where the stress resultants ( $V_x, P_x, P_x, T_x$ ) and the moment resultants ( $S_x, S_x, M_y, M_z$ ) have been defined in Equations (2.6-46a-f) and (2.6-47a,b). The inertial forces ( $Z_u, Z_v, Z_w$ ) and the inertial moments ( $Z_v, Z_w, Z_\phi$ ) are defined in Equations (2.6-64a-f).

### 2.6.5 Finite Element Discretization of the Equations of Motion Using Hamilton's Principle

The equations of motion, which have been derived in the previous section, are a set of nonlinear partial differential equations. They are nonlinear in terms of the space dependent variables. In order to do the free vibration analysis of a rotating propeller, these equations are solved by assuming that the displacements are a superposition of functions which

are space dependent and time dependent. The space-dependent displacement functions are determined by solving the time-independent nonlinear partial differential equations (i.e., neglect all time-dependent terms). Introducing this nonlinear space-dependent solution into the nonlinear partial differential equations results in a standard eigenvalue problem for the time-dependent function, the solution to which yields the natural frequencies and mode shapes of the rotating blade.

The solution of the nonlinear partial differential equations can be determined using approximate methods. Three methods that have been used successfully are the Ritz method, the Galerkin method, and the finite element method. A detailed discussion of these methods can be found in References 2.6-22 through 2.6-25. For this analysis, the finite element is applied.

The finite element method is a generalized form of the Ritz method with all of the advantages and few, if any, of the disadvantages. This method is a piecewise application of the variational method. The solution procedure is not based on solving the nonlinear differential equations of motion, but on minimizing the total dynamic potential of the rotating blade. The variation of the total dynamic potential with respect to the displacements is equal to zero when the total dynamic potential is minimized (i.e., apply Hamilton's principle to each subregion).

Writing Hamilton's principle in discretized form:

$$\int_{t_1}^{t_2} \sum_{i=1}^n (\delta U_i - \delta T_i - \delta W_{e_i}) dt = 0 \quad (2.6-77)$$

where

- n = number of finite elements in the model
- $\delta U_i$  = variation of the strain energy in the  $i^{\text{th}}$  element
- $\delta T_i$  = variation of the kinetic energy in the  $i^{\text{th}}$  element
- $\delta W_{e_i}$  = virtual work of the external forces in the  $i^{\text{th}}$  element

The solution for the displacement function will solve the nonlinear differential equations of motion and the natural boundary conditions which

are implicitly contained in the total dynamic potential. This has been shown using the calculus of variations (Ref. 2.6-20).

The blade is divided into subregions (beam type finite elements) and the variation of the total dynamic potential is applied to each subregion using a set of interpolation functions for the displacements. Since the interpolation functions are not defined over the entire domain, they are not required to solve the boundary conditions, but they have to satisfy convergence criteria (completeness and continuity).

Writing the displacements in terms of space dependent interpolation functions and time dependent functions:

$$\begin{Bmatrix} v \\ w \\ \phi \\ u \end{Bmatrix} = \begin{bmatrix} \{\Phi_V\} & 0 & 0 & 0 \\ 0 & \{\Phi_W\} & 0 & 0 \\ 0 & 0 & \{\Phi_\phi\} & 0 \\ 0 & 0 & 0 & \{\Phi_U\} \end{bmatrix} \begin{Bmatrix} \{V\} \\ \{W\} \\ \{\phi\} \\ \{U\} \end{Bmatrix} \quad (2.6-78)$$

where  $\{\Phi_V\}$ ,  $\{\Phi_W\}$ ,  $\{\Phi_\phi\}$ , and  $\{\Phi_U\}$  are space dependent interpolation functions and  $\{V\}$ ,  $\{W\}$ ,  $\{\phi\}$ , and  $\{U\}$  are time dependent functions of  $v$ ,  $w$ ,  $\phi$ , and  $u$ .

Similarly, the variation of the displacements are written as:

$$\begin{Bmatrix} \delta v \\ \delta w \\ \delta \phi \\ \delta u \end{Bmatrix} = \begin{bmatrix} \{\Phi_V\} & 0 & 0 & 0 \\ 0 & \{\Phi_W\} & 0 & 0 \\ 0 & 0 & \{\Phi_\phi\} & 0 \\ 0 & 0 & 0 & \{\Phi_U\} \end{bmatrix} \begin{Bmatrix} \{\delta V\} \\ \{\delta W\} \\ \{\delta \phi\} \\ \{\delta U\} \end{Bmatrix} \quad (2.6-79)$$

Substituting the displacements (Equation 2.6-78) and the variations of the displacements (Equation 2.6-79) into the  $i^{\text{th}}$  element of the discretized form of Hamilton's principle (Equation 2.6-77) and carrying out the integration will produce the following matrix equations:

$$\int_{t_1}^{t_2} \{\delta q\}^T \left\{ \left[ M \right] \{\dot{q}\} + \left[ M^C \right] \{\dot{q}\} + \left[ \left[ K^T \right] + \left[ K^{NL}\{q\} \right] \right] \{q\} + \{F^T\} \right\} dt = \{0\} \quad (2.6-80)$$

where,

$$\begin{aligned} \{q\} &= \text{Nodal deformation array} \\ \{q\}^T &= \{\{v\}^T, \{w\}^T, \{\phi\}^T, \{u\}^T\} \\ \{\delta q\} &= \text{Variation of the deformation array} \\ \{\delta q\}^T &= \{\{\delta v\}^T, \{\delta w\}^T, \{\delta \phi\}^T, \{\delta u\}^T\} \end{aligned}$$

All of the matrices that are produced represent a complete set of self-adjoint finite elements (i.e., symmetric linear and nonlinear Jacobian matrices and antisymmetric Coriolis matrix). This occurs because the ordering scheme is applied to each of the energies (strain energy, kinetic energy, potential energy) instead of the equations of motion.

Since the virtual displacements ( $\delta v, \delta w, \delta \phi, \delta u$ ) are arbitrary over the time integral, then the variation of the deformation array,  $\{\{\delta v\}^T, \{\delta w\}^T, \{\delta \phi\}^T, \{\delta u\}^T\}$ , is also arbitrary in the discretized form of Hamilton's principle and the equations of motion can be easily written from Equation (2.6-80):

$$[M] \{\dot{q}\} + [M^C] \{\dot{q}\} + \left[ [K^T] + [K^{NL}\{q\}] \right] \{q\} + \{F^T\} = \{0\} \quad (2.6-81)$$

In this development, the finite element matrices are generated using Hermite interpolation polynomials that are substituted into the variation of Hamilton's principle (Equation 2.6-42). A cubic Hermite interpolation polynomial is used for the planar deflections ( $v, w$ ) and a quadratic Hermite polynomial is used for the axial deflection ( $u$ ) and the torsional rotation ( $\phi$ ).

The cubic Hermite interpolation polynomial is ideally suited for the development of the beam bending finite element for the following reasons. It guarantees uniform convergence by having the required rigid body displacement, rigid body rotation, and constant curvature modes. The bending strains vary linearly over the element length, which goes beyond the requirement of a constant strain mode. The resulting elements formed using this polynomial will have interelement compatibility for both displacements and slopes. The nodal parameters are the displacements and slopes at either end of the beam.

The quadratic Hermite interpolation polynomial was chosen for the axial deflection ( $u$ ) and the torsional rotation ( $\phi$ ) because it has a higher level of accuracy than the linear Hermite interpolation polynomial. This same torsional model (i.e., quadratic polynomial) was used successfully in previous finite element studies of helicopter rotor aeroelasticity (Ref. 2.6-23). Both the linear and the quadratic polynomials satisfy all the convergence requirements, but the quadratic polynomial also has the capability of modeling the linear variation of strain along the element length. The quadratic polynomial is used to generate an axial and a torsion element which has the same level of accuracy as the beam bending element. This allows the discretization of the model for torsion or axially loaded analysis to be the same as for a bending analysis. The nodal parameters for the axial element and the torsion element are chosen as the two element boundaries and the element mid-point.

The resulting beam finite element has 14 nodal parameters; four in-plane deflections, four out-of-plane deflections, three axial deflections and three torsion deflections. They are defined as six parameters (three deflections, three rotations) at each end of the beam and two parameters (one axial deflection, one torsion deflection) at the mid-point of the element (Figure 2.6-6). The mid-point node is condensed from the element after formation using the assumption that the axial and torsion loads on the element can be fully described using the two end nodes only (i.e., external loads on the mid-point node are equal to zero). Condensing out the mid-point node will reduce size of the matrices and the bandwidth of the overall model without sacrificing the accuracy of the calculation.

The removal of the mid-point node is done by either static condensation (Ref. 2.6-24) or Guyan reduction (Ref. 2.6-26). In this study, static condensation is used to eliminate the element mid-point node from the static equations of equilibrium. Guyan reduction is used to eliminate the element mid-point node from the dynamic equations of motion. The beam finite element in its final form will have 12 nodal parameters; three deflections and three rotations at each end of the beam.

All of the beam finite elements are defined with a linear variation in properties and loads between the two end nodes. This will increase the

convergence rate and reduce the number of elements that are required for modeling nonuniform blades.

Assembly of the finite elements into system matrices and enforcement of the geometric boundary conditions is handled as in the conventional finite element method (Ref. 2.6-25).

#### 2.6.5.1 Finite Elements Associated With the Strain Energy Variation

The linear and nonlinear stiffness matrices are derived by substituting the assumed shape functions into the strain energy variation,  $\delta U$  (Equation 2.6-46), and carrying out the integration over the length of the beam. Hermite cubic interpolation functions  $\{\phi_C\}$  are used for the planar deformations  $(v,w)$  and for the variation of the planar deformations  $(\delta v, \delta w)$ . Hermite quadratic interpolation functions  $\{\phi_Q\}$  are used for the axial deformation and torsional twist  $(u, \phi)$  and also for the variations of the axial deformation and torsional twist  $(\delta u, \delta \phi)$ . The small angle assumption for  $\phi$  (Equation 2.6-31e) is used for all trigonometric relations of the strain energy variation. The matrix relations will be self-adjoint because the ordering scheme was applied to the strain energy.

The resulting variation of the the strain energy will have the form:

$$\delta U = \begin{Bmatrix} \{\delta V\} \\ \{\delta W\} \\ \{\delta \phi\} \\ \{\delta U\} \end{Bmatrix}^T \left[ \begin{matrix} K^L \end{matrix} \right] + \left[ \begin{matrix} K^{NL} \end{matrix} \right] \begin{Bmatrix} \{V\} \\ \{W\} \\ \{\phi\} \\ \{U\} \end{Bmatrix} \begin{Bmatrix} \{V\} \\ \{W\} \\ \{\phi\} \\ \{U\} \end{Bmatrix} \quad (2.6-82)$$

where,

$\left[ K^L \right]$  is a linear finite element stiffness matrix associated with elastic forces (symmetric), detailed expressions for this matrix are presented in Appendix A.1

$\left[ K^{NL}\{q\} \right]$  is a nonlinear finite element stiffness matrix associated with elastic forces (nonsymmetric), detailed expressions for this matrix are presented in Appendix A.1.

$$\{q\}^T = \{\{U\}, \{V\}, \{\phi\}, \{W\}\}$$

The solution of nonlinear finite element equations requires an iterative method [for example, Newton-Raphson method (Ref. 2.6-21), Brown's algorithm (Ref. 2.6-27), etc.]. Typical studies of this type for rotor blades (Ref. 2.6-23 and 2.6-28) have been based upon the Newton-Raphson method. Such methods require the calculation of the Jacobian matrix, which is the derivative of the of the linear and nonlinear elastic forces with respect to the deflections (v,w,φ,u).

Writing the linear and nonlinear elastic forces.

$$\left[ \left[ K^L \right] + \left[ K^{NL} \{q\} \right] \right] \{q\} \quad (2.6-83)$$

where

$$\{q\}^T = \{\{U\}, \{V\}, \{\phi\}, \{W\}\}$$

Defining the Jacobian as the derivative of the elastic forces with respect to the deflections

$$\left[ J^L \right] + \left[ J^{NL} \{q\} \right] = \left[ K^L \right] + \frac{\delta}{\delta q} \left\{ \left[ K^{NL} \{q\} \right] \{q\} \right\} \quad (2.6-84)$$

The linear Jacobian matrix is equal to the linear stiffness matrix:

$$\left[ J^L \right] = \left[ K^L \right] \quad (2.6-85)$$

The nonlinear Jacobian matrix is equal to the derivative of the nonlinear elastic forces with respect to the deflections.

$$\left[ J^{NL} \{q\} \right] = \frac{\delta}{\delta q} \left\{ \left[ K^{NL} \{q\} \right] \{q\} \right\} \quad (2.6-86)$$

where the terms which constitute the nonlinear Jacobian matrix are presented in Appendix A.1.

### 2.6.5.2 Finite Elements Associated With the Kinetic Energy Variation

All of the finite element matrices, which are associated with the kinetic energy of the rotating beam, are derived by substitution of the shape functions into the kinetic energy variation,  $\delta T$ , and integrating over the beam element length. The shape functions which are chosen are: the Hermite cubic interpolation functions  $\{\Phi_C\}$  for the planar deformations  $(v,w)$  and the variation of the planar deformations  $(\delta v, \delta w)$ , and the Hermite quadratic interpolation functions  $\{\Phi_Q\}$  for the axial deformation and torsional twist  $(u, \phi)$  and also for the variations of the axial deformation and torsional twist  $(\delta u, \delta \phi)$ .

The discretization of the kinetic energy leads to three linear matrices and one column matrix. The three matrices are a symmetric mass matrix, an antisymmetric Coriolis damping matrix, and a symmetric centrifugal stiffening matrix. The column matrix is the result of the centrifugal force. The small angle assumptions of Equation (2.6-31e) are used to linearize the trigonometric functions.

The following matrix relation is written by substituting the shape functions into the kinetic energy variation (Equation 2.6-63) and integrating over the element length.

$$\delta T = - \begin{Bmatrix} \delta v \\ \delta w \\ \delta \phi \\ \delta u \end{Bmatrix}^T \left\{ \begin{bmatrix} M \end{bmatrix} \begin{Bmatrix} v \\ w \\ \phi \\ u \end{Bmatrix} + \begin{bmatrix} M^C \end{bmatrix} \begin{Bmatrix} v \\ w \\ \phi \\ u \end{Bmatrix} + \begin{bmatrix} K^{CF} \end{bmatrix} \begin{Bmatrix} v \\ w \\ \phi \\ u \end{Bmatrix} + \begin{Bmatrix} F^{CF} \end{Bmatrix} \right\} \quad (2.6-87)$$

where

$$\begin{bmatrix} M \end{bmatrix} = \text{mass matrix (symmetric)}$$

$$\begin{bmatrix} M^C \end{bmatrix} = \text{Coriolis damping matrix (anti-symmetric)}$$

$$\begin{bmatrix} K^{CF} \end{bmatrix} = \text{centrifugal stiffening matrix (symmetric)}$$



$$\{F^{CF}\} = \text{centrifugal force array}$$

where detailed expressions for the three finite element matrices and the column matrix are presented in Appendix A.2.

### 2.6.5.3 Finite Elements Associated With the Virtual Work Variation

The finite elements associated with external work are calculated by substituting the Hermite interpolation functions into the definition of the variation of external work and integrating over the beam length.

$$\delta W_e = - \begin{Bmatrix} \delta V \\ \delta W \\ \delta \phi \\ \delta U \end{Bmatrix}^T \left[ [K^I] \begin{Bmatrix} V \\ W \\ \phi \\ U \end{Bmatrix} + \begin{Bmatrix} F^I \end{Bmatrix} \right] \quad (2.6-88)$$

where

$$[K^I] = \text{applied moment stiffening matrix}$$

$$\{F^I\} = \text{applied force array}$$

where detailed expressions for the matrices are presented in Appendix A.3.

### 2.6.5.4 Summary of Finite Element Equations

The equations of motion are written by substituting the strain energy variation (Equation 2.6-82), the kinetic energy variation (Equation 2.6-87), and the variation of the work of external forces (Equation 2.6-88) into the discretized form of Hamilton's principle (Equation 2.6-77). Since the variation of the displacements ( $\delta v, \delta w, \delta \phi, \delta u$ ) are arbitrary over the integral, this implies that the integral will equal zero only if the matrix equation, (Equation 2.6-80), is identically equal to zero, thus:

$$[M] \{\dot{q}\} + [M^C] \{\dot{q}\} + \left( [K^T] + [K^{NL}\{q\}] \right) \{q\} + \{F^T\} = \{0\} \quad (2.6-89)$$

where;

$$[K^T] = [K^L] + [K^{CF}] + [K^I]$$

$$\{F^T\} = \{F^{CF}\} + \{F^I\}$$

$$\{q\}^T = \{\{U\}, \{V\}, \{\phi\}, \{W\}\}$$

#### 2.6.6 Method of Solution

The solution of the structural dynamic analysis of the rotating propeller is accomplished in two steps. The first step is to solve for the nonlinear static equilibrium position of the blade using a Newton-Raphson iteration scheme; this procedure has been used in similar studies (Ref. 2.6-22 and 2.6-23). The second step is to solve for the natural frequencies of vibration of the nonlinear equations of motion by assuming that the motion is a small linear perturbation about the static equilibrium position.

##### 2.6.6.1 Static Analysis

The solution for the nonlinear static equilibrium position is accomplished by neglecting the time-dependent terms in the equations of motion and using a Newton-Raphson iteration scheme to solve the nonlinear equations of equilibrium. The Newton-Raphson scheme, which is based on a Taylor series expansion, has proven itself as one of the best solution techniques available in geometrically nonlinear analyses (Ref. 2.6-29) and is used routinely in rotor blade dynamics (Ref. 2.6-22 and 2.6-23). This method is extremely accurate and possesses second-order convergence. In this study the iteration is considered to have converged when the absolute change of all the deformations is less than  $10^{-4}$  for each cycle.

For the case of free vibration for a rotating blade, the nonlinear equilibrium state of the blade can be considered to be independent of time. Thus the static equations of equilibrium of the rotating blade are obtained by neglecting the time dependent terms of Equation (2.6-89):

$$\{f(q)\} = \left[ [K^T] + [K^{NL}\{q\}] \right] \{q\} + \{F^T\} = \{0\} \quad (2.6-90)$$

where  $[K^T]$ ,  $[K^{NL}\{q\}]$ ,  $\{q\}$ , and  $\{F^T\}$  have been defined in Equations (2.6-82) and (2.6-89).

The Newton-Raphson iteration scheme is applied by writing the linearized form of the Taylor series expansion of Equation (2.6-90). The increment for each step of the iteration is defined by  $n$ .

$$\{f(q)\}_{n+1} = \{f(q)\}_n + \frac{\delta}{\delta q} \{f(q)\}_n \left\{ \{q\}_{n+1} - \{q\}_n \right\} = \{0\} \quad (2.6-91)$$

Solving for the deflected shape at the new iteration step:

$$\{q\}_{n+1} = \{q\}_n + \left[ J\{q\}_n \right]^{-1} \{f(q)\}_n \quad (2.6-92)$$

where the matrix  $[J\{q\}]$  is defined as the Jacobian matrix.

$$\left[ J\{q\}_n \right] = \frac{\delta}{\delta q} \{f(q)\}_n \quad (2.6-93)$$

Taking the derivative of the function  $f(q)$  with respect to the deflected shape,  $\{q\}$ , in Equation (2.6-90) will produce two components of the Jacobian matrix; a linear Jacobian matrix and a nonlinear Jacobian matrix. Both components are symmetric matrices due to the application of the ordering scheme to the strain energy.

$$\left[ J\{q\}_n \right] = \left[ J^L \right] + \left[ J^{NL}\{q\}_n \right] \quad (2.6-94)$$

where the linear Jacobian matrix  $[J^L]$  is equal to the total linear stiffness matrix  $[K^T]$  and the nonlinear Jacobian matrix  $[J^{NL}\{q\}]$  is equal to the derivative of the nonlinear elastic forces with respect to the deflections:

$$\left[ J^L \right] = \left[ K^T \right] \quad (2.6-95a)$$

$$\left[ J^{NL}\{q\} \right]_n = \frac{\delta}{\delta q} \left\{ \left[ K^{NL}\{q\} \right] \{q\} \right\}_n \quad (2.6-95b)$$

The nonlinear equilibrium position for each iteration cycle is calculated using Equation (2.6-92) along with equilibrium position from the previous cycle, the definition of the Jacobian matrix (Equation 2.6-94 and 2.6-95), and the total forcing function (Equation 2.6-90).

The linear solution of the nonlinear equations of equilibrium is calculated during the first cycle of the iteration scheme, because the initial guess in the iteration is taken by setting all of the nonlinear effects equal to zero.

The nonlinear static equilibrium positions are used to calculate the element forces, the hub shear forces and moments, and the stresses within the blade.

#### 2.6.6.2 Structural Dynamic Analysis

The mode shapes and natural frequencies of vibration of the rotating blade are calculated by linearizing the nonlinear equations of motion. This is accomplished by assuming that the vibration is a small linear perturbation,  $\{\Delta q\}$ , about the nonlinear static equilibrium position, and neglecting products of the perturbation quantities (i.e.,  $\Delta q_i \Delta q_j = 0$ ).

$$\left[ M \right] \{\dot{\Delta q}\} + \left[ M^C \right] \{\Delta \dot{q}\} + \left[ \left[ K^T \right] + \left[ J^{NL}\{q\} \right] \right] \{\Delta q\} = \{0\} \quad (2.6-96)$$

Since the effects of structural damping have been neglected, the above equation will reduce to a standard eigenvalue problem based on a previous work (Ref. 2.6-30).

The linearized equations of motion can be written as:

$$\left[ M \right] \{\dot{\Delta q}\} + \left[ \left[ K^T \right] + \left[ J^{NL}\{q\} \right] \right] \{\Delta q\} = \{0\} \quad (2.6-97)$$

Equations (2.6-97) are solved by standard eigenvalue techniques. The calculated eigenvalues are the squares of the natural frequencies of vibration and the corresponding eigenvectors are equal to the mode shapes.

### 2.6.7 Determination of the Shear Center and Structural Constants of a Cross Section of Arbitrary Shape

The static and dynamic behavior of advanced turbo-propellers is influenced by the coordinates of the shear center, the area and mass center, as well as the bending and shear stiffness. However, the location of the shear center cannot be easily found, except for simple cross sections (i.e., single or multicelled thin walled closed sections) because it entails determining the shear stress distribution (St. Venant bending function) within the cross section. The exact solution for the St. Venant bending function of an arbitrary shaped cross section would be highly intractable based on three-dimensional elasticity theory. In this section, a model for determining the shear center location and the structural properties of a cross section will be developed based on an approximate solution procedure (i.e., finite element method) using the general three-dimensional elasticity theory.

The determination of the cross section warping distribution, shear center location, and other section constants is determined by the following steps.

- (1) The total potential energy of a tip-loaded cantilever beam with a user-defined cross section is derived.
- (2) The out-of-plane warping (shear, torsion) of the cross section is determined based upon applying the principle of minimum total potential energy to a discretized representation of the cross section (finite element representation).
- (3) This warping distribution coupled with the generalized displacement of the beam are used to determine the stress distribution within the tip cross section.
- (4) The cross section equations of equilibrium are applied to tip cross section in order to locate the shear center.

- (5) The remaining section constants (2.6-48, 2.6-49) are calculated by numerically integrating the calculated warping distribution over the cross section definition.

#### 2.6.7.1 Basic Assumptions Used in the Analysis

The following assumptions are used in the development of the model.

- (1) The uniform cantilever beam is defined having an arbitrary cross section shape.
- (2) The coordinate system of the beam is defined with the origin located at the area center of the beam root. The z axis coincides with the area centroid of the beam. The x and y axes align with the principal axes of the cross section (Figure 2.6-9).
- (3) The beam is subjected to two applied tip loads,  $V_x$  and  $V_y$ , which act along the x and y axes, respectively. The twist moment acts about the z axis at the tip of the beam. The effects of body forces are neglected.
- (4) The stresses,  $\sigma_{xx}$ ,  $\sigma_{yy}$ , and  $\tau_{xy}$ , are neglected based on the use of St. Venant assumptions.
- (5) The stresses along the contour of the cylindrical surface of the beam surface are equal to zero (i.e., stress boundary conditions).

#### 2.6.7.2 Total Potential Energy of a Tip Loaded Cantilever Beam

The total potential energy,  $\Pi$ , of a tip-loaded cantilever beam is derived based on the theory of elasticity (Refs. 2.6-31 and 2.6-32). The strain energy,  $U$ , of the beam is derived first using the stress components of the cantilever beam. The work of the external forces,  $W_e$ , is calculated based on the assumption that the tip deflections are the average deflections of the cross section due to the applied loads.

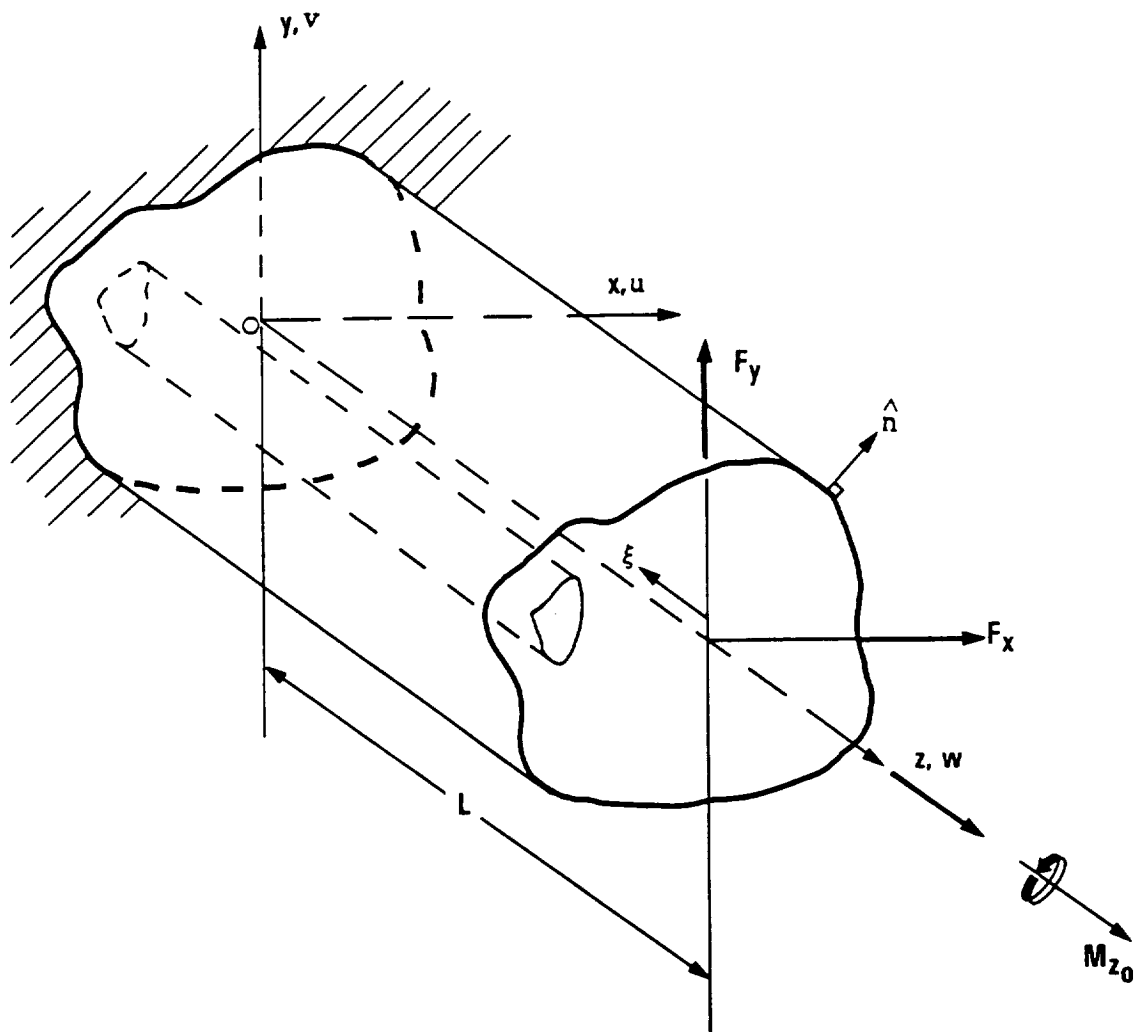


Figure 2.6-9. Cantilever Beam

The normal stress,  $\sigma_{zz}$ , is written based on the above assumptions:

$$\sigma_{zz} = -\frac{x V_x}{I_{yy}}(L - z) - \frac{y V_y}{I_{xx}}(L - z) \quad (2.6-98)$$

where  $I_{xx}$  and  $I_{yy}$  are moments of inertia about the principal axes of the cross section,  $x$  and  $y$ .

The solution for the two remaining shear stresses ( $\tau_{xz}, \tau_{yz}$ ) are determined using the equations of equilibrium and the stress boundary conditions (for a detailed development, Refs. 2.6-31 and 2.6-32):

$$\tau_{xz} = G\theta \left\{ \psi_{,x} - y \right\} - \frac{V_x}{2(1+\nu)I_{yy}} \left\{ \chi_{,x} + \frac{\nu}{2}(x^2 - y^2) \right\} - \frac{V_y}{2(1+\nu)I_{xx}} \left\{ \chi_{,x} + \nu xy \right\} \quad (2.6-99a)$$

$$\tau_{yz} = G\theta \left\{ \psi_{,y} + x \right\} - \frac{V_y}{2(1+\nu)I_{xx}} \left\{ \chi_{,y} + \frac{\nu}{2}(y^2 - x^2) \right\} - \frac{V_x}{2(1+\nu)I_{yy}} \left\{ \chi_{,y} + \nu xy \right\} \quad (2.6-99b)$$

where  $\theta$  is the torsional twist of the cross section,  $\psi$  is the St. Venant torsion warping function, and  $\chi$  is the St. Venant transverse shear warping function.

The displacement state of the beam is determined by integrating the stresses (Equations 2.6-98 and 2.6-99) and by making use of the stress boundary conditions, Hooke's law, and the strain displacement relations (Refs. 2.6-31 and 2.6-32). The boundary conditions, at the beam root, are defined by setting the average values of the displacements and rotations over the whole cross section equal to zero.

$$u(x, y, z) = \frac{V_x}{2EI_{yy}} \left\{ \nu \{x^2 - y^2\} (L - z) + \frac{z^2}{3} (3L - z) \right\} + \frac{V_y}{EI_{xx}} \left\{ \nu xy (L - z) \right\} - \theta y z$$



$$\begin{aligned}
& + \frac{V X}{2EI_{yy}} \frac{Y L \nu}{(I_{xx} + I_{yy})} \iint_A x^2 y + y^3 dA - \frac{V Y}{2EI_{xx}} \frac{X L \nu}{(I_{xx} + I_{yy})} \iint_A y^2 x + x^3 dA \\
& + \frac{Z V X}{2EI_{yy} I_{yy}} \iint_A x \chi(x, y) dA + \frac{Z V Y}{2EI_{xx} I_{yy}} \iint_A x \chi(x, y) dA \\
& + \frac{V X}{EI_{yy}} \left\{ \frac{L \nu}{2A} \left\{ I_{xx} - I_{yy} \right\} + \frac{Z \theta}{I_{yy}} \right\} \iint_A x \psi(x, y) dx dy \quad (2.6-100a)
\end{aligned}$$

$$\begin{aligned}
v(x, y, z) & = \frac{V Y}{2EI_{xx}} \left\{ \nu \left\{ y^2 - x^2 \right\} (L - z) + \frac{Z}{3} (3L - z) \right\} + \frac{V X}{EI_{yy}} \left\{ \nu \chi y (L - z) \right\} + \theta x z \\
& - \frac{V X}{2EI_{yy}} \frac{X L \nu}{(I_{xx} + I_{yy})} \iint_A x^2 y + y^3 dA + \frac{V Y}{2EI_{xx}} \frac{X L \nu}{(I_{xx} + I_{yy})} \iint_A y^2 x + x^3 dA \\
& + \frac{Z V X}{2EI_{yy} I_{xx}} \iint_A y \chi(x, y) dA + \frac{Z V Y}{2EI_{xx} I_{xx}} \iint_A y \chi(x, y) dA \\
& + \frac{V Y}{EI_{xx}} \left\{ \frac{L \nu}{2A} \left\{ I_{yy} - I_{xx} \right\} + \frac{Z \theta}{I_{xx}} \right\} \iint_A y \psi(x, y) dx dy \quad (2.6-100b)
\end{aligned}$$

$$\begin{aligned}
w(x, y, z) & = \frac{V X}{2EI_{yy}} \left\{ \chi(x, y) - xz(2L - z) \right\} + \frac{V Y}{2EI_{xx}} \left\{ \chi(x, y) + yz(2L - z) \right\} + \theta \psi(x, y) \\
& - \frac{X V X}{2EI_{yy} I_{yy}} \iint_A x \chi(x, y) dA - \frac{X V Y}{2EI_{xx} I_{yy}} \iint_A x \chi(x, y) dA - \frac{X \theta}{I_{yy}} \iint_A x \psi(x, y) dA
\end{aligned}$$

$$\begin{aligned}
& - \frac{y V_x}{2EI_{yy}I_{xx}} \iint_A y \chi(x,y) dA - \frac{y V_y}{2EI_{xx}I_{yy}} \iint_A y \chi(x,y) dA - \frac{y \theta}{I_{xx}} \iint_A y \psi(x,y) dA \\
& - \frac{1}{A} \left\{ \frac{V_x}{2EI_{yy}} + \frac{V_y}{2EI_{xx}} \right\} \iint_A \chi(x,y) dA - \frac{\theta}{A} \iint_A \psi(x,y) dA \quad (2.6-100c)
\end{aligned}$$

The strain energy,  $U$ , is defined as:

$$U = \frac{1}{2} \int_0^L \iint_A (\sigma_{xx}\epsilon_{xx} + \sigma_{yy}\epsilon_{yy} + \sigma_{zz}\epsilon_{zz} + \tau_{yz}\gamma_{yz} + \tau_{xz}\gamma_{xz} + \tau_{xy}\gamma_{xy}) dA dz$$

The strain energy is rewritten by substituting in the stresses and carrying out the integration over the beam length:

$$U = \frac{1}{2} \iint_A \left\{ \left\{ \frac{x V_x}{EI_{yy}} + \frac{y V_y}{EI_{xx}} \right\} \frac{L^3}{3} + \frac{L}{G} \{ \tau_{yz}^2 \} + \frac{L}{G} \{ \tau_{xz}^2 \} \right\} dA \quad (2.6-101)$$

where the shear stresses,  $\tau_{yz}$  and  $\tau_{xz}$ , are defined in Equation (2.6-99).

The work of external forces,  $W_e$ , of the cantilever beam subjected to tip shear forces,  $V_x$  and  $V_y$ , and a tip twist moment,  $M_t$  is given by:

$$W_e = V_x \cdot u_1 + V_y \cdot v_1 + M_t \cdot \bar{\theta} \quad (2.6-102)$$

where  $u_1$  and  $v_1$  are the average tip deflections calculated from Equations (2.6-100a) and (2.6-100b), respectively.  $\bar{\theta}$  is the average tip rotation about the  $z$  axis. The tip displacements are given by:

$$\begin{aligned}
u_1 = & \frac{L V_x}{2EI_{yy}I_{yy}} \iint_A x \chi(x,y) dA + \frac{L V_y}{2EI_{xx}I_{yy}} \iint_A x \chi(x,y) dA + \frac{L \theta}{I_{yy}} \iint_A x \psi(x,y) dA \\
& + \frac{V_x}{EI_{yy}} \left\{ \frac{L^2}{2A} \{ I_{xx} - I_{yy} \} + \frac{L^3}{3} \right\} \quad (2.6-103a)
\end{aligned}$$

$$v_1 = \frac{L V_x}{2EI_{yy}I_{xx}} \iint_A y \chi(x,y) dA + \frac{L V_y}{2EI_{xx}I_{yy}} \iint_A y \chi(x,y) dA + \frac{L \theta}{I_{xx}} \iint_A y \psi(x,y) dA + \frac{V_y}{EI_{xx}} \left\{ \frac{Lv}{2A} (I_{yy} - I_{xx}) + \frac{L^3}{3} \right\} \quad (2.6-103b)$$

$$\bar{\theta} = \theta - \frac{V_x}{2EI_{yy}} \frac{Lv}{(I_{xx}+I_{yy})} \iint_A x^2y + y^3 dA + \frac{V_y}{2EI_{xx}} \frac{Lv}{(I_{xx}+I_{yy})} \iint_A y^2x + x^3 dA \quad (2.6-103c)$$

The total potential energy,  $\Pi$ , of the cantilever beam, which is the difference between the strain energy and the work of external forces, is expressed as:

$$\Pi = \frac{1}{2} \iint_A \left\{ \left\{ \frac{x V_x}{EI_{yy}} + \frac{y V_y}{EI_{xx}} \right\} \frac{L^3}{3} + \frac{L}{G} \{ \tau_{yz}^2 \} + \frac{L}{G} \{ \tau_{xz}^2 \} \right\} dA - V_x \cdot u_1 - V_y \cdot v_1 - M_t \cdot \bar{\theta} \quad (2.6-104)$$

where the shear stresses,  $\tau_{xz}$  and  $\tau_{yz}$ , are defined in Equation (2.6-98) and the average displacements,  $u_1$ ,  $v_1$ , and  $\theta$ , are defined in Equation (2.6-103).

### 2.6.7.3 Solution of the St. Venant Functions by the Finite Element Method

A closed form solution of the St. Venant bending function,  $\chi$ , and the torsion function,  $\psi$ , can only be determined for a few simple cross sections (i.e., circle, ellipse, rectangle, etc.) because of the difficulty in solving the two-dimensional elasticity problem. The finite element method has been shown to be a reliable method for the solution of two-dimensional elasticity problems (i.e., plane stress, plane strain) because it can model structures of arbitrary shape.

The finite element method is a piecewise application of the variational principle. The solution procedure is not based on solving the governing equations of equilibrium, but the minimization of the total potential energy of the beam. In order to minimize the total potential energy, the first variation of the total potential with respect to the dependent variables (i.e.,  $\chi(x,y)$  and  $\psi(x,y)$ ) must equal zero. The solution will satisfy both the differential equations of equilibrium and the natural boundary conditions.

This method is applied by discretizing the cross section into subregions (finite elements) and taking the local variation of each region. A three node triangular element, with a two-dimensional linear polynomial, has been used by other authors for the solution of two-dimensional elasticity problems (Refs. 2.6-33 and 2.6-34). This linear interpolation function will lead to a fully conforming element that monotonically converges to exact solution when the mesh is refined.

A two dimensional linear serendipity interpolation polynomial is used for both the St. Venant functions and the coordinate definitions. Writing in matrix form:

$$\left. \begin{aligned} \chi(x,y)^m &= \{N(\zeta_1, \zeta_2)\} \{\chi\}^m & x^m &= \{N(\zeta_1, \zeta_2)\} \{X\}^m \\ \psi(x,y)^m &= \{N(\zeta_1, \zeta_2)\} \{\psi\}^m & y^m &= \{N(\zeta_1, \zeta_2)\} \{Y\}^m \end{aligned} \right\} \quad (2.6-105)$$

where the linear interpolation function for the  $m^{\text{th}}$  element is defined as  $N(\zeta_1, \zeta_2)$  and  $\zeta_1$  and  $\zeta_2$  are the generalized coordinates.

Writing the variation of the St. Venant functions in matrix form:

$$\delta\chi(x,y)^m = \{N(\zeta_1, \zeta_2)\} \{\delta\chi\}^m \quad \delta\psi^m = \{N(\zeta_1, \zeta_2)\} \{\delta\psi\}^m \quad (2.6-106)$$

A set of linear equations is obtained by substituting the functions (Equation 2.6-105) and the variations of the functions (Equation 2.6-106), into the variation of the element total potential energy and carrying out the integration over the element area. Standard finite element procedures

are used to assemble the finite elements into a global set of linear equations. The solution of these equations is the distribution of the St. Venant functions over the cross section.

$$[K] \begin{Bmatrix} \chi \\ \psi \end{Bmatrix} = \{F\} \quad (2.6-107)$$

#### 2.6.7.4 Location of the Shear Center

The location of the shear center is determined using the moment equilibrium equation and the shear stress distribution within the tip cross section. The applied tip load is assumed to act through the shear center of the cross section so that the beam will not twist (Figure 2.6-10). From the definition of the stresses (Equation 2.6-98 and 2.6-99), it can be seen that the St. Venant torsion function,  $\psi$ , will be uncoupled from the bending problem.

#### X Coordinate of the Shear Center

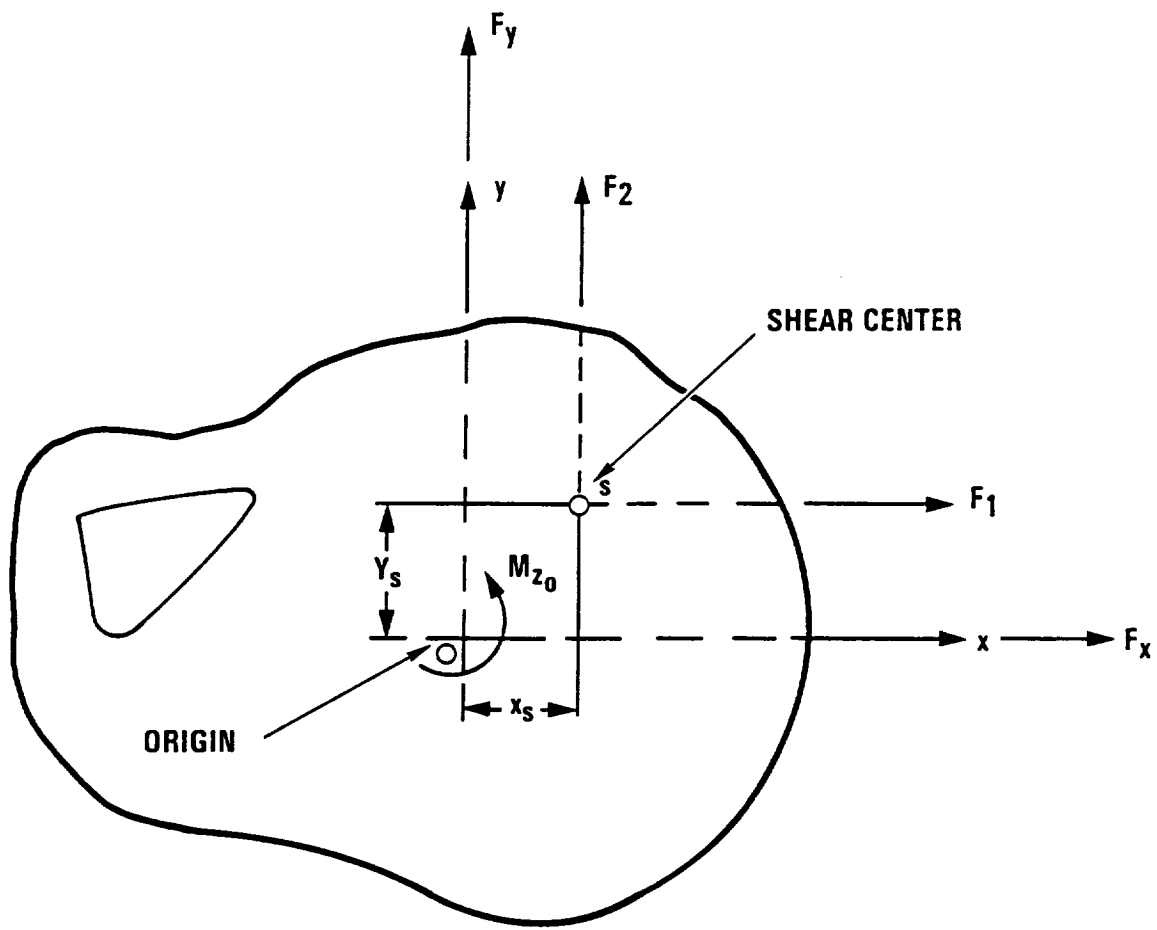
In order to determine the x coordinate of the shear center, the St. Venant functions must be calculated for the case of an applied load acting through the shear center and parallel to the y axis,  $V_y$ .

The variation of the total potential energy,  $\delta\Pi$ , for this load condition is defined as:

$$\delta\Pi = \frac{L}{G} \left\{ \frac{V_y}{2(1+\nu)I_{xx}} \right\}^2 \iint_A \left\{ \delta\chi_{,x} \left\{ \chi_{,x} + \nu xy \right\} + \delta\chi_{,y} \left\{ \chi_{,y} + \frac{\nu}{2}(y^2 - x^2) \right\} - \delta\chi_y (1+\nu) \right\} dA \quad (2.6-108)$$

The element matrices are derived by substituting the interpolation functions (Equations 2.6-105 and 2.6-106) into the variation and carrying out the integration over the element area. Writing the matrix relations for the  $m^{\text{th}}$  finite element by setting the above variation equal to zero:

$$[K^m] \{\chi^m\} = \{F^m\} \quad (2.6-109)$$



$$M_{z_0} = x_s \cdot F_2 - y_s F_1$$

Figure 2.6-10. Shear Center Location on the Beam Cross Section

where:

$$[K^m] = \iint_{A^m} N(s_1, s_2)_{,x} N(s_1, s_2)_{,x} + N(s_1, s_2)_{,y} N(s_1, s_2)_{,y} dA^m \quad (2.6-110a)$$

$$\{F^m\} = \iint_{A^m} -N(s_1, s_2)_{,x} \{vxy\} - N(s_1, s_2)_{,y} \left\{ \frac{v}{2}(y^2 - x^2) \right\} + N(s_1, s_2) \{y(1 + v)\} dA^m \quad (2.6-110b)$$

The solution of Equation (2.6-109) is the St. Venant bending function for a load applied through the shear center in the y direction. The shear stress distribution of the cross section is determined by substituting the calculated function into Equation (2.6-99). The shear center offset in the x direction is calculated by carrying out the moment equilibrium equation for the tip cross section using the shear stress distribution.

$$x_s = \frac{1}{V_y} \sum_{m=1}^n \iint_{A^m} \{x \tau_{yz}^m - y \tau_{xz}^m\} dA^m \quad (2.6-111)$$

### Y Coordinate of the Shear Center

The y coordinate of the shear center is determined using the same procedure as above except a tip load is applied through the shear center and parallel to the x axis. The element matrices will have the form:

$$[K^m] = \iint_{A^m} N(s_1, s_2)_{,x} N(s_1, s_2)_{,x} + N(s_1, s_2)_{,y} N(s_1, s_2)_{,y} dA^m \quad (2.6-112a)$$

$$\{F^m\} = \iint_{A^m} -N(s_1, s_2)_{,y} \{vxy\} - N(s_1, s_2)_{,x} \left\{ \frac{v}{2}(x^2 - y^2) \right\} + N(s_1, s_2) \{x(1 + v)\} dA^m \quad (2.6-112b)$$

Once the St. Venant distribution is known for an applied load through the shear center in the x direction, then the stress distribution over the cross section can be calculated. The shear center offset in the y direction is calculated by carrying out the moment equilibrium equation for the tip cross section:

$$y_s = \frac{-1}{V_y} \sum_{m=1}^n \iint_{A^m} \{x \tau_{yz}^m - y \tau_{xz}^m\} dA^m \quad (2.6-113)$$

#### 2.6.7.5 Determination of the Torsional Constant and the Warping Constants

The torsional constant, J, and the warping constants are determined using the St. Venant torsion function and the finite element method. The distribution of this function is calculated by applying a tip torsion load to the cantilever beam.

The variation of the total potential energy with respect to the St. Venant torsion function for a cantilever beam with an applied tip moment is:

$$\delta\Pi = L G \theta^2 \iint_A \delta\psi_{,x} \{\psi_{,x} - y\} + \delta\psi_{,y} \{\psi_{,y} + x\} dA \quad (2.6-114)$$

The element matrices are derived by substituting the interpolation functions (Equations 2.6-105 and 2.6-106) into the variation and carrying out the integration over the element area. Writing the matrix relations for the m<sup>th</sup> finite element by setting the above variation equal to zero:

$$[K^m] \{\psi^m\} = \{F^m\} \quad (2.6-115)$$

where:

$$[K^m] = \iint_{A^m} N(s_1, s_2)_{,x} N(s_1, s_2)_{,x} + N(s_1, s_2)_{,y} N(s_1, s_2)_{,y} dA^m \quad (2.6-116a)$$



$$\{F^m\} = \iint_{A^m} y N(\zeta_1, \zeta_2)_{,x} - x N(\zeta_1, \zeta_2)_{,y} dA^m \quad (2.6-116b)$$

Once the St. Venant torsion function is known, the torsion constant,  $J$ , is directly calculated based on its definition.

$$J = \sum_{m=1}^n \iint_{A^m} \{x^2 + y^2 + x \psi_{,y}^m - y \psi_{,x}^m\} dA^m \quad (2.6-117)$$

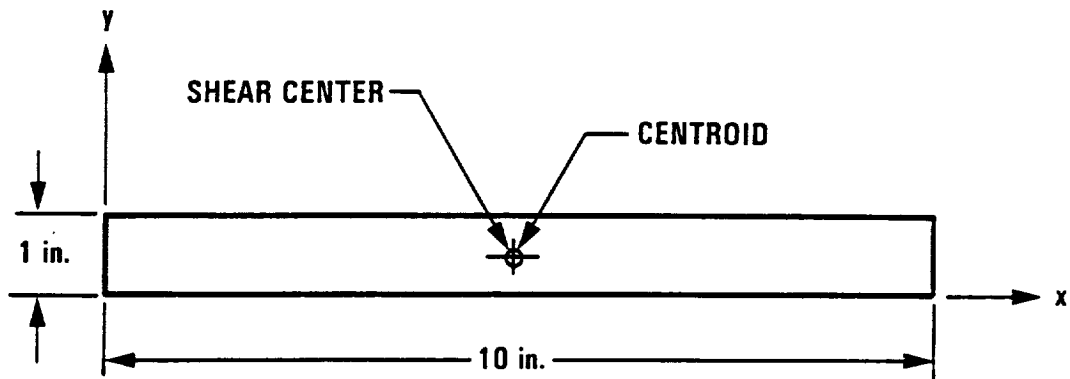
The warping constants, which are required for the beam analysis, are calculated using the St. Venant torsion warping function distribution,  $\psi(x,y)$ , and by expressing the cross section integrals as a summation of the integrals over each finite element (i.e., similar to Equation 2.6-117).

#### 2.6.7.6 Sample Calculations

In order to demonstrate the versatility and accuracy of the approximate analysis, five example problems are presented. Two problems for which known exact solutions are available were selected so that a comparison could be made. Three problems of cross sections from an SR-3 turbo-propeller are also selected.

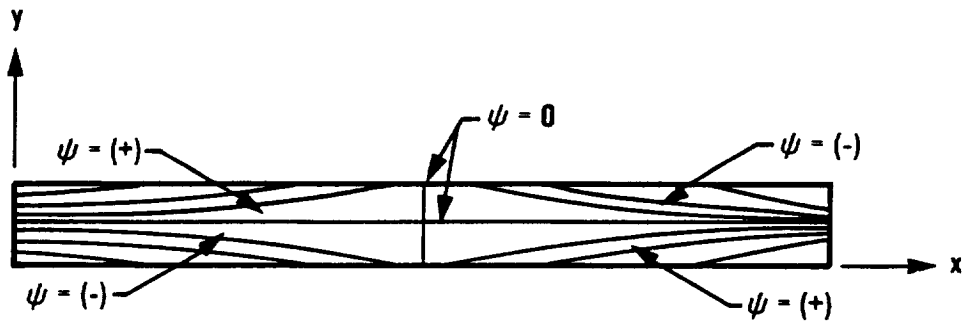
The results of an analysis of a rectangular cross section show excellent agreement between the exact solution and the approximate solution (Figure 2.6-11) using 64 finite elements. The calculated warping constants were negligible compared to the other constants of the cross section, and could be neglected. The calculation of the shear center location for an L-section (Figure 2.6-12) shows excellent agreement for a relatively few number of elements (36 elements).

An analysis of cross sections from an SR-3 propeller was also done. Three of the cross sections will be discussed here. The first section was located at a point roughly 25% from the blade root (Figure 2.6-13). From Figure 2.6-13a, it can be seen that there is a significant offset between the shear center, mass center, and the chord line. The warping constants, which were considered negligible in previous beam developments (Ref. 2.6-35 and 2.6-36), are of significant magnitude and cannot be neglected



A) CROSS SECTION DEFINITION

$$J = 3.12 \text{ in.}^4$$



B) WARPING FUNCTION DISTRIBUTION (CONTOUR LINES OF CONSTANT VALUES OF  $\psi$ )

Figure 2.6-11. Analysis of an Isotropic, Rectangular Cross Section

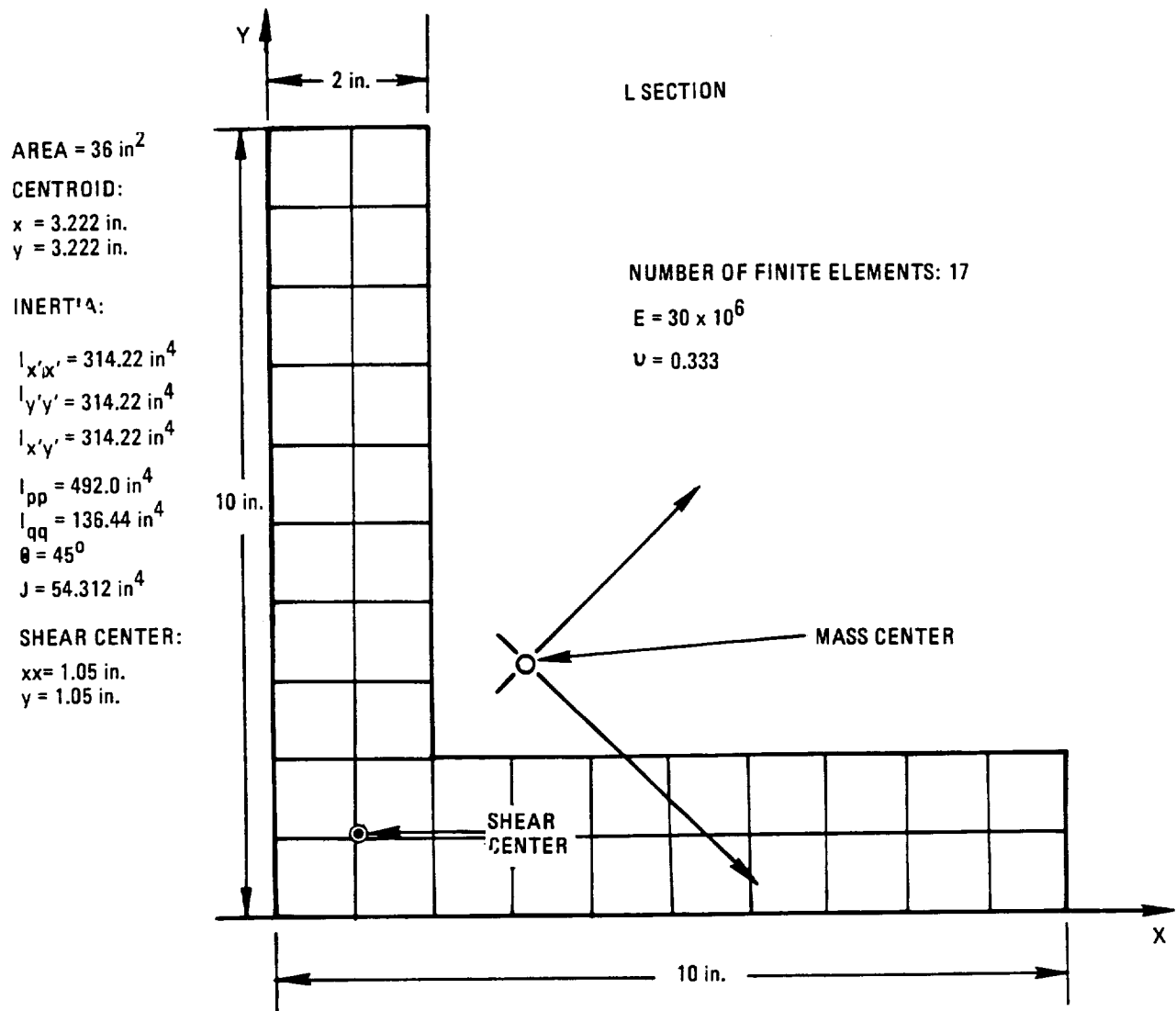


Figure 2.6-12. L-Shaped Cross Section

CHORD: 4.620 in.

CENTROID:  $x = 2.077$  in.  
 $y = 0.04756$  in.

SHEAR CENTER:  $x = 1.938$  in.  
 $y = 0.06923$  in.

AREA:  $0.9319$  in.<sup>2</sup>

MOMENTS OF INTERIA:

$I_{pp} = 0.00491$  in.<sup>4</sup>

$I_{qq} = 1.034$  in.<sup>4</sup>

$\theta_p = 0.1312^\circ$

TORSION CONSTANT,  $J$ :  $0.02053$  in.<sup>4</sup>

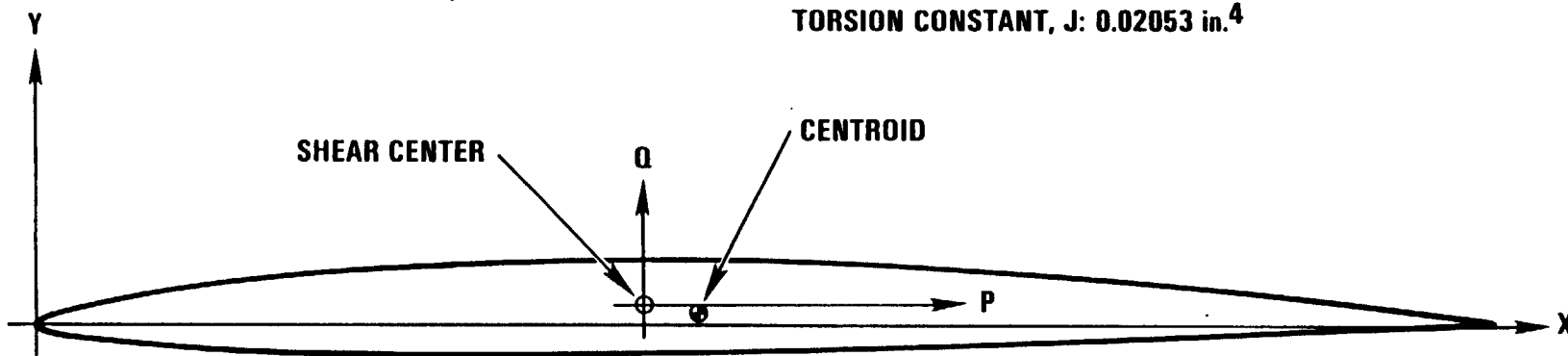


Figure 2.6-13a. Profile of SR-3 Turbo-Propeller at Location 25% from Hub

(Figure 2.6-13b). This is a result of; (1) the camber in the cross section which shifts the St. Venant warping function distribution and the shear center away from the mean chord line, and (2) the antisymmetric nature of the warping function coupling with the asymmetries of the cross section. A cross section from the root of the blade (Figure 2.6-14a) and also from the tip of the blade (Figure 2.6-14b) were also analyzed. From these figures, it can be seen that the use of camber has a tendency to shift the shear center above the camber line and the mass center below the camber line. This is expected based on exact results of shear center calculations for thin arcs. It is also important to note that all of the warping constants are of significant magnitude and should not be neglected.

### 2.6.8 Numerical Results

Numerical results, aimed at illustrating the characteristics and behavior of the nonlinear rotating beam finite element model, will be presented. This includes comparisons of linear and nonlinear, static and dynamic results with known analytical and experimental solutions.

#### Sample Calculations

Part of the initial verification of this model was to test its ability to analyze curved beams. An analysis of a tip-loaded circular cantilever beam was done using the present finite element model, Figure 2.6-15. A series of tests was performed for different radii of curvature and the solutions were compared with known analytical results (Ref. 2.6-37). The beam finite element model consisted of six finite elements that were placed along the curved elastic axis of the beam. Six different tip loads (i.e., three tip forces, three tip moments) were applied separately to the model and the tip displacements were calculated for fixed values of beam curvature. Excellent results (Figures 2.6-16 through 2.6-18) were obtained as the curvature of the beam varied from zero degrees (straight cantilever beam) to 90 degrees (semicircular cantilever beam).

The calculation of the natural frequencies and mode shapes of a curved beam was performed and a comparison was made with known analytical solutions (Ref. 2.6-38). This part of the verification of the model was more difficult because even though there are plenty of results for straight

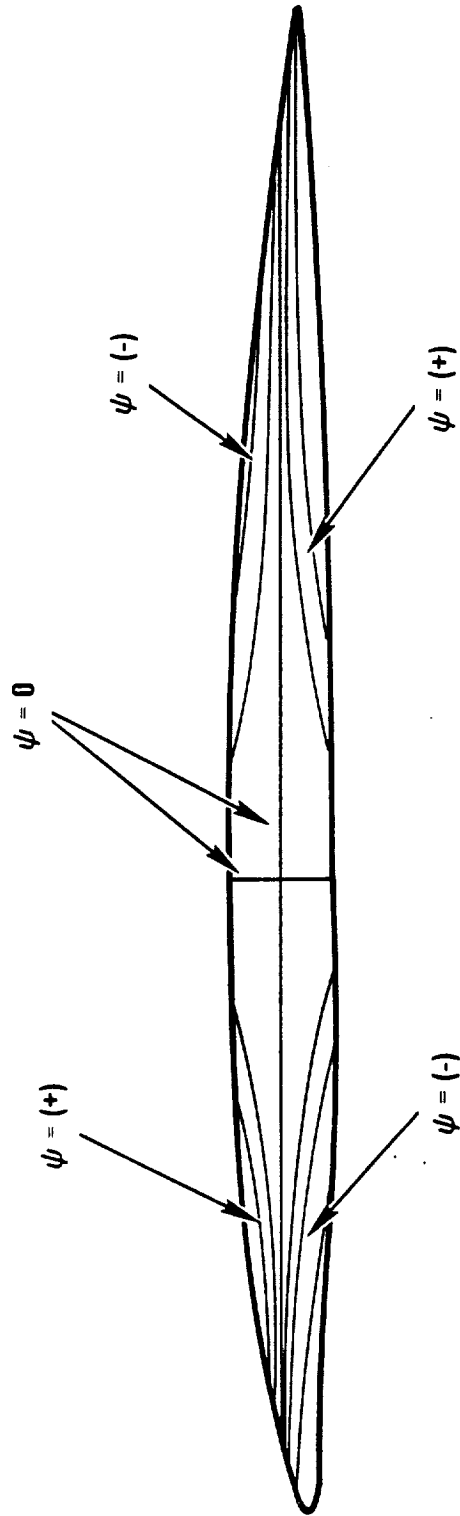


Figure 2.6-13b. Warping Function Distribution of a NASA SR-3  
(25% from Blade Root)

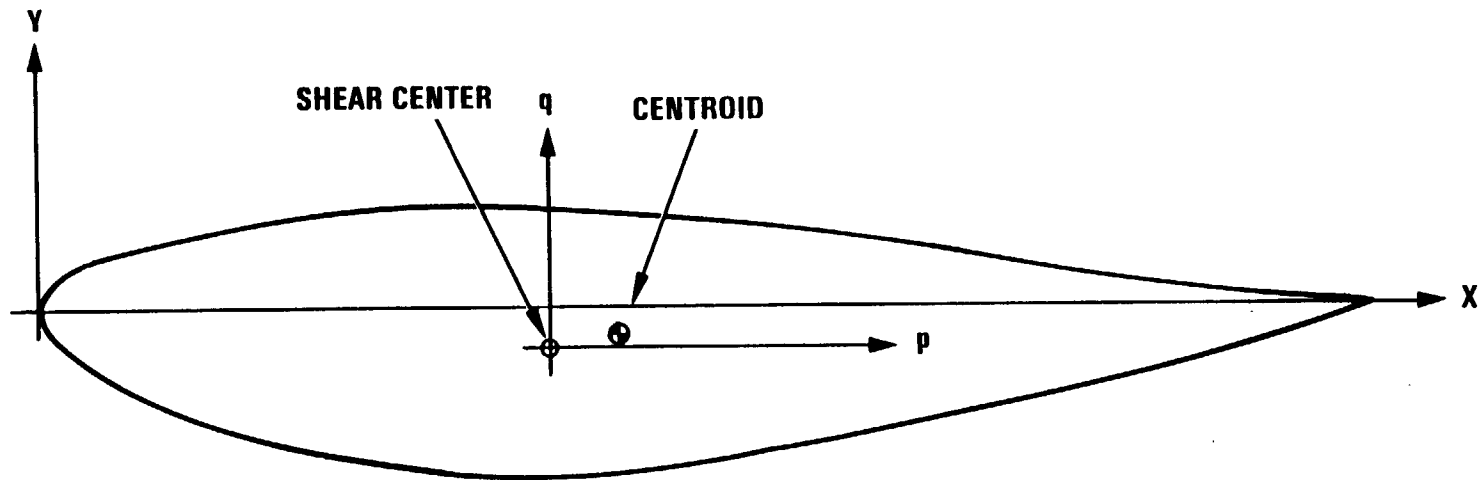
**CHORD: 4.039 in.****AREA: 2.103 in<sup>2</sup>****CENTROID:****X = 1.722 in.****Y = -0.0845 in.****MOMENTS OF INERTIA:****I<sub>pp</sub> = 0.07628 in.<sup>4</sup>****I<sub>qq</sub> = 1.716 in.<sup>4</sup>****SHEAR CENTER:****X = 1.520 in.****Y = -0.1164 in.** **$\theta_p = -0.2058^\circ$** **TORSION CONSTANT, J = 0.2896 in.<sup>4</sup>**

Figure 2.6-14a. Profile of SR-3 Turbo-Propeller at Blade Root

**CHORD: 1.675 in.**

**AREA: 0.05309 in.<sup>2</sup>**

**CENTROID: x = 0.7836 in.**

**MOMENTS OF INTERIA:**

**y = 0.03617 in.**

**$I_{pp} = 1.06 \times 10^{-5} \text{ in.}^4$**

**SHEAR CENTER: x = 0.7196 in.**

**$I_{qq} = 8.206 \times 10^{-3} \text{ in.}^4$**

**y = 0.05071 in.**

**$\theta_p = 0.1165^\circ$**

**TORSION CONSTANT, J:  $3.926 \times 10^{-5} \text{ in.}^4$**

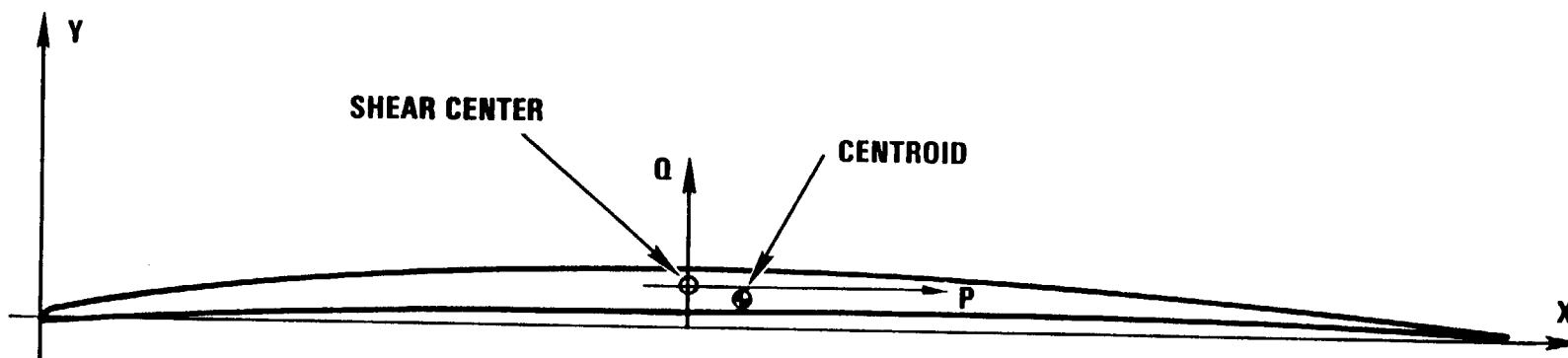
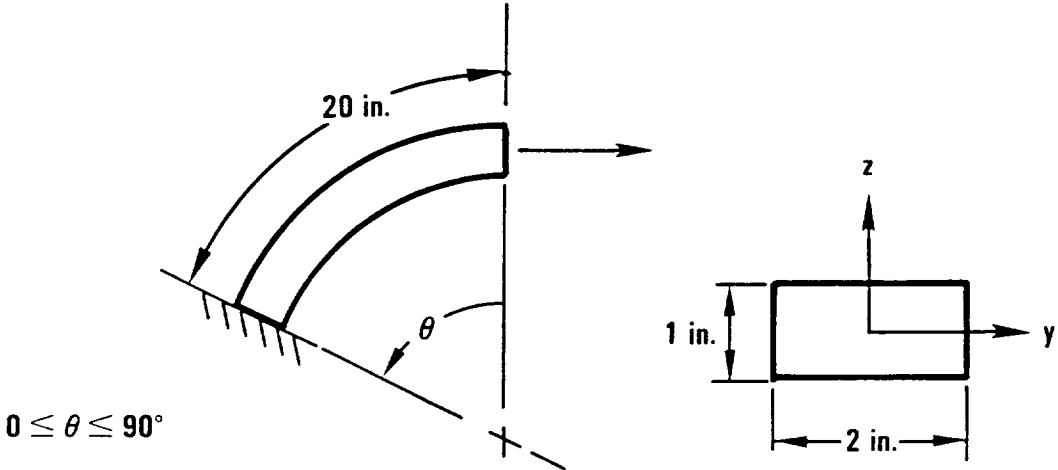


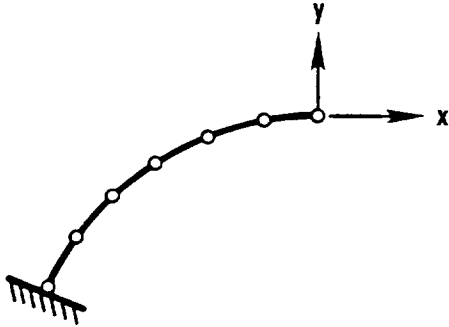
Figure 2.6-14b. Profile of SR-3 Turbo-Propeller at Blade Tip



MATERIAL: ALUMINUM - 6061 T6



A) DESCRIPTION OF CURVED BEAM



B) FINITE ELEMENT REPRESENTATION (6 ELEMENTS)

Figure 2.6-15. Tip-Loaded Analysis of Curved Cantilever Beam

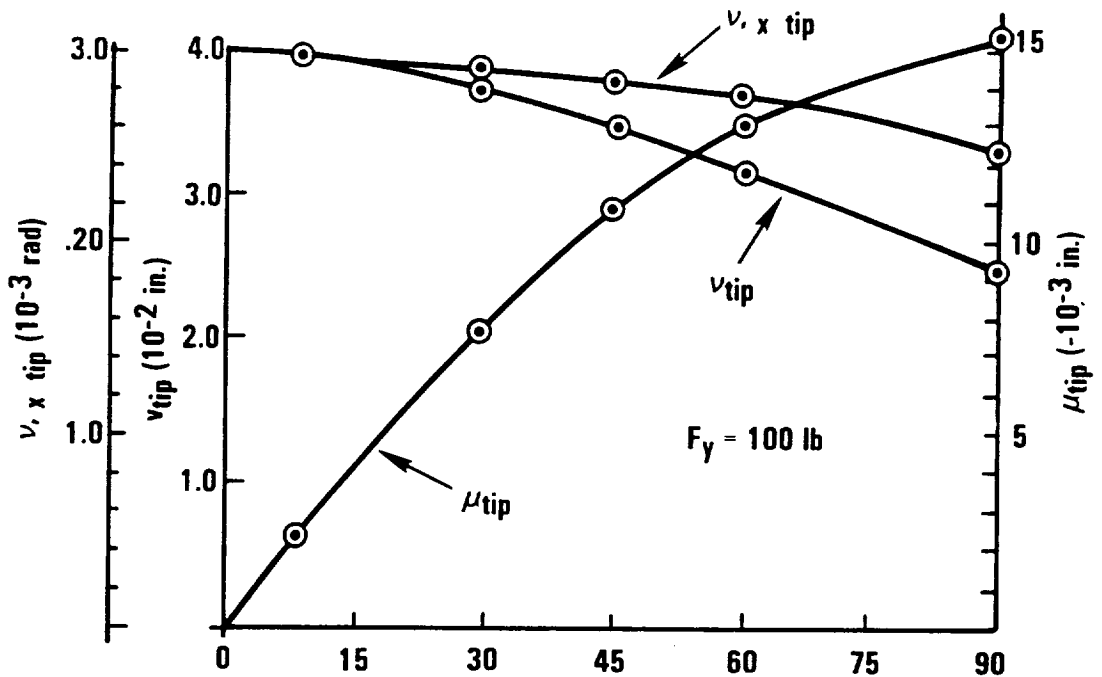
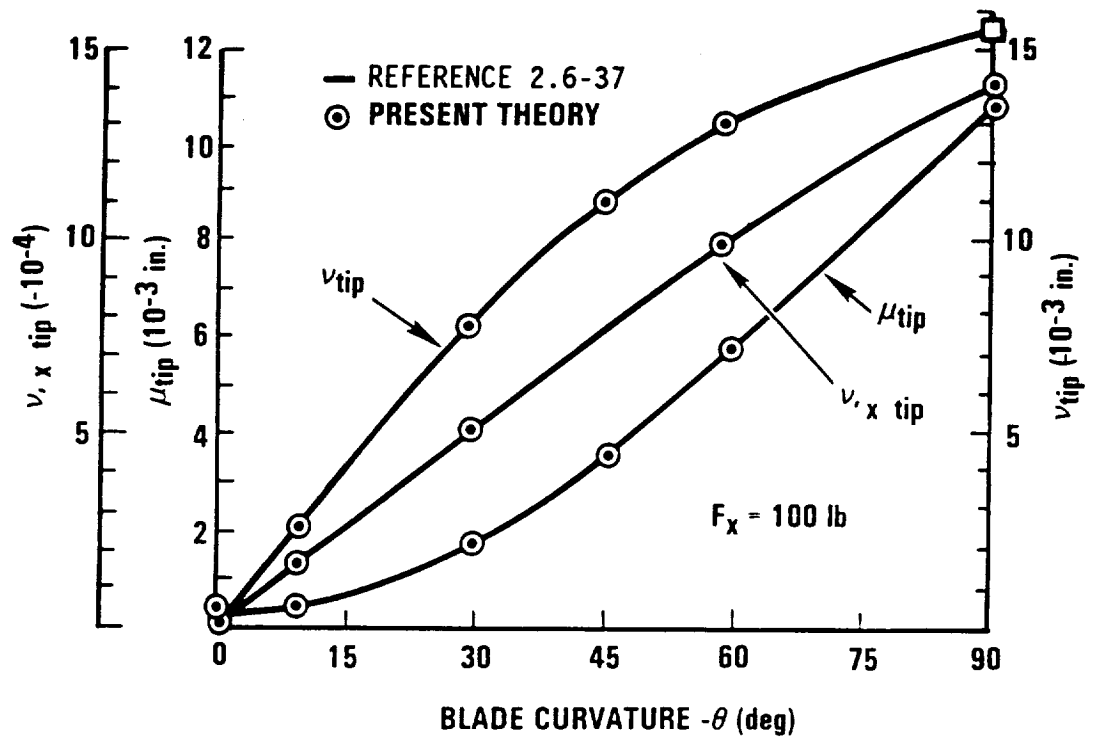


Figure 2.6-16. Tip Displacements for Tip-Loaded Curved Cantilever Beam. Applied axial force (top) and in-plane transverse shear force (bottom).

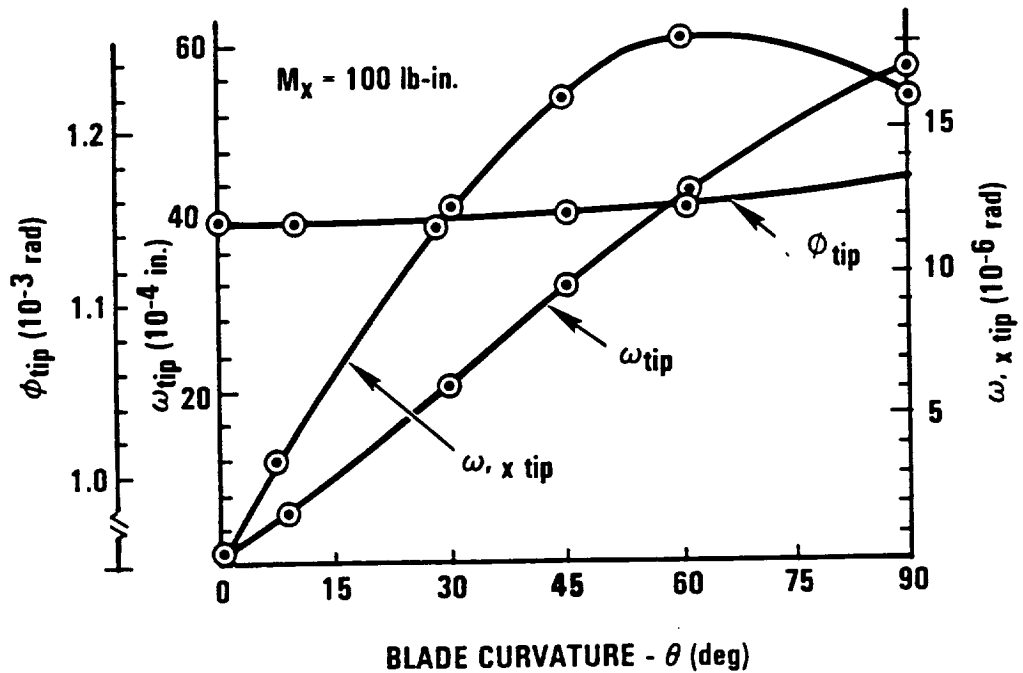
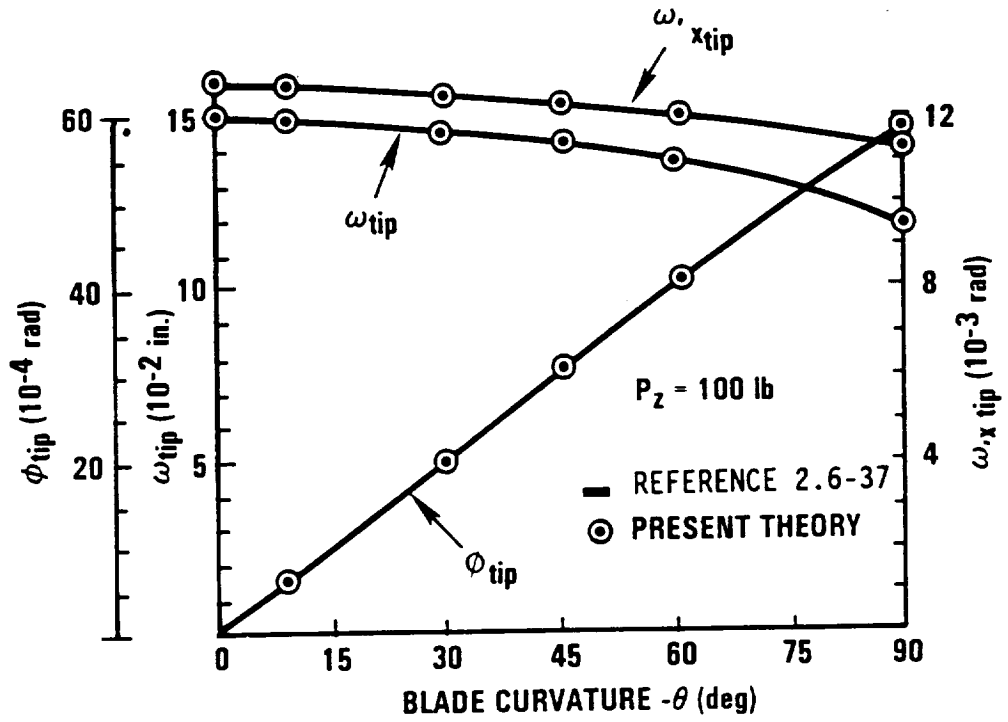


Figure 2.6-17. Tip Displacements for Tip-Loaded Curved Cantilever Beam. Applied out-of-plane transverse force (top) and torsion moment (bottom).

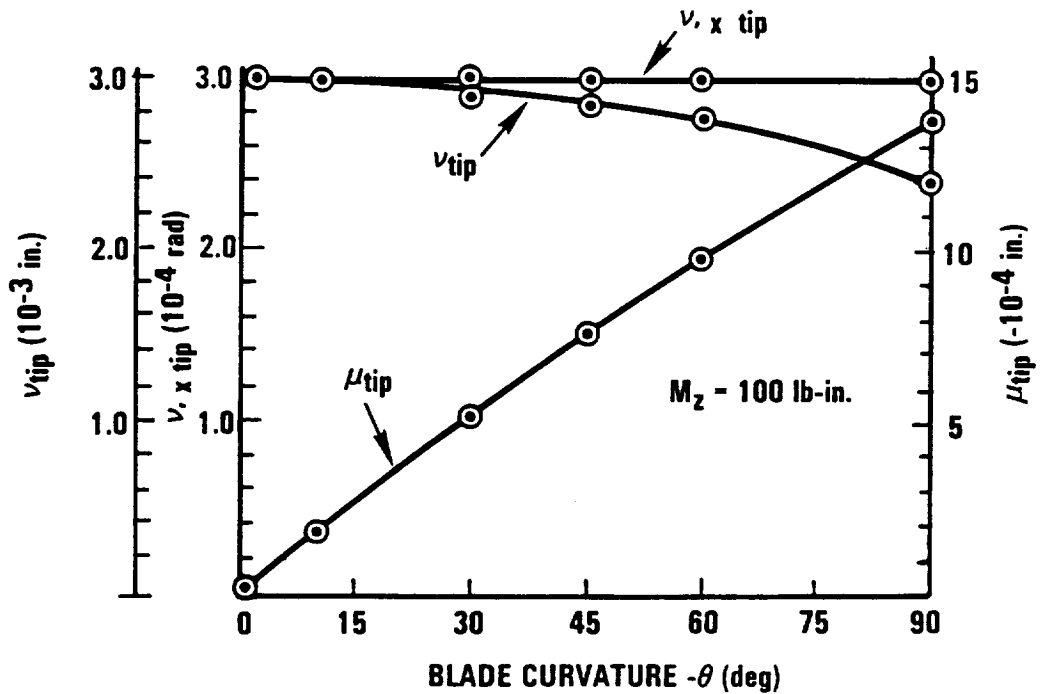
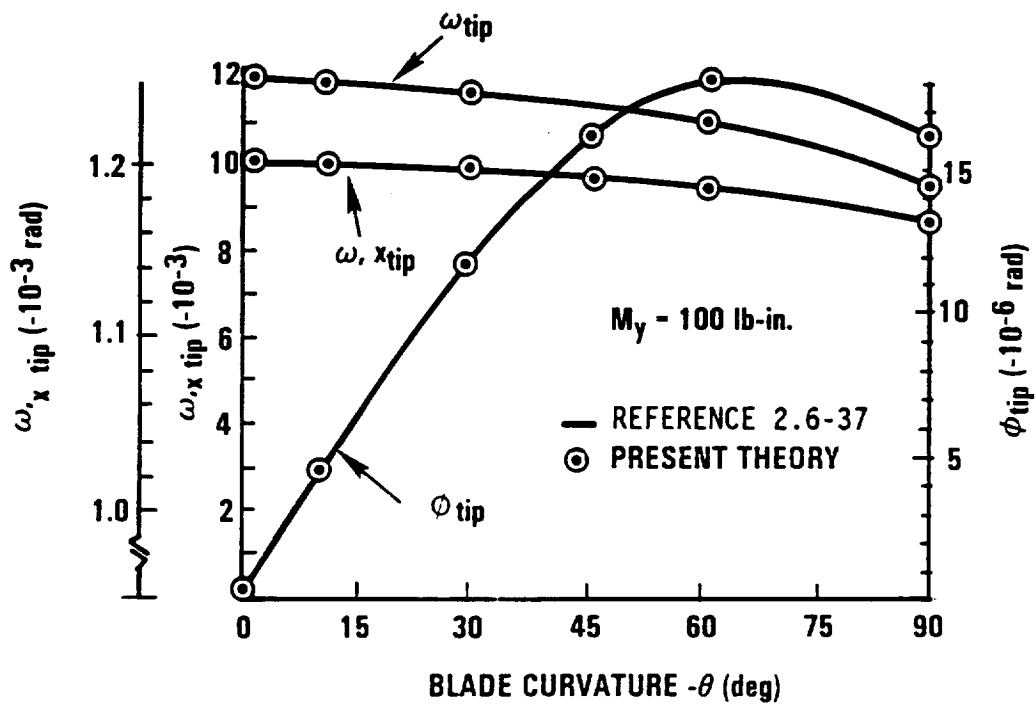


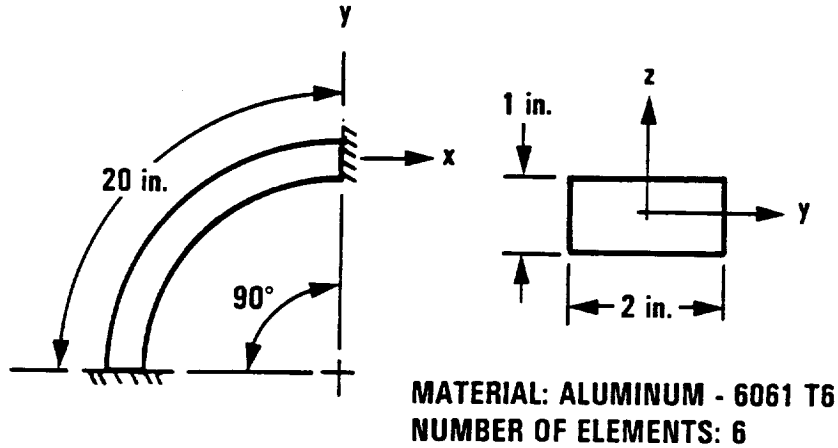
Figure 2.6-18. Tip Displacements for Tip-Loaded Curved Cantilever Beam. Applied out-of-plane bending moment (top) and in-plane bending moment (bottom).

beams, there are very few closed form results that exist for curved beams. This is because the mode shapes for curved beams are geometrically coupled together and their definition is usually described in either two- or three-dimensions.

The natural frequencies and mode shapes for a straight cantilever beam were calculated, using six finite elements, and a comparison was made with analytical results (Ref. 2.6-38, p. 108). Excellent agreement was obtained for the first 11 mode shapes and frequencies, errors of less than 0.5% were found between the analytical solution and the present model.

The natural frequencies and mode shapes of a clamped-clamped semi-circular arc were calculated and the solutions were compared with analytical results (Ref. 2.6-38, p. 207-208), see Figure 2.6-19. Six finite elements were placed along the curved elastic axis of the beam and the first eight eigenvalues and eigenvectors were determined. These mode shapes were compared with the analytical solutions for the in-plane flexural modes and the first out-of-plane flexure mode. From Figure 2.6-19, it can be observed that a good correlation between the finite element model and the analytical solution was obtained for very few finite elements. A comparison of the torsion modes and the higher order out-of-plane flexural modes could not be done because an analytical solution could not be found.

A study was also done to see how the natural frequencies and mode shapes of a cantilever beam would vary as in-plane, out-of-plane, and pretwisted curvatures were introduced. The definition of the geometry of the beam and the results are presented in Figures 2.6-20 through 2.6-22. A series of analyses were performed by starting with a straight cantilever beam and increasing the in-plane curvature (Figure 2.6-20). The results of this study show that introducing in-plane curvature will cause a slight reduction in the natural frequencies of the out-of-plane modes (z direction) over the complete range of curvatures. This can be accounted for by noting that as curvature is introduced into the beam, the beam is actually becoming shorter (in the x direction) and the mass effects are no longer evenly distributed, but are concentrated closer to the tip. This shortening of the beam and shifting of the mass center of the beam will

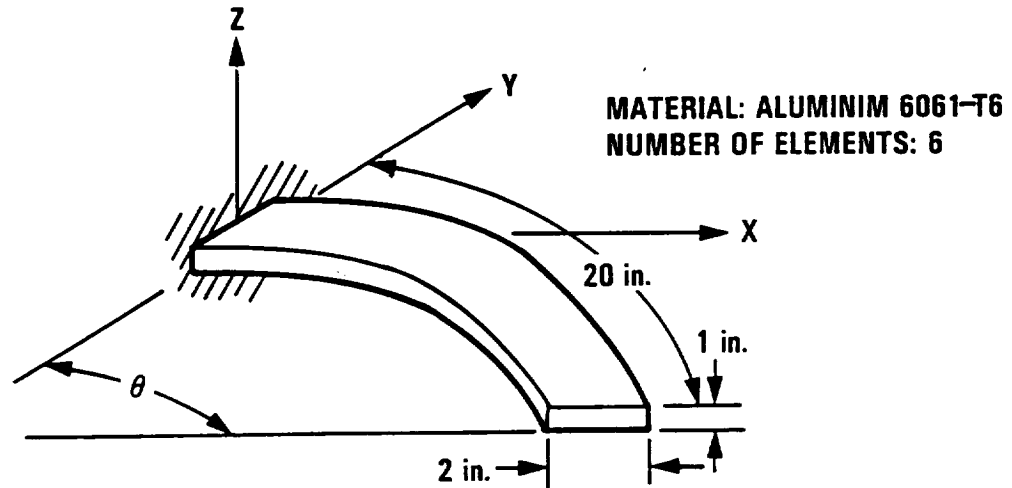


A) DESCRIPTION OF CURVED BEAM

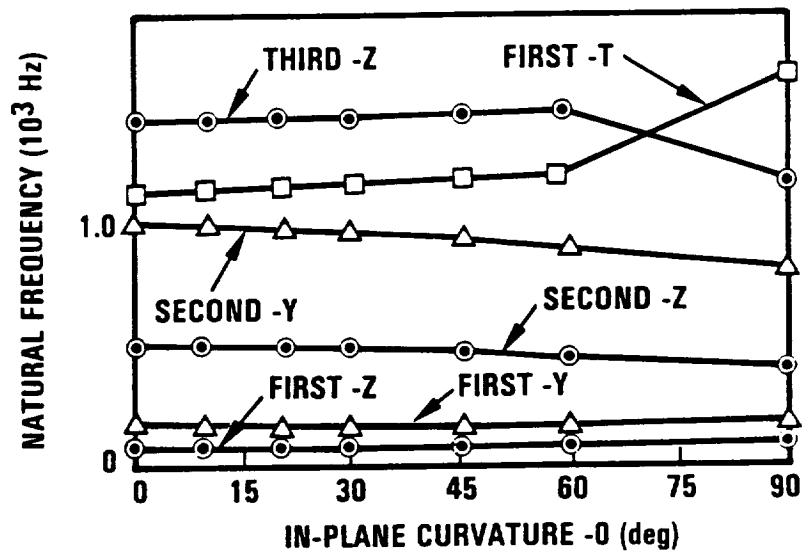
MODE DESCRIPTION	PRESENT THEORY EIGENVALUE (Hz)	REFERENCE 2.6-38 EIGENVALUE (Hz)	ERROR (%)
FIRST OUT-OF-PLANE FLEXURE	472.21	473.2	0.21
SECOND OUT-OF-PLANE FLEXURE	1343.8		
FIRST IN-PLANE FLEXURE (HOOP)	2237.26	2277.5	1.8
SECOND IN-PLANE FLEXURE	2494.7	2542.5	1.9
FIRST TORSION	2578.3		
THIRD OUT-OF-PLANE FLEXURE	2716.2		
FOURTH OUT-OF-PLANE FLEXURE	4537.6		
SECOND TORSION	4953.6		

B) COMPARISON OF PRESENT THEORY WITH PREVIOUS CLOSED FORM SOLUTIONS  
REFERENCE 2.6-38

Figure 2.6-19. Natural Frequency Calculation of a Clamped-Clamped Circular Arc



A) DESCRIPTION OF CANTILEVER BEAM WITH IN-PLANE CURVATURE



B) NATURAL FREQUENCY OF CANTILEVER BEAM

Figure 2.6-20. Natural Frequency Calculation of a Cantilever Beam with In-Plane Curvature

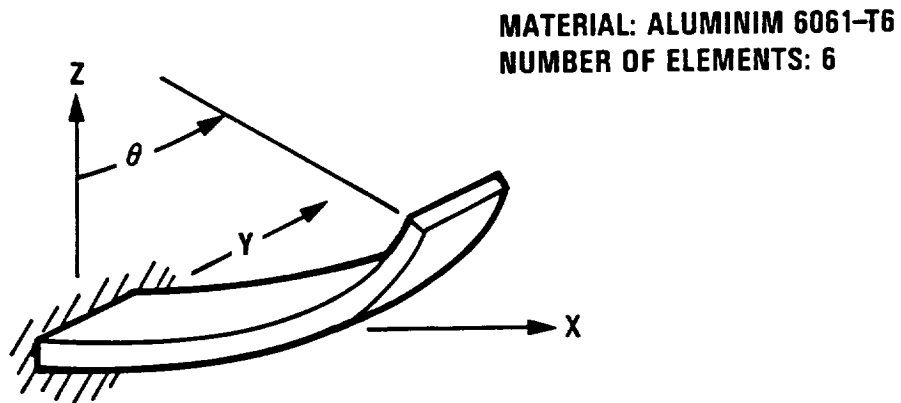
lower the natural frequencies of the out-of-plane modes. A sharp increase in the first torsion natural frequency and a sharp decrease in the third z natural frequency in the region of high in-plane curvatures can be accounted for due to the geometric coupling.

The introduction of out-of-plane curvature into the cantilever beam model will produce effects that are very similar to the effects of in-plane curvature, Figure 2.6-21. Out-of-plane curvature will generally lower both the in-plane and out-of-plane natural frequencies of the beam because of the geometric shortening and shifting of the mass center that is taking place. The natural frequency for the first torsion mode makes a sharp increase as out-of-plane curvature is added to this model, due to the geometric coupling between the torsion mode and the in-plane motion.

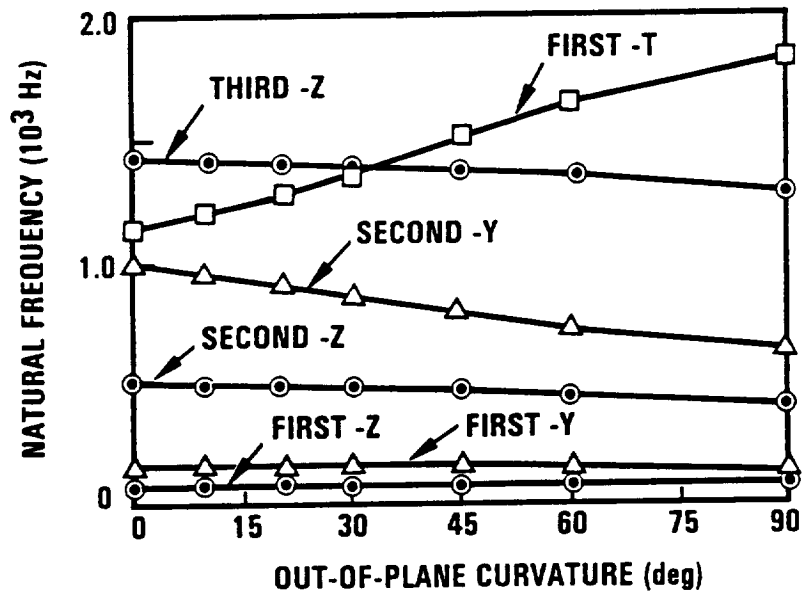
The incorporation of a uniform distribution of pretwist along the beam length will cause the in-plane and out-of-plane natural frequencies to coalesce, Figure 2.6-22. This can be accounted for due to the geometric coupling that is introduced between the in-plane and out-of-plane degrees of freedom. The first torsion natural frequency also decreases due to the incorporation of beam pretwist that reduces the torsional stiffness of the model.

The nonlinear terms associated with the moderate deflection theory were verified using experimental results that were obtained by loading a straight cantilever beam in the moderate deflection regime, Figure 2.6-23. The results of the calculations using this finite element model were identical to the results produced by Rosen and Friedmann (Refs. 2.6-3 and 2.6-39). This was expected because when the effects of beam pretwist and curvature are neglected, the equations developed by the present theory will reduce to the equations of Rosen and Friedmann (Ref. 2.6-3). One of the figures from their paper has been included to show how this theory compares with the other theories. From Figure 2.6-23b, it can be observed that the present theory follows the experimental results very closely in nonlinear range, whereas the linear theory falls away and theory developed by Hodges and Dowell (Ref. 2.6-2) appears to be unstable. Further discussion of the comparison between the different theories can be found in Reference 2.6-39.





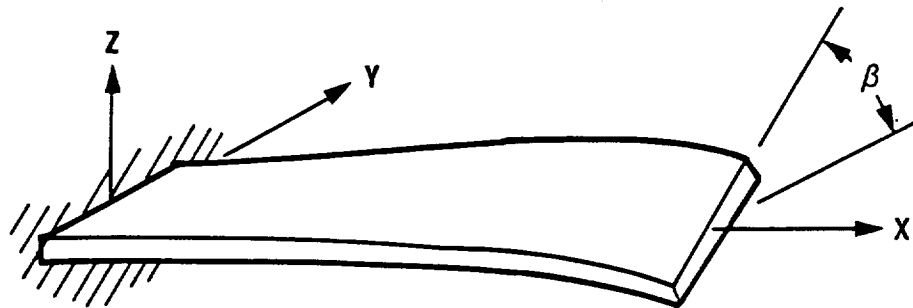
**A) DESCRIPTION OF CANTILEVER BEAM WITH OUT-OF-PLANE CURVATURE**



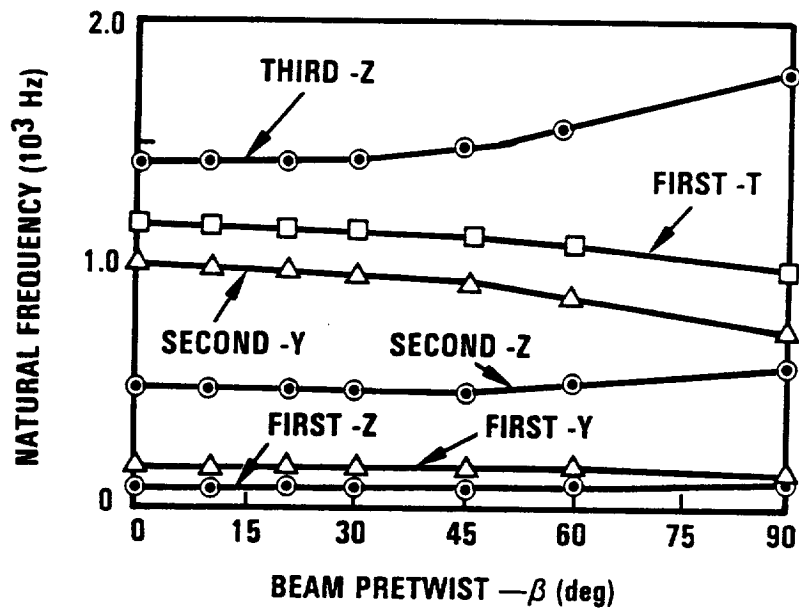
**B) NATURAL FREQUENCIES OF CANTILEVER BEAM**

Figure 2.6-21. Natural Frequency Calculation of a Cantilever Beam with Out-of-Plane Curvature

MATERIAL: ALUMINIM 6061-T6  
 NUMBER OF ELEMENTS: 6

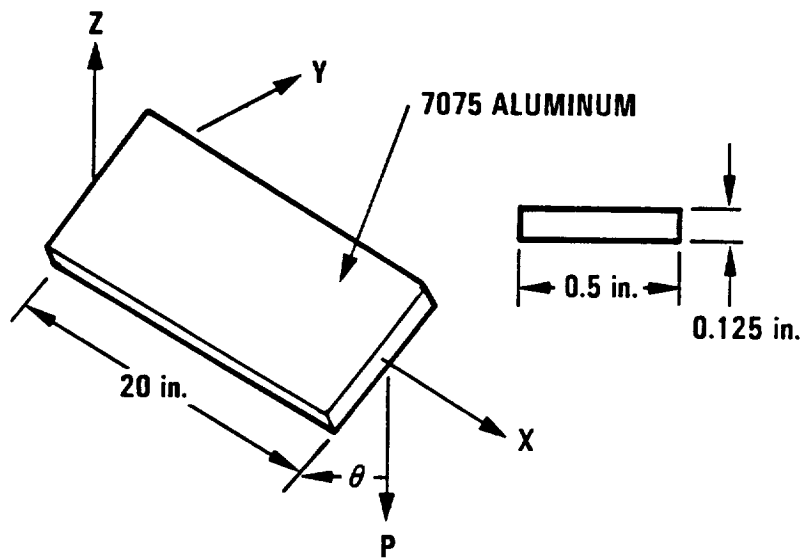


A) DESCRIPTION OF PRETWISTED CANTILEVER BEAM

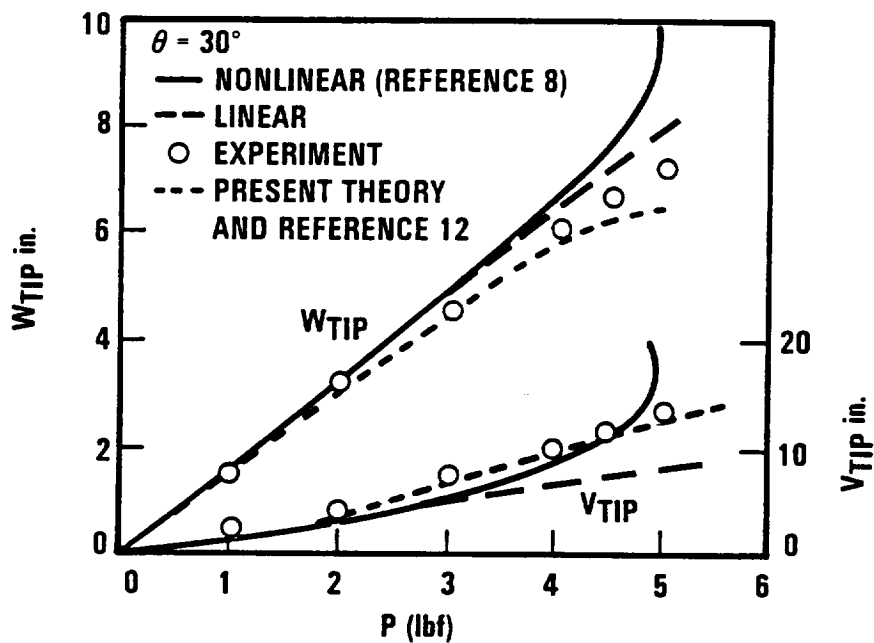


B) NATURAL FREQUENCIES OF CANTILEVER BEAM

Figure 2.6-22. Natural Frequency Calculation of Pretwisted Cantilever Beam



A) DESCRIPTION OF CANTILEVER BEAM



B) TIP DEFLECTIONS OF THE BEAM

Figure 2.6-23. Uniform Cantilevered Beam Under Transverse Loading

### 2.6.9 General Description of Program Structure

The nonlinear beam structural dynamic analysis program uses a conventional structure found in most finite element programs. There is an initial subroutine "main" which is used to allocate blank common storage and pass the blank common to "smain" (sub-main). The subroutine "smain" is the driver for the entire program. Space for all of the input data, generated element matrices, global matrices, and solution arrays is allocated in "smain" using both the blank common and out-of-core storage. The linear element matrices (nxn) are generated only once and written to out-of-core storage so that they can be used for each iteration of the nonlinear solution. The global matrices are all generated in skew form (#dof,bandwidth) and written to out-of-core storage. An out-of-core equation solver and an out-of-core eigenvalue package are used to calculate the nonlinear static equilibrium position and the eigenvalues, respectively. The out-of-core eigenvalue solver package was developed by Dr. Gupta of NASA-Dryden and is available through COSMIC.

Many of the subroutines that are included in this version of the code are not being used if a beam-type analysis is being done. These subroutines are going to be used for doing a nonlinear structural dynamic analysis of rotating advanced composite propellers with plate-type finite elements. The plate-type routines that are in the code now, include all of the propeller coordinate generation routines, the plate material properties routines (including composite cross sections with isotropic and orthotropic materials), the plate element routines, the input plate pressure load routines, boundary condition routines, and a bandwidth minimizer. The incorporation of a bandwidth minimizer is an absolute necessity when designing a two-dimensional finite element program because of the limited computer storage that can not be wasted by inefficient node numbering. A flowchart of the program is presented in Figure 2.6-24.

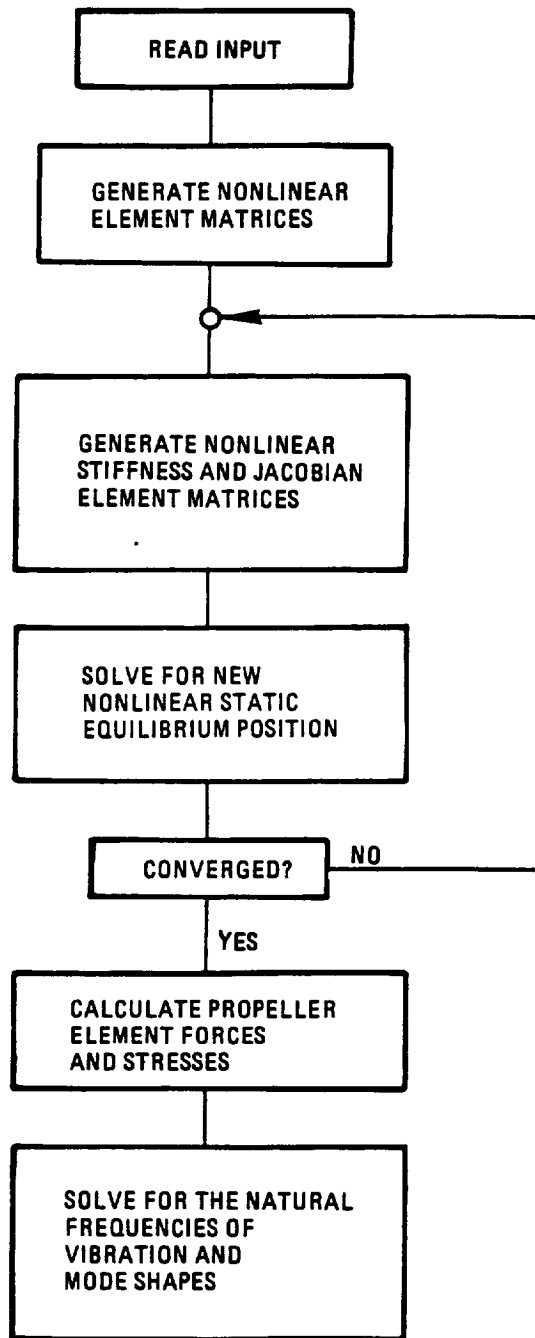


Figure 2.6-24. Flowchart of Nonlinear Structural Dynamic Analysis



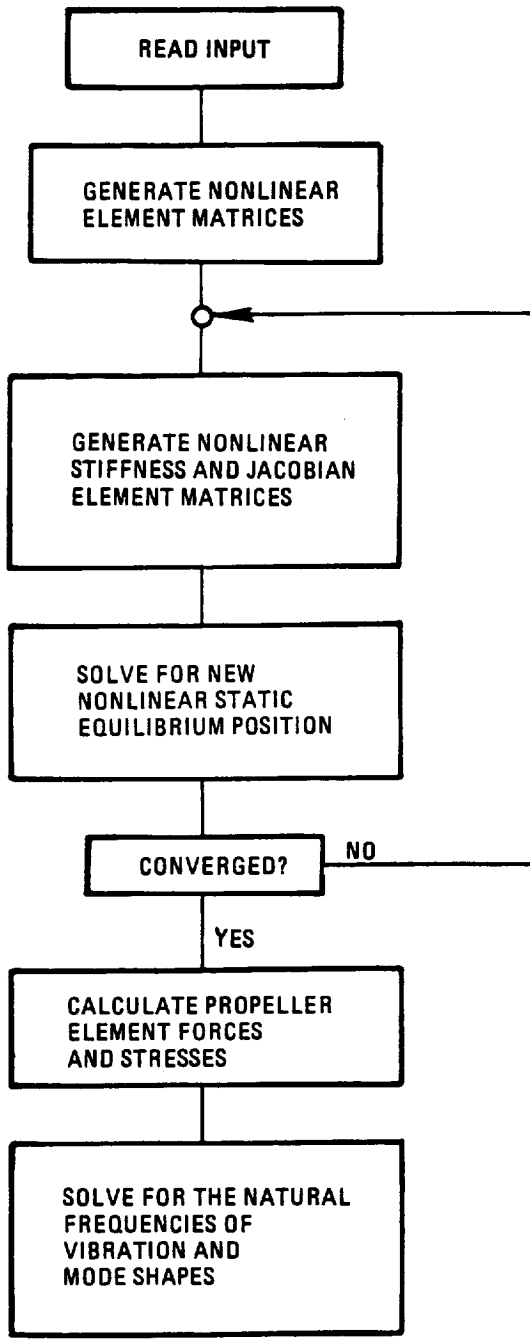


Figure 2.6-24. Flowchart of Nonlinear Structural Dynamic Analysis





### 3. GAPAS SYSTEM

This section describes terms and definitions necessary for a user to use the Generalized Advanced Propeller Analysis System. It explains how the GAPAS executive system is designed, how data handling requirements are implemented, and how the user controls the program flow. Once the user becomes familiar with the basic concepts of the system, he can refer to subsequent sections for the control statements and functional modules needed to solve his problem.

#### 3.1 SYSTEM DESIGN

The functional design requirements for GAPAS are defined as:

- (1) Flexibility for addition, replacement, or removal of analysis modules. This conforms to the modularity requirement necessary for the maintenance of large software.
- (2) User control for the selective and effective use of the various methodologies. Either single pass or multiple pass, or logical bypass of redundant calculations can be implemented with ease.

GAPAS meets these requirements by separating the overall system into two distinct parts; the executive and functional program segments. All executive system control tasks are performed in executive modules; and all performance analysis is contained in the functional modules. Such a modular structure of GAPAS provides the desired flexibility for adding and replacing functional modules.

The GAPAS executive system, which is derived from the NASA Langley Aircraft Noise Prediction Program (ANOPP), allows the user to execute either the single-pass or multiple-pass modes. To do that, the user prepares a set of GAPAS input control statements which determine the processing flow during a run. Using various control statements, the user selects the analysis modules, and accesses or creates data units in the data base required for the execution of each module.

The executive system provides for all interfacing with the host computer's operating system. It initializes the program, edits the control statements, processes each control statement by calling the executive and/or functional module into execution, and performs error handling. It

also performs data base management within the central memory. It manages the global dynamic storage (GDS), a permanent area established within the program field length, where data structures and internal tables are maintained. It also manages the local dynamic storage (LDS), where intramodule data transfer may be accomplished.

A functional module is a logically independent group of subprograms that perform a specific analysis function, for example, the airfoil loading or structural module. A functional module is loaded into central memory for execution when the executive system processes a user control statement requesting its execution. Upon completion, other executive or functional modules may be loaded to process, depending on subsequent control statements.

The functional module internally satisfies its input data requirements by referencing the named data unit and data member, or user parameters maintained in the data base. Prior to the control statement that requests the execution of a particular functional module, the user includes other control statements (for example, UPDATE, LOAD) which create the required named data units. The data units may have been created in a previously executed functional module in the same run, and therefore already exist in the run environment. In that case, no further control statements are required to establish such named data units. For user parameter, control statement PARAM can be used. These control statements are described in a separate section.

### 3.2 GAPAS DATA BASE

With the exception of user parameters, all data units within a GAPAS run reside within the data base on random access devices such as disks. The data base provides a method of storing and retrieving data. In addition, through the convention of named data units and members, and the capabilities of the data base manager, the user is relieved of the responsibilities of interfacing with the host operating system when opening, reading, writing, and closing external files. In that, all external files, with the exception of the input and output files, are dynamically created as a GAPAS run proceeds.

A data unit is a named collection of data members. It is equivalent to a random access file in the external host operating system. A data member consists of one or more of logically related and organized records. Each member resides on a named data unit and must possess a name unique to all members within the same unit. Since no two data units may have the same name within a GAPAS run, any combination of data unit name and member name uniquely determines the location of data to be fetched. Therefore a unit (member) combination provides the address to the data of interest.

A member consists of one or more records, each conforming to the structure specified in the member's format specification. Records are arranged in sequential order. The data member supports three general format types:

- (1) Unformatted records: variable-length stream of data with structure defined for the member.
- (2) Card image records: fixed-length (80 characters) alphanumeric records, each with only one element corresponding to a Hollerith card image.
- (3) Formatted records: fixed- and variable-length records with their structure defined by a member format specification of the sequence of data element types for each record.

In the format specification of a member, the data types of the elements within a record are designated by: integer (I); real single precision (RS); real double precision (RD); complex single precision (CS); logical (L); and alphanumeric string of n characters (An). Member format specification rules are discussed in a subsequent section.

User parameters are considered as part of the overall GAPAS data base. They are named data arrays of one or more elements of the same data type. The data types of user parameters are designated as those of data members discussed above. User parameters are central memory words, and as such they can be accessed expediently and conveniently. The values can be modified either through subsequent control statements in the same input stream, or within functional modules using executive system supplied routines.

### 3.3 GAPAS EXECUTIVE SYSTEM CONTROL STATEMENTS

Control statements are the functional directives by which the user controls the flow of a GAPAS run. The statements are all structured in accordance with general specifications, and each provides a specific format that reflects the functional capabilities provided to the user. Specific capabilities provided by control statements include the following:

- (1) Establishing the beginning and end of the input stream (STARTCS and ENDCS)
- (2) Setting the executive system run control parameter (ANOPP)
- (3) Selecting print options for summary information or content of data units, members, and user parameters
- (4) Establishing and re-establishing user parameter values
- (5) Establishing conditional processing thereby controlling the flow of execution stream (IF GOTO)
- (6) Assigning data units to the run via creation, attachment, or loading techniques (UPDATE,LOAD)
- (7) Terminating a data unit's current assignment
- (8) Creating a data unit, data table and data member by using existing data currently present in a run
- (9) Executing functional modules
- (10) Establishing the checkpoint and restart capabilities within a run.

#### 3.3.1 Control Statement Format

Each executive control statement (CS) has a specific format which adheres to the conventions described below.

The CS directive is a free-form sequence of fields on card images, using columns 1 through 80. A field is defined as a name or constant required for the directive, separated from other fields by a delimiter, for example a comma. The field may start from any column as long as the sequence of fields specified for a directive is correct. A CS directive is

terminated by a "\$" sign. The directive may continue on as many as five cards if required, and the end-of-data sign "\$" only occurs on the last card.

A CS field may not be split across two cards; however, the following exceptions are allowed: a complex single precision type field may specify one of the two single real constants on one card, and have the second real constant entirely on the following card; a string type field, where the character value may be split across two cards.

The general format of a control statement is as follows:

LABEL,CSNAME,OPERANDS \$ COMMENTS

where:

LABEL - The label is an optional field specified by one to eight characters. It is normally used only in CS which provides conditional branching via "GOTO label" control statement. A label is allowed on any CS except ANOPP, RSTRT, and STARTCS statements.

"," - The specification of "," on a CS format indicates the requirement for a field delimiter. The delimiter may be one or more commas or space.

CSNAME - This is the name of a control statement identified with one to eight characters. The GAPAS executive system supports the following control statements:

ANOPP, ARCHIVE, ATTACH, CALL, CATALOG, CKPNT, CONTINUE CREATE,  
DATA, DETACH, DROP, ENDCS, EVALUATE, EXECUTE, GOTO, IF, LOAD,  
MEMLIST, PARAM, PROCEED, PURGE, RSTRT, SETSYS, STARTCS, TABLE,  
TABLIST, TITLE, UNLOAD, UPDATE, UPLIST

OPERAND - These are the operand fields required for each control statement.

In the representation of control statements the following additional conventions are used:

- (1) Optional fields are always shown enclosed within a set of [ ] brackets.
- (2) A field specification where the user has a choice of two or more operand types, one of which must be selected, is always shown within a set of { } brackets.

- (3) Special delimiter characters are shown where required on control statement format specifications. These special delimiters are (A), =, /, nd ;.

### 3.3.2 Description of GAPAS Control Statements

The following section describes the syntax and applications of various control statements supported by GAPAS. This section is taken directly from the ANOPP reference manual and is included in the GAPAS for completeness. Therefore any references to ANOPP in the text, for example "ANOPP," "ANOPP run" etc., should be regarded as meaning "GAPAS," "GAPAS run," etc.

#### 3.3.2.1 ANOPP Control Statement

**PURPOSE:** The ANOPP CS allows the user to assign values to specified executive initialization parameters and system parameters during the GAPAS run.

**Format:** ANOPP param<sub>1</sub>=value<sub>1</sub> [..... param<sub>n</sub>=value<sub>n</sub>] \$

**param** the name of an executive system parameter or initialization parameter, subject to change via the ANOPP CS, whose value the user wishes to replace (see Table 3.3-1).

**value** the value, of the correct data type, which is to replace the default value for the specified parameter

#### Examples:

```
ANOPP JECHO=.TRUE. $
ANOPP JLOG=.FALSE. LENG=3500 $
ANOPP NLPPM=45, NOGO=.TRUE., NAEUD=6 $
```

#### Restrictions:

The ANOPP CS is valid only as the first CS in the user's run input deck.

The label field is illegal on the ANOPP control statement.

Assigned values must be of the correct data type and within the range prescribed for the system or initialization parameter.

Table 3.3-1. ANOPP CS Initialization and System Parameters

Parameter		Description	Value		Default value
Name	Type		Type	Range	
ACCOUNT	System <sup>a</sup>	Controls accumulation of cost-accounting data associated with each functional module or executive processor  ACCOUNT=.TRUE. - accumulate accounting data ACCOUNT=.FALSE. - do not accumulate accounting data	Logical	.TRUE. .FALSE.	.FALSE.
CKPNT	Initialization	Identifies external name of checkpoint file to be generated during the run. Its specification also results in system parameter JCKPNT being initialized as .TRUE., automatically activating checkpoint processing in the run (See section 3.10)	Alphanumeric name of seven characters or less		CPFILE
JECHO	System <sup>a</sup>	Controls printing of CS card image upon validation in primary edit phase (primary input stream editing) and secondary edit phases (procedure member editing)  JECHO=.TRUE. - print CS card images JECHO=.FALSE. - do not print CS card images	Logical	.TRUE. .FALSE.	.FALSE.
JLOG	System <sup>a</sup>	Controls printing of CS card images upon execution in executive processing phases  JLOG=.TRUE. - print CS card images JLOG=.FALSE. - do not print CS card images	Logical	.TRUE. .FALSE.	.TRUE.
LENGL	Initialization	Controls size (number of computer words within user's job field length) assigned to global dynamic storage (GDS) for this ANOPP run	Integer	≥3000	12000
MAXCARDS	Initialization	Controls maximum number of data cards allowed in primary input stream following an UPDATE, DATA, or TABLE CS	Integer	≥1	10000
NAETD	Initialization	Controls number of table-directory entries initially allocated for this ANOPP run	Integer	≥1	10
NAEUD	Initialization	Controls number of data-unit-directory entries initially allocated for this ANOPP run	Integer	≥5	25
NLPPM	Initialization	Controls number of lines per page to be used for ANOPP printed output during this run	Integer	≥15	48
NOGO	Initialization	Determines if ANOPP run is to be limited to primary edit phase execution only  NOGO=.TRUE. - primary edit phase only run NOGO=.FALSE. - normal ANOPP run	Logical	.TRUE. .FALSE.	.FALSE.

<sup>a</sup>System parameter values may also be set via the SETSYS CS.

### 3.3.2.2 STARTCS CS

Purpose: The STARTCS control statement indicates the beginning of the user's primary input stream.

Format: **STARTCS \$**

Restrictions:

STARTCS is valid only when used as the first control statement in the primary input stream. A label field is not allowed on the STARTCS.

A STARTCS CS may optionally be preceded by an ANOPP CS in the user's input deck.

### 3.3.2.3 ENDCS CS

Purpose: The ENDCS control statement indicates the end of the user's primary input stream and terminates the ANOPP run. Special usage of this control statement within ANOPP restart runs is documented in section 3.10.

Format: **[label]ENDCS \$**

**label**      label name

Examples:

```
ENDCS $
EOJ ENDCS $
```

Restriction: ENDCS is valid only when used as the last control statement in the primary input stream.

Examples

A normal ANOPP run would consist of a primary input stream consisting of

```
STARTCS $
  .
  .
other functional control statements
  .
  .
ENDCS $
```

The primary input stream may optionally be preceded by an ANOPP CS in the user's input deck:

```
ANOPP JLOG=.FALSE. $
STARTCS $
  .
  .
ENDCS $
```



#### 3.3.2.4 SETSYS CS

Purpose: The SETSYS control statement allows the user to assign new values to specified executive system parameters during an ANOPP run.

Format: [label<sub>*i*</sub>]SETSYS<sub>*i*</sub> sysparam<sub>*i*</sub>=value<sub>*i*</sub> [.....] sysparam<sub>*n*</sub>=value<sub>*n*</sub> ] \$

**label** label name

**sysparam** the name of an executive system parameter subject to change via the SETSYS CS (see Table 3.3-2)

**value** the value of the correct data type which is to replace the old value for the specified system parameter

Examples:

```
SETSYS JCON=.FALSE. $  
LABEL SETSYS JLOG=.TRUE., JECHO=.TRUE. $
```

Restriction: Assigned values must be of the correct data type and within the range prescribed for the system parameter.

Table 3.3-2. SETSYS CS System Parameters

System parameter name	Description	Value		Default value
		Type	Range	
ACCOUNT <sup>a</sup>	Controls accumulation of cost-accounting data associated with each functional module or executive processor  ACCOUNT=.TRUE. - accumulate accounting data ACCOUNT=.FALSE. - do not accumulate accounting data	Logical	.TRUE. .FALSE.	.FALSE.
JCKPNT	Controls performance of defined checkpoint operations during execution of CKPNT CS  JCKPNT=.TRUE. - defined checkpoint operations will be performed during execution of a CKPNT CS JCKPNT=.FALSE. - no operation will be performed during execution of a CKPNT CS	Logical	.TRUE. .FALSE.	.FALSE. (.TRUE. if CKPNT keyword present on ANOPP or RSTRT)
JCON	Determines the following executive action when the system nonfatal error flag is set during processing of a control statement or functional module  JCON=.TRUE. - execution will continue with processing of the next CS JCON=.FALSE. - execution will continue with next PROCEED CS encountered in the CS stream; if none is found, the ENDCS will normally terminate the ANOPP run	Logical	.TRUE. .FALSE.	.FALSE.
JECHO <sup>a</sup>	Controls printing of CS card images upon validation in primary edit phase (primary input stream editing) and secondary edit phases (procedure member editing); note that to effect printing of primary CS stream card images, JECHO must be set to .TRUE. via an ANOPP CS (see section 3.1.3.2)  JECHO=.TRUE. - print CS card images JECHO=.FALSE. - do not print CS card images	Logical	.TRUE. .FALSE.	.FALSE.
JLOG <sup>a</sup>	Controls printing of CS card images upon execution in the executive processing phases  JLOG=.TRUE. - print CS card images JLOG=.FALSE. - do not print CS card images	Logical	.TRUE. .FALSE.	.TRUE.
JRSTRT	Controls action of executive manager when ENDCS (simulated if not present) is encountered in inserted CS stream of a restart run  JRSTRT=.TRUE. - this restart run will continue with execution of the defined, checkpointed CS stream; execution will begin with the CS following the last CKPNT CS processed in the checkpoint run JRSTRT=.FALSE. - this restart run will terminate after execution of the ENDCS CS in the inserted input stream	Logical	.TRUE. .FALSE.	.FALSE. (.TRUE. if RSTRT CS begins run input stream)

<sup>a</sup>Parameter value may also be set via the ANOPP CS.

### 3.3.2.5 TITLE CS

Purpose: The TITLE control statement sets the value of the title line to be printed within the ANOPP standard page header. (See Figure 3.3-1.)

Format: [label]<sub>a</sub> TITLE<sub>a</sub> { paramname } \$

label label name

paramname the valid ANOPP name of a user parameter (defined via a PARAM CS) that contains the title line value, which must be a string constant not exceeding 128 characters (16A8)

string constant the title-line Hollerith string value, not to exceed 128 characters (128Hxxxx...x)

Examples:

```
LABEL1 TITLE 9HJET NOISE $
TITLE PARAM1 $
```

Restriction: The title value must not exceed 128 characters and will be printed with the left margin justified.

Line 1	mm/dd/yy	ANOPP	Lnn/nn/nn	PAGEnnnn
Line 2		Title	(128 character maximum)	
Line 3		Subtitle	(128 character maximum)	
Line 4		Label	(128 character maximum)	
Line 5			blank	

ANOPP page-header contents include:

mm/dd/yy	calendar date of ANOPP run
Lnn/nn/nn	current release level of ANOPP system
nnnn	page count initiated at zero for each run
Title	title-line value last assigned via a TITLE CS; default value = spaces
Subtitle	subtitle-line value last assigned from within a system or functional module
Label	label-line value last assigned from within a system or functional module

Figure 3.3-1. ANOPP Standard-Page-Header Content

### 3.3.2.6 PARAM CS

Purpose: The PARAM control statement establishes a new user parameter's value array of one or more elements, or it changes the value array of an already existing user parameter. A parameter's current data type and number of array elements is determined by the last value assigned it.

Format:

[label<sub>d</sub>]PARAM<sub>d</sub> pname<sub>1</sub>=expression<sub>1</sub> [...pname<sub>n</sub>=expression<sub>n</sub>] \$

where expression may take any of the following forms:

$$\text{pname}_2 \begin{pmatrix} \text{pname}_3 \\ \text{integer} \end{pmatrix}$$
$$\text{pname}_2 \left\{ \begin{array}{l} + \\ - \\ * \\ / \end{array} \right\} \text{numerical constant}$$

value

**label** label name

**pname<sub>1</sub>** the name of the user parameter for which a value is to be established or changed

**pname<sub>2</sub>** the name of a previously established user parameter

**pname<sub>3</sub>** the name of a previously established single-element user parameter of integer type

**value** an array of one or more elements of type numerical constant (I, RS, RD, or CS), logical constant (L), or string constant (A). Array elements must be separated by one or more delimiters (comma or space). Examples of each value-type specification are presented below. For all variations of a data-type control statement specification, see table 1 in section 3.1.2. When the value specification is for a multielement Hollerith string-type parameter, each string constant must specify the same character field length (nHxx...x)

**numerical constant** a numerical constant of type I, RS, RD, or CS

In the first form, the current value of an unsubscripted parameter `pname2` will be assigned to user parameter `pname1`. If `pname2` has a subscript, then the value of the specified element from the `pname2` array will be assigned to the user parameter `pname1`. If `pname3` is the subscript, it must be of integer type and contain only one element. Otherwise, the subscript must be an integer constant. `pname2` must contain at least the number of elements specified in the subscript.

In the second form, the specified algebraic operation will be performed against each element of this parameter's value array, with the result of each operation established in the corresponding element of parameter `pname1`'s value array. The type of `pname2` (I, RS, RD, or CS) must be the same as that of the specified numerical constant.

When performing an algebraic operation on complex single-precision-type (CS) numbers, each CS number is treated as a two-element single-precision array (RS, RS). Then, addition, subtraction, multiplication, and division of complex numbers will be calculated as follows, with A, B, C, and D each being real single-precision numbers:

$$\begin{aligned} (A,B)+(C,D) &= (A+C,B+D) \\ (A,B)-(C,D) &= (A-C,B-D) \\ (A,B)*(C,D) &= ((AC+BD),(AD+BC)) \\ (A,B)/(C,D) &= (((AC+BD)/(C**2+D**2)),((BC-AD)/(C**2+D**2))) \end{aligned}$$

Examples:

The following establishes real double-precision-type parameter L (1 element) and integer-type parameters G (1 element) and C (1 element):

```
PARAM L=1.5D-2, G=1234, C = 2 $
```

The following establishes string-constant-type user parameter KEYS (3 elements):

```
LABEL1 PARAM KEYS=4HFAN 4HJET 4HPROP $
```

The following establishes logical parameter RESULTS (2 elements):

```
PARAM RESULTS=.TRUE. .FALSE. $
```

The following establishes integer-type parameter NUMBS (2 elements) and real double-precision-type parameter STATS (2 elements):

```
PARAM NUMBS=3476 5962, STATS=17.67D27 -1.D+194 $
```

The following establishes complex single-precision-type parameter AGROUP (2 elements):

```
PARAM AGROUP=(7.,177E+17) (265.17E+2 -1.75) $
```

The following establishes real double-precision-type parameter T by using an algebraic operation based on the current value of parameter L; establishes integer-type parameter A with the value of current parameter G; and establishes real double-precision parameter B based on the current value of parameter T (note that the add operator + is followed by spaces):

```
PARAM T=L*17.62D-07, A=G, B=T + 2.0D00 $
```

The following assigns the third element of parameter KEYS to parameter E and the Cth element of parameter KEYS to parameter D:

```
PARAM E=KEYS(3) , D=KEYS(C) $
```

Restrictions:

The numerical constant used in a PARAM CS specified algebraic operation must be of the same numerical type as the specified user parameter (pname<sub>2</sub>) used in that operation.

The operators + or - must be followed by a blank.

Each value specified for a multielement user-parameter array must be of the same type.

Each value specified for a multielement Hollerith string parameter must be of the same character length (nHxx...x).

For a subscripted parameter name, that parameter must contain at least the number of elements specified within the parentheses. If the subscript (value inside the parentheses) is a parameter name, it must be of integer type and contain only one element.

### 3.3.2.7 EVALUATE CS

Purpose: The EVALUATE CS establishes a new user parameter of one element or changes the value of an existing user parameter element.

Format: [label<sub>1</sub>] EVALUATE<sub>1</sub> param-exp \$

where param-exp may take any of the following forms:

pname<sub>1</sub> = expression

pname<sub>2</sub>(pname<sub>3</sub>  
integer) = expression

**label** label name

**pname<sub>1</sub>** the name of the user parameter for which a value is to be established or changed. The value assigned to pname<sub>1</sub> is the result of the evaluated expression. The type assigned to pname<sub>1</sub> is the type associated with the expression.

**pname<sub>2</sub>** the name of a previously established user parameter whose specified element will be changed to the result of the evaluated expression. If **pname<sub>3</sub>** is the subscript, it must be of integer type and contain only one element. Otherwise, the subscript must be an integer constant. **Pname<sub>2</sub>** must contain at least the number of elements specified in the subscript. The type associated with the expression must match the type associated with **pname<sub>2</sub>**.

**pname<sub>3</sub>** the name of a previously established single-element user parameter of integer type

**expression** a sequence of constants, user parameters, subscripted user parameters, and function references separated by operators and parentheses

The arithmetic operators are as follows:

+ addition  
 - subtraction  
 \* multiplication  
 / division  
 \*\* exponentiation

The following functions are available:

Name	Definition	Number of arguments	Type of arguments	Example
ABS	$ x $	1	Any type	Y=ABS(X)
ANTILOG	$10^X$	1	I,RS,RD	Y=ANTILOG(X)
COS	cos(X)	1	Any type	Y=COS(X) with X in deg
INT	Express argument as integer	1	Any type	Y=INT(X)
LOG	$\log_{10}(X)$ where $X > 0$	1	I,RS,RD	Y=LOG(X)
REAL	Express argument as real	1	Any type	Y=REAL(X)
SIN	sin(X)	1	Any type	Y=SIN(X) with X in deg
SQRT	$\sqrt{X}$ where $X > 0$	1	Any type	Y=SQRT(X)
TAN	sin(X)/cos(X)	1	Any type	Y=TAN(X) with X in deg

All operations must be specified explicitly. For example, to multiply two variables A and B, the expression A\*B must be used. Thus, AB, (A)(B), or A.B will result in a syntax error with the exception of AB which is treated as a parameter name.

An expression is comprised of subexpressions. In the example

$$A/B - C*D**E$$

the following are subexpressions:

- (1) A/B
- (2) D\*\*E
- (3) C\* temp<sub>1</sub> (where temp<sub>1</sub> = D\*\*E)
- (4) temp<sub>3</sub> - temp<sub>2</sub> (where temp<sub>2</sub> = C\*(D\*\*E) and temp<sub>3</sub> = A/B)

The sequence in which an expression is evaluated is governed by the following rules which are listed in descending precedence:

- (1) Subexpressions delimited by parentheses are evaluated, beginning with the innermost subexpressions.
- (2) Subexpressions defined by arithmetic operators are evaluated according to the following precedence hierarchy:
  - \*\*      exponentiation
  - / \*     division or multiplication
  - + -     addition or subtraction
- (3) Subexpressions containing operators of equal precedence (+ and - or \* and /) are evaluated from left to right.
- (4) Subexpressions containing the exponentiation operator (\*\*) are evaluated from right to left.

In exponentiation, the following types of base and exponent are permitted:

Base	Exponent
Integer .....	Integer
Real .....	Integer, real, and double precision
Double precision .....	Integer, real, and double precision
Complex .....	Integer

The exponentiation is evaluated from right to left. The expression A\*\*B\*\*C is evaluated as (A\*\*(B\*\*C)).



A subexpression in a given level is evaluated at the highest type (CS RD RS I) of any parameter or constant in that subexpression. In exponentiation both the base and exponent are converted to the highest type in the subexpression with the exception of a complex base. For a complex base, any noninteger-type exponent is converted to an integer type. The type of the expression is the type associated with the last subexpression evaluated.

Examples:

Given the following parameter values, the expression to the left of the equal sign on the EVALUATE CS is evaluated and the result, 43.24, is stored in the user parameter table (UPT) under parameter X:

```
PARAM BAD=16.3, BC=5.3, CD=.96 $
PARAM DE=.3, EF=2.0, F=3 $
EVALUATE X = (BAD - BC)*F + (CD/DE)**EF $
```

The following results in the parameter Z taking the value 3.5 in the UPT:

```
EVALUATE Z = SIN(30.0) + REAL(3) $
```

Given the following parameter values, the following operations are performed. The square root of 100.0 is taken, converted to an integer, and then is added to the negative value of Z. The result, 6.5, is stored as the 1th, or 2nd, element of the parameter Y. The value of the expression must have the same type as Y:

```
PARAM Y=10.0, 3.9, 8.2, I=2 $
EVALUATE Y(I) = -Z + INT(SQRT(100.0)) $
```

Restrictions:

The operators + and - must be followed by a blank. For a subscripted parameter, that parameter must contain at least the number of elements specified within the parentheses.

If a multielement nonsubscripted parameter is in the expression, only the first element is used.

Each left parenthesis must have a corresponding right parenthesis.

The type associated with the expression must match the type associated with the parameter to the left of the equal sign if that parameter is a subscripted parameter.

### 3.3.2.8 UPLIST CS

Purpose: The UPLIST control statement will produce a printed list of all user parameters currently defined within the ANOPP run. The list includes each user parameter array's name, the alpha data type, internal ANOPP numeric data-type code, and the number and value of each element in the array.

Format: label<sub>1</sub>UPLIST \$

label label name

Example:

The following control statement

LABEL1 UPLIST \$

if executed when the user parameters established in a run were those created by the PARAM CS examples and EVALUATE CS examples presented in the two previous sections, would produce the following listing:

USER-PARAMETER-TABLE DUMP

NAME	TYPE <sup>a</sup>	TYPE CODE	ELEMENT	VALUE
AGROUP	CS	4	( 1)	.70000000000000E+01
			( 2)	.17700000000000E+20
A	I	1	( 1)	.26517000000000E+05
B	RD	3	( 1)	1234
BAD	RS	2	( 1)	.20000000264299999999999999999999D+01
BC	RS	2	( 1)	.16300000000000E+02
C	I	1	( 1)	.53000000000000E+01
CD	RS	2	( 1)	2
D	A	-4	( 1)	.96000000000000E+00
DE	RS	2	( 1)	JET
E	A	-4	( 1)	.30000000000000E+00
EF	RS	2	( 1)	PROP
F	I	1	( 1)	.20000000000000E+01
G	I	1	( 1)	3
I	I	1	( 1)	1234
KEYS	A	-4	( 1)	2
			( 2)	FAN
			( 3)	JET
L	RD	3	( 1)	PROP
NUMBS	I	1	( 1)	.14999999999999999999999999999999D-01
			( 2)	3476
RESULTS	L	6	( 1)	5962
			( 2)	T
STATS	RD	3	( 1)	.176699999999999999999999999999998D+29
			( 2)	-.9999999999999999999999999999999975+194
T	RD	3	( 1)	.2642999999999999999999999999999989D-07
X	RS	2	( 1)	.43240000000000E+02
Y	RS	2	( 1)	.10000000000000E+02
			( 2)	.65000000000000E+01
			( 3)	.82000000000000E+01
Z	RS	2	( 1)	.35000000000000E+01

<sup>a</sup>TYPE CODE is an integer internal-system-type code. Of interest to the external ANOPP user is the TYPE CODE for type A (Holerith string), expressed as a negative n value, where n is the uniform character length of the string for elements of the user parameter array.

Restrictions: None.

### 3.3.2.9 EXECUTE CS

**Purpose:** The EXECUTE control statement calls a specified functional module (FM) into execution and allows the user to identify alternate data items to be used in satisfying specific data requirements of the functional module.

**Format:**

```
[label1]EXECUTE1fmname1[refname1=altname1, ..., nrefnamen=altnamen] $
```

**label** label name

**fmname** the name of a functional module to be executed.

**refname** the name known within the functional module for a required data unit, data member, table member, or user parameter. It is also the name by which the data item is identified in the section 4 documentation of the FM's data requirements.

**altname** the altname name corresponding to the FM known refname. The altname is the name of a data unit, data member, table member, or user parameter to be used in satisfying the requirement for a data item known within the FM by the refname.

**Examples:**

```
L1 EXECUTE JET UNIT1=MYUNIT, B=D $
```

```
EXECUTE PROP $
```

**Restrictions:**

The fmname must be a functional module currently installed on the ANOPP system.

User-supplied data items must be of the type and format specified for the FM application.

**Examples**

The following examples illustrate typical control statements required to satisfy FM data requirements. Functional documentation of the various control statements used are found in other subsections of this manual.

**Example 1:** The FM JET1 is to be executed. It requires a logical user parameter JSELECT, a real single-precision user parameter NSPEED with three elements, and data unit JETUNIT on which reside members TAB1, TAB2, DMEM1, and DMEM2. No substitute data items are to be used; once established in the run, all required items will be accessible by the FM.

```

STARTCS $
:
ATTACH JETUNIT/FILE29/ $ COMMENT - ESTABLISH JETUNIT IN RUN
PARAM JSELECT=.TRUE., NSPEED=1.33E05 -0.52 19.7 $
EXECUTE JET1 $
:
ENDCS $

```

Example 2: Execute FM MACH8. Build a required data member identified in FM documentation as UNIT1 (STATJ) via the CREATE CS and UPDATE CS. Functional module MACH8 creates a logical user parameter documented as RESULT. Because the user has already created a parameter named RESULT earlier in the run for a different purpose, an alternate name specification is used to cause the FM-created user parameter to be called M8RESULT.

```

STARTCS $
:
CREATE UNIT1 $
UPDATE NEWU=UNIT1 SOURCE=* $
-ADDR OLDM=*, NEWM=STATJ FORMAT=11H2I,RS,RD,L$ MNR=2 $
1757 294372 2E+01 3D-01 .FALSE. $
297 87536 375.15E-12 15.81D19 .TRUE. $
END* $
EXECUTE MACH8 RESULT=M8RESULT $

```

### 3.3.2.10 ATTACH CS

Purpose: The ATTACH control statement establishes one or more previously created data units in an ANOPP run via entries in the system data unit directory (DUD). Each data unit must have previously been created on an external mass-storage file currently assigned to the job in the external system.

Format: [label<sub>*a*</sub>]ATTACH<sub>*a*</sub> dun<sub>*1*</sub>/efn<sub>*1*</sub>/[...<sub>*a*</sub> dun<sub>*n*</sub>/efn<sub>*n*</sub>/] \$

*label* label name

*dun* the name of the data unit to be established in the ANOPP run

*/efn/* the name of the external mass-storage file, currently assigned to the user's job, on which data unit *dun* resides

#### Examples:

```

APP ATTACH JETU1/FILEJ/, FANU1/FILEF/ $
ATTACH PROPU1/EFN/ $

```

#### Restrictions:

Each data unit name (*dun*) must be unique with respect to all other data unit names currently known in the run.

Each external file name (*efn*) must be unique with respect to all other *efn*'s currently known in the run.

Each data unit (dun) specified must have previously appeared on a DETACH CS.

If the data unit was created in a previous ANOPP job, the external file (efn) it resides on must be currently assigned to this job via external control cards.

### 3.3.2.11 CREATE CS

Purpose: The CREATE control statement initially defines in the ANOPP run an empty data unit, to reside on a random-access mass-storage device, and makes it available for subsequently generated table, data, and procedure members.

Format: [label<sub>1</sub>]CREATE<sub>1</sub>dun<sub>1</sub> [/efn<sub>1</sub>/] [...<sub>1</sub>dun<sub>n</sub> [/efn<sub>n</sub>/]] \$

**label** label name

**dun** the name of the new data unit to be entered in the system data unit directory whose members are to be generated during the run

**/efn/** the name of the external mass-storage file, assigned to the job in the external system, on which the corresponding dun will reside. If efn is omitted, a scratch (temporary) file will be established by the ANOPP system.

#### Examples:

```
LABEL1 CREATE UNIT1, UNIT2/EFN2/, JETUN1/JETFILE/ $  
CREATE JOBFILE1, JOBFILE2 $
```

#### Restrictions:

The dun and efn cannot, respectively, be the same as any other dun and efn that are either currently known to the ANOPP run (entered in the data unit directory) or appear on the same CREATE CS.

Any data that currently reside on external file efn will be destroyed during the creation of new data unit dun, even if the old data were on an ARCHIVE'd ANOPP data unit.

Note: Reference to a unit that has not been defined by a CREATE CS during an ANOPP run will cause the unit to be dynamically created on a scratch file.

### 3.3.2.12 DETACH CS

Purpose: The DETACH control statement removes a data unit from the run-life data base. Unless the data unit resides on a scratch (temporary) file, it may be assigned to the current or a future ANOPP run via an ATTACH control statement.

Format:

[label<sub>a</sub>]DETACH<sub>a</sub>dun<sub>1</sub>[...<sub>a</sub>dun<sub>n</sub>] \$

label label name

dun name of the data unit, known to the ANOPP run, that is to be detached

Example:

```
L1 DETACH PROPUI, JETU1 $
DETACH AIRPRTS $
```

Restriction: Each data unit named must be currently known in the ANOPP run (via an entry in the system data unit directory).

### 3.3.2.13 PURGE CS

Purpose: The PURGE control statement removes a data unit from the run-life data base; in addition, the external file on which the unit resides will no longer be assigned to the user's job in the external system.

Format: [label<sub>a</sub>]PURGE<sub>a</sub>dun<sub>1</sub>[.....<sub>a</sub>dun<sub>n</sub>] \$

label label name

dun name of a data unit whose current run and job assignment is to be terminated

Example:

```
LABEL1 PURGE DU1, DU2, DU3 $
PURGE JETU1 $
```

Restriction: If the data unit is to be re-ATTACH'ed in the same job, or the external file on which the dun resides is to be made permanent via external control cards that follow the ANOPP run (in the user's job deck), the DETACH CS should be used instead of the PURGE CS.

### 3.3.2.14 CATALOG CS

**Purpose:** The CATALOG control statement produces printed summary information about named data units or, optionally, about all data units known to the ANOPP run. In addition, an optional FULL report presents summary information about each resident member of each specified unit.

**Format:** [label<sub>d</sub>]CATALOG<sub>d</sub> [LIST={SUMMARY  
FULL}]<sub>d</sub> [du<sub>1</sub>.....<sub>d</sub>du<sub>n</sub>] \$

**label** label name

**LIST=** optional specification allows selection of the type of catalog report desired for the specified du's (data units). If omitted, LIST=FULL is assumed.

**LIST=SUMMARY:** A SUMMARY CATALOG report will be produced for each data unit specified, and will include:

- The data unit name
- The name of the external file on which the data unit resides
- The name of each member that resides on the unit

**LIST=FULL (default):** A FULL CATALOG report will be produced for each data unit specified and will include:

- The SUMMARY report of items listed above for each member on the data unit:
- The maximum number of records allowed
- The current number of records
- The member (record) format
- The maximum record length
- The date and time of day the member was created

**du** (optional) name of a data unit, currently known in the run, about which the user wants a CATALOG report of the selected type. If omitted, the selected report will be produced for all run-life data units, except for system units XSUNIT and XRUNIT (if a restart run). The user may specify: XSUNIT and XRUNIT, in a checkpoint/restart run.

Examples:

The following will produce FULL reports on all run-life data base units, except for system units XSUNIT and XRUNIT:

LABEL1 CATALOG \$

The following will produce a SUMMARY report on data unit DUN7:

CATALOG DUN7 LIST=SUMMARY \$

The following will produce FULL reports on all run-life data base units except for system units XSUNIT and XRUNIT:

CATALOG LIST=FULL \$

The following will produce FULL reports on data units DUN1 and XSUNIT:

CATALOG LIST=FULL, DUN1, XSUNIT \$

Restriction: Specified du's (data units) must be currently established in the run-life data base.

3.3.2.15 MEMLIST CS

Purpose: The MEMLIST control statement produces a printed report about specified members residing on run-life data units. The report may optionally be limited to summary information (member structure and unit residence) on each of the specified members; or the FULL report can be produced to include the summary information plus the listing of values currently stored in the specified members' records.

Format: [label<sub>d</sub>]MEMLIST<sub>d</sub> [LIST={FULL  
SUMMARY}]<sub>d</sub> member<sub>1</sub> [.....]<sub>d</sub> member<sub>n</sub> \$

where each specified member<sub>i</sub> combination has the form

du(mn) [(rm[,rn])] [<sub>d</sub>FORMAT=format]

**label** label name

**LIST=** optional keyword allows selection of the type of MEMLIST report desired for the specified members. If omitted, LIST=FULL is assumed.

**LIST=SUMMARY:** A SUMMARY report will be produced for each member specified and will include:

The name of the data unit on which the member resides  
The name of the external file on which the member's data unit resides  
The name of the member



The member's:

- Maximum number of records
- Current number of records
- Format specification
- Maximum record length
- Date of creation
- Time of creation

**LIST=FULL** (default): A FULL MEMLIST report will be produced for each member specified and will include:

The SUMMARY information listed above  
The data values of each member record specified by param (rm,rn) below, or all member records if (rm,rn) is omitted; printed in the format specified by keyword **FORMAT=** below, or in the format under which the member was created if **FORMAT=** is omitted.

**du(mn)** the name of the data unit (member) to be reported on

**(rm)** the optional specification of the integer sequence number (rm)  
or  
of the record or the range of integer sequence numbers (rm,rn)  
**(rm,rn)** of the records on the member whose values are to be listed. If omitted, the values of all the member's records will be displayed. If **LIST=SUMMARY**, this parameter is ignored.

**FORMAT =** optional keyword parameter specifies the member format to be used in printing the indicated records. This field is ignored if **LIST=SUMMARY**.

If omitted, the indicated records will be printed in accordance with the format specification under which the member was created via a DATA CS, LOAD CS, TABLE CS, or UPDATE CS. Unformatted records are always printed in octal format.

**FORMAT=0** produces an octal printout of the data content of the specified records, regardless of the format under which the member was created.

**FORMAT=2HCI** indicates that the member has a card-image format; each card-image record will be listed on a single print line.

**FORMAT=specified ANOPP** format. A valid ANOPP format specification produces a printout of records in that format. The specified format is assumed to be compatible with the records to be printed and is not validated internally. A member (record) format is expressed as a character string terminating with a S, as shown in the following examples:

FORMAT=12H5(I),A30,RSS (fixed length format)  
FORMAT=17HI,2A10,\*(I,10RS)\$ (variable length format)

Examples:

The following will print a SUMMARY report on data members DMEN1 and DMEN2, both resident on data unit DUN1:

```
LABEL1 MEMLIST LIST=SUMMARY, DUN1(DMEN1), DUN1(DMEN2) $
```

The following will print a FULL report on data member DMEN3 of data unit DUN1. Each record on the member will be listed, and the data values will be displayed in octal code.

```
MEMLIST LIST=FULL, DUN1(DMEN3) FORMAT=0 $
```

The following will print a FULL report on data member DM9 of data unit DU1, listing the values of member records 11 and 12 in the format under which the data member was created:

```
MEMLIST DU1(DM9), (11,12) $
```

Restrictions:

Specified data units and members, du(mn), must be currently assigned to the run-life data base.

Specified member record occurrences, (rm) or (rm,rn), must be valid for the member being listed.

3.3.2.16 TABLIST CS

Purpose: The TABLIST control statement produces a printed report containing summary information about the ANOPP data tables that reside on specified run-life members. In addition, a FULL TABLIST report may be selected that includes the listing of the specified tables' assigned values.

Format: [label<sub>d</sub>]TABLIST<sub>d</sub> [LIST={FULL  
SUMMARY}<sub>d</sub>] du<sub>1</sub>(tm<sub>1</sub>)[...<sub>d</sub>du<sub>n</sub>(tm<sub>n</sub>)] \$

**label** label name

**LIST=** optional keyword allows selection of the type of TABLIST report desired for the specified member resident tables. If omitted, LIST=SUMMARY is assumed.

**LIST=SUMMARY** (default): A SUMMARY report will be printed about the ANOPP data table that resides on each specified data unit. The SUMMARY report includes the following:

- Residence data unit name
- Residence external file name
- Residence table member name
- Creation date
- Creation time of day

Number of independent variables  
Length in words  
Allowable interpolation procedures

**LIST=FULL:** A FULL TABLIST report will be printed on each specified table and will include:

The SUMMARY report information (above)  
The following data on each independent variable defined for the table:

The type code  
The number of table values  
The procedure for extrapolation beyond the last assigned value  
The assigned values

and the following data on the dependent variable:

The type code  
The number of table values  
The assigned values

**du(tm)** the run-life data unit name (table member name) on which the data table to be reported on resides

Examples:

The following will produce a FULL TABLIST report on data unit DUN1 resident table member DTBL1:

```
LABEL1 TABLIST LIST=FULL, DUN1(DTBL1) $
```

The following will produce a SUMMARY report on data unit DUN2 resident table member DTBL2:

```
TABLIST LIST=SUMMARY DUN2(DTBL2) $
```

The following will produce a SUMMARY report on data unit DUN1 resident table members TBL1 and TBL2, and on data unit DUN3 resident table member TBL1:

```
TABLIST DUN1(TBL1), DUN1(TBL2), DUN3(TBL1) $
```

Restrictions:

The ANOPP table residing on a specified member must have been created either via a TABLE CS or internally by a functional module.

Each specified-table residence data unit (table member) must currently be assigned to the run.

### 3.3.2.17 Archiving (ARCHIVE CS)

The ARCHIVE CS is used to assign a permanent ANOPP read-only status to an existing data unit. It is intended for the user who has built a data unit and now wants to prevent any possibility of a future change being made to the data content of that unit as a result of subsequent TABLE CS, UPDATE CS, or functional module processing. The archive (read-only) status, once established, is recorded in the internal header of the unit and cannot be reversed during the current or any future ANOPP run or job.

Purpose: The ARCHIVE control statement prohibits any future request to write on the named unit or units via TABLE CS, UPDATE CS, or functional module processing, whether in the current run or in any subsequent ANOPP run or job.

Format: [label<sub>a</sub>]ARCHIVE<sub>a</sub> dun<sub>1</sub> [.....<sub>a</sub> dun<sub>n</sub>] \$

label     label name

dun        name of the data unit to be archived

Examples:

```
LAB ARCHIVE UN1, UN2, UN20 $
ARCHIVE, U50 $
```

Restrictions:

Each unit name must be unique with respect to the other unit name(s) on the control statement.

Each unit to be archived must be currently assigned in the ANOPP run.

The archive (read-only) status prevents data unit content modification via subsequent TABLE CS, UPDATE CS, or functional module processing. It does not prevent either the modification of the content of the file via subsequent external (non-ANOPP) processing or the creation of a new unit on the file via CREATE CS or LOAD CS processing.

ANOPP system scratch units (a unit whose associated external file name begins with the letters "XEFN") may not be archived.

### 3.3.2.18 LOAD CS

**Purpose:** The LOAD control statement creates new run-life data units by using as a source of data the units and members previously stored on a sequential library file via an UNLOAD CS (in either the current or a previous ANOPP run). The specified library file must be currently assigned to the user's job in the external system. The LOAD CS provides for selective loading and/or renaming of data units and data members stored on the sequential library file.

**Format:** [label<sub>i</sub>]LOAD/sefn/[du<sub>1</sub>,...,du<sub>n</sub>] \$

where each du<sub>i</sub> has the format

$$\text{ndu} \left[ \left[ \text{/efn/} \right] \left[ \text{=odu} \right] \left[ \left( \text{mn}_1 \left[ \dots \text{mn}_n \right] \right) \right] \right]$$

where each mn<sub>i</sub> has the format

~~nmn~~=~~omn~~

**label** label name

**sefn** the name of a sequential library file, currently assigned to the user's job in the external system, from which stored data unit (member) data will be copied

**ndu** (optional) the name of a new data unit to be established in the run-life data base. If members residing on more than one LF stored data unit (odu) are to be copied to the same new ndu, repeat specification of this field is required. If no ndu is specified, all of the library resident units will be established in the run-life data base on scratch files with no change in names or data contents.

**/efn/** (optional) the name of the external random access file, assigned to the user's job, on which the new data unit will reside. If omitted, a scratch (temporary) file will be used for that ndu. If more than one LF stored data unit (odu) is to supply members for the new du, the /efn/ specified for that new ndu should occur only once on the LOAD CS.

**=odu** (optional) the name of the library resident data unit whose members are to be copied to the new du. If omitted, the library resident data unit (odu) and the run-life data unit (ndu) will have the same name.

**nmn** (optional) the name of the member to be created on the ndu. If no nmn is specified, all odu members are copied to ndu.

**=omn** (optional) the name of library unit odu resident member to be copied to the ndu member nmn. If omitted, the odu resident member (omn) and the run-life member (nmn) will have the same name.

Examples:

The following examples will illustrate the various features of the LOAD CS. Each example will use a previously created sequential library file named OLDLF, assumed to be currently assigned to the user's job in the external system. The OLDLF was created by an UNLOAD CS that stored copies of data units UNIT1, UNIT2, UNIT3, and UNIT4; each stored data unit contains members MEM1, MEM2, MEM3, and MEM4.

Example 1: The following would result in all four OLDLF stored units being established as run-life data units under their stored names with no change in their member content. All of the new units would be created on scratch (temporary) external files since no efn's were specified.

```
LABELA LOAD/OLDLF/ $
```

Example 2: The following would result in new run-life data unit UNIT3 being established on external file FILEB. The UNIT3 would contain MEM1, MEM2, MEM3, and MEM4.

```
LOAD/OLDLF/UNIT3/FILEB/ $
```

Example 3: The following would result in new run-life data unit JETU being established on external file FILEC. Unit JETU would contain members (MEM1 (a copy of library stored member UNIT4 (MEM1)) and member ALTTBL (a renamed copy of library stored member UNIT4 (MEM2))).

```
LOAD/OLDLF/JETU/FILEC/=UNIT4 (MEM1, ALTTBL=MEM2) $
```

Example 4: The following would result in the establishment of new run-life data units JU and PU. Unit JU would reside on external file JFILE and would be a renamed copy of OLDLF stored unit UNIT1. Unit PU would reside on external file PFILE and contain six members. The PU members MEM1 through MEM4 would be exact copies of OLDLF stored unit UNIT2's four members. The PU members MEM5 and MEM6 would, respectively, be renamed copies of members MEM1 and MEM2 of OLDLF stored unit UNIT3. Note that /PFILE/, the external file name for new unit PU, is specified only once.

```
LOAD/OLDLF/JU/JFILE/=UNIT1
      PU/PFILE/=UNIT2
      PU=UNIT3 (MEM5=MEM1, MEM6=MEM2) $
```

Restrictions:

Each specified new run-life data unit (ndu) and resident external file name /efn/ must be currently unknown to the run.

Each specified /efn/ and the external library file (/sefn/) must be assigned to the user's job in the external system. Any data currently residing on file efn will be destroyed during the creation of run-life unit ndu.

Each specified library file must have been previously created via an UNLOAD CS, either in the current or a prior ANOPP run.

Each new member must possess a name unique as to the names of all other members created on the new unit.

No member residing on the library file may be the data source for more than one new run-life member created by a single LOAD CS. Following the specification of JU/JFILE/=UNIT1, no subsequent data source specified on that LOAD CS, for any new data unit, could be a member of OLDF stored unit UNIT1.

### 3.3.2.19 UNLOAD CS

Purpose: The UNLOAD control statement establishes a new ANOPP library file on a sequential external tape or mass-storage file. Copies of current run-life data units, optionally consisting of all or some (specified) of their resident members, are unloaded onto the library file. Once established, the library file is available as a data source for subsequent LOAD CS applications within the current or a future ANOPP run.

Format: [label<sub>i</sub>]UNLOAD/sefn/[du<sub>1</sub>,.....du<sub>n</sub>] \$

where each du<sub>i</sub> has the format

du(mn<sub>1</sub>,.....mn<sub>n</sub>)

**label** label name

**/sefn/** the name of the sequential external tape or mass-storage file, assigned to the user's job in the external system, onto which the run-life unit (member) data are to be copied. The sefn must not be currently known in the run via a previous UNLOAD CS or LOAD CS.

**du** the name of a current run-life data unit whose member data are to be copied (unloaded) onto the sefn. If no du is specified, all run-life data base units (except system units DATA, XSUNIT, and XRUNIT) will be copied to the library file.

**mn** the name of a specified du resident member that is to be copied to the library. If one or more mn's are specified, only those named members are copied from that du. If no mn list is specified for a du, all of that unit's members are copied to the library.

#### Examples:

All data units currently known to the ANOPP run, except for system data units XSUNIT, DATA, and XRUNIT, will be unloaded onto the following sequential library file TAPE1:

```
LABEL1 UNLOAD/TAPE1/ $
```

All run-life data units UNIT1 and UNIT3 will be unloaded as follows; UNIT2 will be unloaded but will contain only members MEM3 and MEM5:

UNLOAD/BACKUP/UNIT1, UNIT2(MEM3, MEM5),UNIT3 \$

Restrictions:

No duplication of a combination of data unit and data member is allowed on the UNLOAD CS. If a data unit appears more than once on the same UNLOAD CS, each occurrence must specify a unique (for that du) member list.

Once a library file has been established in a run via either a LOAD CS or UNLOAD CS, a subsequent UNLOAD CS may not address the same file in the same run unless it has been removed from the system by using the DROP CS.

3.3.2.20 DROP CS

Purpose: The DROP control statement terminates the ANOPP run assignment and the job assignment of the external system of a sequential library file. The dropped library file will be inaccessible for the rest of the run and inaccessible for the rest of the user's job unless its job assignment is reestablished via subsequently processed external control cards.

Format: [label<sub>1</sub>]DROP /sefn/[ ,.....,/sefn/] \$

label label name

sefn the external file name of a run-life sequential library file  
whose run and job assignments are to be dropped

Examples:

```
END DROP /TAPE1//, /DISK1//, /JETSET/ $
DROP /MASTERF/ $
```

Restrictions:

When a tape resident library file is dropped, the tape is physically unloaded by the external system.

When a mass-storage resident library file is dropped, that file is no longer assigned to the user's job in the external system.

A new library file (created via an UNLOAD CS) must not be dropped if the user intends to reference that file in the external system via subsequently processed external control cards.

Once a specific library file (sefn) has been dropped in a user's job, the same sefn specification may appear on a subsequent UNLOAD CS. However, unless the job assignment of the dropped file was reestablished via external control cards, the subsequent UNLOAD CS will not address the same physical file that was dropped.



### 3.3.2.21 END\* Input Terminator

Purpose: The END\* input terminator card indicates the end of a primary input stream card-image set required by the previous DATA, UPDATE, or TABLE control statement.

Format: END\* \$

Examples:

See DATA CS, UPDATE CS, and TABLE CS for specific usage examples.

Restrictions:

The END\* CS is valid in the primary input stream only.

A label field is illegal on the END\* card.

### 3.3.2.22 DATA CS

Purpose: The DATA control statement creates or recreates a card-image (CI) formatted member on system work unit DATA. The created member may be a new version of a member created via a previous DATA CS in the same run. The records of the member will be the card images that immediately follow the DATA CS in the user's primary input stream. This input card-image set is terminated by the first END\* card encountered following the DATA CS.

Format: [label<sub>d</sub>]DATA<sub>d</sub>DM=mn \$

label      label name

DM=mn      name of the card-image member to be created on system data unit  
DATA

Examples:

```
LABELA DATA DM=FM1DATA $
1    2    3    4    5    .TRUE.
1.5 2.5 3.4 4.5 5.5 .TRUE.
END* $
```

```
DATA DM=PROC MEM $
LOAD/ANOPDAT/COANJET, JRS $
EXECUTE FMNEWJ UNIT1=COANJET UNIT2=JRS $
DETACH COANJET JRS $
END* $
```

Restrictions:

The DATA CS is legal only in the user's primary input stream.

The terminating END\* card will not be a card image on the created member.

### 3.3.2.23 TABLE CS

**Purpose:** The TABLE control statement builds a user-specified ANOPP data table on a run-life table member. The data table is structured in accordance with a set of user-supplied table-description card images. The card-image set may reside immediately following the TABLE CS in the user's primary input stream or on a user-specified card-image run-life member. Description cards required for the definition of ANOPP data table types are documented below.

**Format:** [label<sub>a</sub>]TABLE<sub>a</sub>du(tm)<sub>a</sub>type<sub>a</sub>SOURCE={dun(mn)} \$

**label** label name

**du(tm)** the data unit name (table member name) on which the data table is to reside. The du may be any user-established run-life unit or system unit DATA.

**type** the integer ANOPP table type of the data table being built. Current valid table types are documented below.

**SOURCE=** specifies where the table-description card-image set, required to define the ANOPP table-type specified, resides. Table-description card sets are documented below.

**SOURCE=\*** indicates that the card-image set follows immediately behind the TABLE CS in the user's primary input stream. The set must be terminated by an END\* input terminator card.

**SOURCE=dun(mn)** indicates run-life data unit name (member name) on which the card-image table-description set resides.

#### Examples:

A type-1 data table will be built on unit FMIUNIT resident table member FANTAB. The table-description set will be read from card-image member FANTDCS residing on system unit DATA as follows:

```
LABELA TABLE FMIUNIT(FANTAB), 1, SOURCE=DATA(FANTDCS) $
```

The same table creation results as for example 1 in section 3.6.1, but here the table-description set is read from the user's primary input stream and is terminated by an END\* card as follows:

```
TABLE FMIUNIT(FANTAB) 1 SOURCE=* $
  INT= .....
  IND1=.....
  DEP= ..... } table-type-1 description card set
END* $
```

**Restriction:** A TABLE CS having a specification of SOURCE=\* is valid only in the user's primary input stream; specifications of SOURCE=dun(mn) are valid in any CS stream.

ANOPP data type-1 description cards: The table type-1 description cards are of the following format:

```
INT= ipc, ipc, ipc
INDn= ifc,nv,exu,exl,ival1,.....,ivalnv
DEP= dfc,dval1,dval2.....
```

which are defined as follows:

INT card contains the integer codes for the interpolation procedure that will be permitted on this table

ipc integer code:  
0 no interpolation  
1 linear interpolation  
2 cubic-spline interpolation

IND<sub>n</sub> card contains the description of and the table values for the nth independent variable where  $n < 4$

ifc (format code) the alpha data-type code of the nth independent variable  
0 ordered position from 1 to nv; independent variable values not entered  
I integer  
RS real single precision  
RD real double precision

nv number of values for the nth independent variable

exu integer code for the extrapolation procedure to be used (during interpolation) if a specified value for this independent variable falls beyond the last table value (ival<sub>nv</sub>) for the independent variable  
0 no extrapolation allowed  
1 use the independent-variable table value closest to the specified value  
2 extrapolation is linear when using the last two table values for the independent variable

exl integer code for the extrapolation procedure to be used (during interpolation) if a specified value for this independent variable falls before the first table value (ival<sub>1</sub>) for the independent variable  
0 no extrapolation allowed  
1 use the table value of the independent variable closest to the specified value  
2 extrapolation is linear when using the first two table values for the independent variable

ival      nv table values for the nth independent variable in ascending or descending order. Values are separated by blank or comma and may extend over several card images. If ifc=0, values are not included.

NOTE: If only a single value is given for the independent variable (nv=1) and either exu or exl equals 2, the system will print an informative diagnostic and build the table with the extrapolation param (exu or exl) set to 1.

DEP card      contains the description and table values for the dependent variable and must follow the IND<sub>n</sub> cards

dfc      (format code) alpha data type of the dependent variable

I      integer  
RS      real single precision  
RD      real double precision  
CS      complex single precision

dval      the table value for the dependent variable, separated by commas or blanks, and extending over several card images as required. The order of dependent variables is such that the first independent variable varies first, the second variable varies second, the third variable varies third, and the fourth variable varies last. See table 1 in section 3.1.2 for numeric-value specification examples.

END\* card      required if SOURCE=\*

Examples of TABLE CS type-1 table descriptions:

```
LAB TABLE UNI(DMS),1,SOURCE=* $
INT=0,1
IND2=I,2,0,1,5,10
IND1=RS,3,0,1,1.5,2.0,4.5
DEP=I,3,5,7,8,9,10
END* $
```

```
TABLE UNI(DMT) 1 SOURCE=* $
INT=0 1
IND1=I 2 1 1 2 3
IND2=RS 2 2 2 3.4 7.8
IND3=RD 2 2 2 4DO 3DO
IND4=RS 2 2 2 3. 2.
DEP=CS (12.,12.) (13.,13.) (13.,13.) (14.,14.) (11.,11.)
      (12.,12.) (12.,12.) (13.,13.) (11.,11.) (12.,12.)
      (12.,12.) (13.,13.) (10.,10.) (11.,11.) (11.,11.) (12.,12.)
END* $
```

NOTE: Currently, the only valid data-table type recognized by ANOPP is type 1.

### 3.3.2.24 The UPDATE CS

#### Overview

The processing of an UPDATE CS in the user's input stream initiates the update processing phase. During this phase, the system will generate all new members onto a specified run-life data unit. During the update phase, and in accordance with a user's set of update directives, the system will generate the new unit members by:

- (1) Copying the contents of specified existing run-life members
- (2) Copying new-member data records directly from the primary input stream
- (3) Building new members by combining records copied from an existing run-life member and from the user's update directive set

The UPDATE CS that initiates the update processing phase specifies:

- The new unit to receive the generated members
- The source where the card-image update-directive set resides
- The processing mode (create or revise) for this update phase
- The content of the update processing report

The new unit to be built during the update phase should have been previously established in the run via a CREATE, ATTACH, or LOAD CS. At the start of the update phase, any resident members on the user-specified new unit are destroyed. The update-directive set is a set of card images that either resides on a user-specified run-life member or that immediately follows the UPDATE CS in the user's primary input stream (an END\* input terminator card establishing the end of the directive set). The update directives allow the user to define the new members to reside on the unit being built and to specify the source of each new member. The update processing mode is determined by the user's inclusion or omission of a primary old data unit specification on his UPDATE CS.

The update create mode is in effect if a primary old unit is not specified on the user's UPDATE CS. Under this mode, the -ADDR member-level directive may be used to generate new members copied from specified run-life units or from within the input stream. The -CHANGE member-level directive, and its subordinate record-level directives, may be used to mix existing and new records onto a new member. In create mode the user must always specify on each member-level directive the run-life unit residence of any data-source member. Each new member must be generated by a single member-level directive application.

The update revise mode is in effect if the user specifies a primary old unit on his UPDATE CS. A specified primary old unit should be the primary source of run-life member and record data for the new unit being built. Typically, this is an existing unit, the content of which is being revised or updated. UPDATE CS applications never result in a change of data occurring on the old source unit or member. Under the revise mode, the user has access to all the directive capabilities available under the create mode, including the use of any run-life member as a record source. In addition,

he has access to two additional member-level directives, the -COPY and -OMIT directives. The -COPY directive allows for copying (unchanged) specified members from the primary old data unit. The -OMIT directive allows for specific identification of primary old data-unit resident members that are not to be copied to the new unit. The -COPY and -OMIT directives, used in conjunction with the optional UPDATE CS keyword ALL, allow the user to copy by default all the primary old unit members not specifically omitted, copied, replaced, or changed by a member-level directive. Revise-mode processing also allows the user to default on his data-source unit specification on most update directives, in which case the primary old unit is assumed.

A printed update processing report is produced by the system to describe the data unit, member, and record generation that occurred during the processing phase just completed. The report will contain as a minimum the header section described below. The optional LIST= parameter of the UPDATE CS allows the user to select additional sectional content to appear in the report. He may specify any combination of the four alpha characters, each of which requests a different titled update report section to be printed. The report sections, the LIST= value specification that selects them, and their information content are given as follows:

**Header section:**

Printed following the completion of each update processing phase  
Contains:

- The update mode (create or revise) in effect during the processing phase
- The name of the new data unit generated
- If revise mode, the name of the primary old unit used
- The source from which the update directive set was read, either the user's primary input stream or a run-life data unit name (member name)
- The user specified UPDATE CS LIST= selections

**Directive echo section:**

Included in report if LIST= selection contains letter E  
Prints each card image in the user's update directive set

**Summary section:**

Included in report if LIST= selection contains letter S  
Contains:

- The total number of members generated on the new data unit
- For each new member generated:
  - The name of the member
  - The number of records on the member
  - The format specification of the member (records)
  - The maximum record length (number of computer storage words)
  - The type of the update member-level directive (-COPY, -ADDR, or -CHANGE), or the UPDATE CS keyword ALL that controlled the generation of the member

**ADDR member section:**

Included in report if LIST= selection contains letter A  
Contains, for each member generated by a user's -ADDR member-level directive:

The name of the member

The format specification of the member (records)

A list containing the stored data in each of the member's records, presented in either:

Card-image (10A8) form if member is type CI (card image)

Octal code, five internal computer words per line, if member is type UNFORMATTED or FORMATTED

**CHANGE member section:**

Included in report if LIST= selection contains letter C

Contains, for each member generated by a user's -CHANGE member-level directive (and its subordinate record-level directives), the same information as presented on ADDR member-section reported members

The typical ANOPP user will build data units via the UPDATE CS specifying either an existing run-life member or a set of input stream records as the only source of data for each new member generated. Since the -ADDR member-level directive will be their primary tool, these users may wish to skip the documentation of the other directives. The large ANOPP application user requires the capability to perform record-level member modification and generation. He will use the -COPY, -OMIT, and -CHANGE member-level directives and the subordinate -INSERT, -DELETE, and -QUIT record-level directives.

The UPDATE control statement initiates the ANOPP update processing phase, during which new resident members are generated onto a user-specified run-life data unit. The UPDATE CS requires a set of subordinate update directives, read either from within the user's primary input stream or from a specified card-image run-life member. The update directives allow the user to copy specific old members from run-life units onto the new unit; to add new members consisting of records read from within the user's update directive set; and to build new members by using selective copying, insertion, and omission of records residing on an old run-life member and/or new user input records. At the initiation of the update processing phase, the specified run-life unit being built (or rebuilt) is wiped clean.

Format:

[label<sub>a</sub>]UPDATE<sub>a</sub>[OLDU=pdu<sub>a</sub>]NEWU=ndu<sub>a</sub>[ALL<sub>a</sub>]SOURCE= {d(mn)}<sub>a</sub>[LIST=x...x] \$

label      label name

OLDU=pdu (optional) keyword specifies the name of a run-life data unit (pdu) as the primary source of member and record data for the new member ndu being generated

If specified, the update revise mode will be in effect: the -COPY and -OMIT update member-level directives may be used; the UPDATE CS keyword parameter ALL may be specified; and record-source data unit specifications on certain directives may be defaulted to primary unit pdu.

If omitted, the update create mode will be in effect: the -COPY and -OMIT directives may not be used, and the UPDATE CS keyword parameter ALL is ignored by the system.

**NEWU=ndu** required keyword parameter specifies the name (ndu) of the run-life data unit onto which new members are to be generated. Any members resident on the unit will be destroyed at initiation of the update processing phase.

**ALL** (optional) keyword is meaningful only in update revise mode processing.

If specified under this mode, the final-member generation step performed in the processing phase will be a copy to the new unit of every member resident on the specified primary old data unit (OLDU=pdu), except for:

A member that was specified as the data-source member via the OLDLM= keyword parameter on a -ADDR or -CHANGE directive

A member that was specified on a -COPY or -OMIT directive

A member whose name matches the name of a new member generated on the new unit via a -ADDR or -CHANGE directive

**SOURCE=** this keyword specification identifies where the card-image update directive set resides

**SOURCE=\*** is legal only on an UPDATE CS occurring in the user's primary input stream. It indicates that the directive-set card images follow immediately behind the UPDATE CS. Directive-set termination is indicated by the occurrence of an END\* input terminator card.

**SOURCE=dun(dmn)** is a legal specification on any UPDATE CS. It indicates that the update directive set resides on run-life unit dun (card-image member dmn).

**LIST=x...x** (optional) value assigned must be a list from one to four alpha characters. Each valid character in the specified list results in a specific update report section being printed. If LIST= is omitted, the update report will contain only the header section. The valid list characters and the report section they generate are:

E Directive echo section  
S Summary section



C CHANGE member section  
 A ADDR member section

Examples:

The following run-life unit U1 will contain the new members generated in accordance with the update directive set. Since no primary data-source unit was specified (OLDU=), only create mode valid directives may be used. An update processing report will be printed which contains all sections except for the directive echo section.

```

LAB1 UPDATE NEWU=U1, SOURCE=*, LIST=SCA $
  -ADDR.....} member-level directives }
  -CHANGE.....}                       } Primary input stream
  -INSERT... } record-level directives } update directive set
  -DELETE... }
  -QUIT..... }
END* $

```

The following run-life unit NEWFAN will contain the new members generated in accordance with the card-image directive set residing on unit U7(member MEM9). The update revise mode is in effect with run-life unit FANU1 identified as the primary data-source unit; all update directives may be used. The ALL keyword specification will result in all unit FANU1 members that were not specified on any member-level directive being copied to unit NEWFAN. The omission of the LIST= keyword will result in an update report consisting only of the header section.

```
UPDATE OLDU=FANU1 NEWU=NEWFAN ALL SOURCE=U7(MEM9) $
```

The general format of update member-level directives is

```
directivenamed fieldd.....fieldd $
```

The valid directivenames are as follows:

- COPY (valid in update revise mode only)
- OMIT (valid in update revise mode only)
- ADDR (valid in all update processing modes)
- CHANGE (valid in all update processing modes)

The fields are as specified in the following subsection that document the individual directives. Labels are illegal on member-level directives.

### 3.3.2.24.1 -COPY Directive

Purpose: The -COPY update member-level directive will generate one or more user-specified new unit members. Each new member is an exact copy of the member of the same name residing on the primary old unit (identified by the UPDATE CS keyword OLDU=).

Format: -COPY<sub>mn</sub>[...<sub>mn</sub>] \$

**mn** the name of the member to be generated on the new unit that will be an exact copy of the primary old-unit (OLDU=) resident member of the same name

#### Examples:

```
-COPY LJETO23, LJET127, LJET219 $
-COPY LJETO19 $
```

#### Restrictions:

Label fields are not valid on member-level directives.

The -COPY directive is valid in update revise mode processing only.

Each specified new member name (mn) must be unique as to all other members generated on the new unit via any member-level directive in the same update directive set.

### 3.3.2.24.2 -OMIT Directive

Purpose: The -OMIT member-level directive is valid only during update revise mode processing initiated by an UPDATE CS on which the keyword "ALL" was specified. The -OMIT directive is used to specify the names of one or more primary old-unit (UPDATE CS parameter OLDU=) resident members that are not to be copied to the new unit during the final revise mode processing step.

Format: -OMIT<sub>mn<sub>1</sub></sub>[...<sub>mn<sub>n</sub></sub>] \$

**mn** the name of a member residing on the UPDATE CS specified primary old unit (OLDU=) that is to be omitted from the new unit (NEWU=) during update revise mode "ALL" processing

#### Examples:

```
-OMIT LJETO24, LJET126, LJET220 $
-OMIT LJETO20 $
```

#### Restrictions:

Label fields are not valid on member-level directives.

The -OMIT directive is valid only in a revise mode processing phase initiated by an UPDATE CS on which the keyword "ALL" was specified.

If the same OLDU resident member is specified on both a -COPY directive and -OMIT directive in the same set, the specified member will be copied to the new unit.

### 3.3.2.24.3 -ADDR Directive

Purpose: The -ADDR update member-level directive allows the user to define and generate a member on the new unit being built under any update processing mode. The record content of the new member may be copied from any specified run-life member (format 1 below) or read from the card images that follow the -ADDR in the user's primary input stream (format 2 below).

When the record data are read from primary input stream card images, the user may optionally define the format specification and maximum number of records for the new member.

Format 1: (valid within a primary or secondary input stream)

$$-ADDR_{\alpha} OLDLM=\left\{ \begin{array}{l} pnm \\ du(mn) \end{array} \right\} [_{\alpha} NEWM=nmn] \$$$

**OLDLM=** required keyword specification of the run-life member whose records are to be copied to the new unit member being generated. Also, the name of the new member if optional keyword **NEWM=** is omitted

**OLDLM=pnm** is a valid specification only in update revise mode processing. The specified pnm must be the name of the member residing on the primary old unit identified by UPDATE CS keyword **OLDU=**.

**OLDLM=du(mn)** is a specification of the run-life data unit name (member name) whose records are to be copied to the new member. (This is valid in any update processing mode.)

**NEWM=nmn** (optional) specifies the name of the new member nm n being generated on the new unit. If omitted, nm n equals the name of the copied member, regardless of its unit residence.

#### Format 1 examples:

The following revise mode must be in effect since the run-life residence of old member JETMEM1 is assumed to be the primary old unit specified by UPDATE CS keyword **OLDU=**. The new member will be named JETO08 and will be an exact copy of member JETMEM1.

```
-ADDR OLDLM=JETMEM1, NEWM=JETO08 $
```

The following specification is valid in any update processing mode. The record content of member MEM8, resident on run-life unit UNIT7, will be

copied to the new unit member being generated. The new member will also be named MEM8.

**-ADDR OLD=UNIT7(MEM8) \$**

Format 1 restrictions:

The name of the new member, whether specified or assumed, must be unique on the new unit.

Label fields are illegal on member-level directives.

Format 2: (valid within a user's primary input stream only)

**-ADDR OLD=\*NEW=mn[FORMAT=format][MNR=n] \$**

**OLD=\*** this specification indicates that the record data for new member mn is to be read from the card images immediately following the -ADDR in the user's update directive set; the record data set is terminated by the next member-level directive read or by the END\* card that terminates the update directive set in the user's primary input stream. The record card-image data must correspond to the FORMAT specification supplied or assumed for the new member, as documented below.

**NEW=mn** required specification of the name (mn) of the new member being generated. The name cannot be a duplicate of the names of the other members generated on the new unit.

**FORMAT=** (optional) keyword specification of the format type of new member mn. If omitted, FORMAT=2HCI.

**FORMAT=0** indicates that the records are of undefined format and variable length (unformatted). Each data field on the card images will be interpreted as an ANOPP data type. The internally converted data will be written to the unformatted records. The occurrence of a \$ on a card image terminates and is not part of the data record being built; the data for the next record are assumed to begin on the following card image.

**FORMAT=2HCI** (default) indicates that a card-image member is to be generated. Each image following the -ADDR, up to but not including the next member-level directive or END\* card, will be a record on the new member.

**FORMAT=nHc<sub>1</sub>,...,c<sub>n-1</sub>** \$ indicates a fixed or variable length format specification for the new member. The input data card images must contain value expressions that correspond in type and sequence with the specified format. The occurrence of a \$ terminates and is not part of the data record being generated. The next input record's data are assumed to begin on the next input card image.

generated. The next input record's data are assumed to begin on the next input card image.

**MNR=n** (optional) integer specification of the maximum number of records that new member mn may contain. If omitted, the system default of MNR=10000 is used for the new member.

Format 2 examples:

New unit member JETM8 will be generated. Its format will be default-type card image (CI). It will contain three records, each consisting of one of the subsequent card images. Member JETM8, which may contain a maximum of 10 records, is given as follows:

```
-ADDR OLDM=* NEWM=JETM8 MNR=10 $
RECORD 1 WILL BE THIS CARD-IMAGE
RECORD 2 " " " " " "
RECORD 3 " " " " " "
(Next update member-level directive or the END* input terminator
card)
```

Member M4 is created as a new unit unformatted member containing three records and having a default maximum number of record value of 10000. The individual records generated will each possess the record length (in internal computer storage words) required to store the converted data values supplied for it. Those values will be:

```
For record 1 - one type RS (real single) value
For record 2 - two type RD (real double precision) values, one type
                 L (logical), one type A3 value (Hollerith string of
                 three characters)
For record 3 - three type I (integer) values
```

Member M4 is given as follows:

```
-ADDR OLDM=*, NEWM=M4, FORMAT=0 $
.693 $ COMMENTS ANYONE?
.70D+01 .898D+20 .TRUE. 3HABC $
10 20 30 $
(Next member-level directive or the END* input terminator card)
```

The following member M5 will be a formatted fixed-length member containing two records and a value of default of 10000 for a maximum number of records:

```
-ADDR OLDM=*, NEWM=M5, FORMAT=17HI,2RS,A9,RD,CS,LS $
1 1.5 2E+01 9HJET-STATS 3D-01 (1.5,2.0) .TRUE. $
11 977.8 -.696E-110 9HPROP-STAT 1075.5D+01 (.696E+29
.32E+31) .FALSE. $
(Next member-level directive or the END* input terminator card)
```

The following member COORD contains the X, Y, Z needed for the geometry module:

```
-ADDR OLDM=*, NEWM=COORD, FORMAT=4H3RSS $
0. 0. 4. $
END*
```

Format 2 restrictions:

The ADDR keyword specification OLDM=\* is valid only within a user's primary input stream.

The name of the new member (NEWM=) must be unique on the new unit being built.

```
*****
*
*                               NOTE
*
*****
*
*   The remaining directives in this section are intended for large-
*   application users who will establish and maintain large data base
*   applications.
*
*****
```

3.3.2.24.4 -CHANGE Directive

Purpose: The -CHANGE update member-level directive allows for the generation of a new unit member, whose records will be selectively copied from a user-specified old run-life member, with optional insertion of additional new records read from within the user's card-image update directive set. The sequence and source of new member records to be generated are in accordance with a set of card-image record-level directives subordinate to the -CHANGE directive.

The new member will possess the same format as does the specified old member; the name and maximum-number-of-records value of the new member may optionally be changed from that of the old member.

A set of record-level directives must immediately follow the -CHANGE directive it supports in the update directive set. The occurrence of a subsequent member-level directive, or the termination of the user's update directive set, ends the -CHANGE subordinate record-level set.

Format: -CHANGE<sub>d</sub> OLDM= $\left\{ \begin{matrix} \text{pnn} \\ \text{du}(\text{mn}) \end{matrix} \right\}$  [<sub>d</sub>NEWM=nmn] [<sub>d</sub>MNR=n] \$

**OLDM=** specifies the run-life member whose resident records will be selectively copied onto the new member in accordance with the record-level directives set

**OLDM=pmn** is valid only in update revise-mode processing. The specified old member (pmn) must be resident on the primary old unit identified via the UPDATE CS keyword **OLDU=** specification.

**OLDM=du(mn)** is valid in any update processing mode and specifies the run-life data unit name (member name) of the old member.

**NEWM=nmn** (optional) specifies the name of the new unit member being generated. If omitted, the name of the specified old member (**OLDM=**) will also be assigned to the new member.

**MNR=n** (optional) specifies the maximum number of records for integer n that the new member may contain. If omitted, the system will calculate the new member's MNR value based on the old member's MNR value and current number of records.

Examples:

Since the run-life data unit residence of old member M1 was not specified, this must be a revise-mode update, initiated by an UPDATE CS on which a primary old unit (**OLDU=**) was specified. In accordance with the record-level directive set, the following new unit member JETM1 will contain records selectively read from primary old-unit member M1 and, optionally, records read from within the record-level directive set:

```
-CHANGE OLDLM=M1,NEWM=JETM1 $
:
: (Record-level directives)
:
: (Member-level directive or the end of the user's update directive
: set)
```

The following new unit member MEM2 will be created with an assigned value of 2000 for maximum number of records. In accordance with the record-level directive set, records will be selectively copied to the new member from run-life unit FANU member MEM2.

```
-CHANGE OLDLM=FANU(MEM2) MNR=2000 $
:
: (Record-level directives)
:
: (Member-level directive)
```

The general format for -CHANGE subordinate record-level directives is

directivename[<sub>i</sub>field ...<sub>j</sub>field] \$

The valid directivenames are

-INSERT      -DELETE      -QUIT

The fields are as specified in the following subsections that document the individual directives. Labels are illegal on record-level directives. The record-level directives are processed under control of the preceding -CHANGE member-level directive. The -CHANGE directive, documented in the previous section, performs the following:

Identifies the new unit member whose record content will be a changed copy of the current content of a run-life old member

Identifies the old member (OLDM=)

Remains in control until each of its subordinate record-level directives have been processed

Each directive generates the copying of single or groups of records either from the old member or from within the directive set. Throughout a record-level directive set, the integer operands *i* and *j* are used as pointers to relative record positions in the old member. Directives within a set must be processed (and positioned) sequentially with respect to any *i* and *j* operand specifications they possess. This is necessary because the new member is generated in one pass.

The system monitors an old member reference pointer *P* throughout the generation of a new member via a -CHANGE and its record-level directive set. Here, *P* is initiated at zero. Upon completion of each directive's processing, *P* assumes one of the following values:

- (1) The *j* value (if specified on the directive)
- (2) The *i* value (if specified, and *j* value not specified)
- (3) Unchanged (if *i* value not specified)

The next record-level directive must have an *i* value (if present) greater than the new value of *P*.

The *i* operand on a directive specifies the relative record position of the first record in a sequence of one or more old-member records. Depending



on the directive, this single or sequence of old-member records will be treated as follows:

Copied to the new member, optionally to be followed by inserted records read from the card images immediately following the directive or from elsewhere in the old member

Omitted from copying to the new member

The  $j$  operand, optionally used on the -DELETE directive, is used to specify the last old-member record in a series to be omitted from copying to the new member. The  $i$  value (if present) on any directive must be greater than the  $i$  (and  $j$ , if present) value specified on a preceding directive in the set. For example, a directive referencing old-member records 5 through 8 ( $i_1$  through  $j_1$ ) must be processed before one referencing 9 through 10 ( $i_2$  through  $j_2$ ), and never vice versa.

The -QUIT directive, if encountered, terminates the new-member generation process and the -CHANGE directive's subordinate set. It may specify that some, all, or none of the remaining old-member records be copied to the new member as a final generation step. If -QUIT is not used, the set terminates with the occurrence of another update member-level directive or with the termination of the update directive set. Deletion (or omission) of old-member records from the new-member copy is performed via the -DELETE directive. Insertion of new records read from directive set card images is performed via the -INSERT directive. Resequencing of old-member records onto the new-member records is performed via the -INSERT directive.

#### -INSERT Record-Level Directive

Purpose: The -INSERT record-level directive allows the user to insert onto the new member a group of one or more sequentially positioned records, either currently residing on the old member or immediately following the directive card image in the record-level directive set.

Optionally, the user may specify that a group of one or more old-member records will be copied onto the new member to precede the inserted records.

Format: -INSERT <sub>$P$</sub> [ $i$ ] <sub>$P$</sub>  [FROM={OLDM} <sub>$P$</sub> ][[ $m$ , $n$ ]] \$

$i$  (optional) the integer relative position of the last old-member record to be copied to the new member prior to the specified records to be inserted. Integer  $i$  must be greater than old-member record pointer  $P$ . Records  $P + 1$  through and including  $i$  will be copied to the new member.

If omitted,  $P$  remains unchanged, allowing records to be inserted at any position on the new member, including at the beginning when  $P = 0$ .

**FROM=** (optional) specifies the source of the records to be inserted onto the new member. If omitted, FROM=\* is assumed.

**FROM=\*** (default) indicates that the records to be inserted on the new member immediately follow in the directive-set card images. The new record card images are terminated by the next record-level directive, member-level directive, or termination of the directive set. FROM=\* is valid in any CS stream. The new record data must correspond to the format specification of the -CHANGE specified old member.

If the format of OLDM is unformatted, fixed, or variable, then a \$ must terminate the data of each input record and the next record must begin on a separate card image. Applicable examples are presented below and in the following subsections.

**FROM=OLDM** indicates that the records to be inserted on the new member reside on the -CHANGE specified OLDM and are specified by the next operand field.

**(m[,n])** (optional) specifies which old-member record (m) or sequence of old-member records (m,n) are to be inserted on the new member. This field specification has no effect on old-member record pointer P. The integer value of m or n may be greater, less than, or equal to an i-parameter specification; n, when specified, must be greater than m. If FROM=\*, this field is ignored. If omitted, and FROM=OLDM, all OLDM records will be inserted on the new member.

Examples:

Example 1: Assume that N1 is a card-image formatted member on the UPDATE CS specified primary old data unit. Assume that the user desires to insert two new cards following the sixth card-image record of the old member. The following would copy the first six records onto the new member and then insert the two new ones:

```
-CHANGE OLDM=N1 $
-INSERT 6 $
  (Card image)
  (Card image)
  (Next member-level directive)
```

Example 2: The following directive would copy the first six old-member records to the new member, to be followed by an insertion of a second copy of old-member records 1 through 3. Following this process, P would equal 6.

```
-INSERT 6, FROM=OLDM (1,3) $
```

Example 3: The following is a second example of the -INSERT capability to reorganize old-member records onto the new member. The old member contains 30 records, consisting of 3 functionally related groups of 10 records each. Assume that the user wishes to retain all the records but intends to

position the second related group (old-member records 11 through 20) at the beginning of the new member. The following would perform the desired resequencing:

```
-INSERT FROM=OLDM (11,20) $           Copy records 11 to 20.
-INSERT 10 FROM OLDM (21,30) $       Copy records 1 to 10.
                                       Copy records 21 to 30.
```

Example 4: For the following example, assume that the OLDU data unit is UN1 and the user wishes to create a new unit member based on run-life unit UN3 member MEM7. MEM7 has a fixed-length format specification of FORMAT=6H3I,3L\$. It contains 10 records, all to be retained on the new member. The user intends to insert from within his directive set one new record behind old-member records 5 and 10. The new member will retain the name MEM7.

```
-CHANGE OLDM=UN3(MEM7) $
-INSERT 5 FROM=* $
  2765 8329 10754 .TRUE. .TRUE. .FALSE. $
-INSERT 10 FROM=* $
  6439 10376 7420 .TRUE. .FALSE. .FALSE. $
(Next member-level directive)
```

NOTE: Additional -INSERT directive examples are included in the following subsections presenting the -DELETE and -OMIT record-level directives.

#### Restrictions:

New data records must correspond to the member format of OLD M.

A value specification for operand i must be greater than that specified for operands i or j on any preceding record-level directive in the set.

#### -DELETE Record-Level Directive

Purpose: The -DELETE member-level directive allows the user to bypass old-member records that are not to be copied to the new member being generated. Optionally, the user may cause a preceding group of old-member records to be copied to the new member.

Format: -DELETE<sub>i</sub>,i[,j] \$

i[,j] the integer operands i and j specify a range of old-member positioned records that are not to be copied to the new member. The i must be greater than the current value of old-member pointer P; and j, if specified, must be greater than i.

If P + 1 is less than i, old-member records P + 1 through i - 1 are first copied to the new member.

Pointer P is set to the value of j, or to the value of i if j is not specified.

Example:

Assume that old-member JET1 resides on the primary old data unit and contains 20 records. Old-member records 9, 10, 15, 16, and 17 are to be deleted. New records are to be inserted following current old-member records 5 and 12 on the new member. In addition, current old-member records 2 and 3 are to be moved to the end of the records on the new member.

-CHANGE OLDM=JET1 \$	At initiation, old-member record pointer is initiated. (P=0)
-DELETE 2,3 \$	Old-member record 1 is copied. (P=3)
-INSERT 5 FROM=* \$ (New record N1 card images) (New record N2 card images)	Old records 4 and 5 are copied. New records N1 and N2 are inserted. (P=5)
-DELETE 9,10 \$	Old records 6 through 8 are copied. (P=10)
-INSERT 12 FROM=* \$ (New record N3 card images)	Old records 11 and 12 are copied. New record N3 is inserted. (P=12)
-DELETE 15,17 \$	Old records 13 and 14 are copied. (P=17)
-INSERT 20 FROM=OLDM (2,3) \$	Old records 18, 19, and 20 are copied. Old records 2 and 3 are then copied. (P=20)

(Next member-level directive)

The new member would now contain:

OLDM Rec	1
" "	4
" "	5
New Rec	N1
" "	N2
OLDM Rec	6
" "	7
" "	8
" "	11
" "	12
New Rec	N3
OLDM Rec	13
" "	14
" "	18
" "	19
" "	20
" "	2
" "	3

## -QUIT Record-Level Directive

Purpose: The -QUIT record-level directive allows the user to terminate the generation of the -CHANGE identified new member, optionally specifying the final old-member record to be copied to the new member.

Format: -QUIT<sub>a</sub>[i] \$

i (optional) specifies the integer relative position of the old-member record that will be the last record copied to the new member. If specified, i must be greater than old-member record pointer P. Records P + 1 through and including record i will be copied to the new member. If omitted, processing of the new-member generation is terminated immediately.

### Examples:

Assume for both subsequent examples that the old member contains 20 records and that the preceding record-level directive processed was the following:

```
-INSERT 15 FROM=* $  
(New record N1 card images)
```

Example 1: The following would result in termination of the -CHANGE-initiated new-member generation. The last new-member record will be N1.

```
-QUIT $
```

Example 2: The following would result in old-member records 16 and 17 being the last records to be generated on the new member:

```
-QUIT 17 $
```

### Restrictions:

The -QUIT directive, if present, must be the last record-level directive under the control of the -CHANGE directive.

A value specification of operand i must be greater than that specified for operand i or j on any preceding record-level directive in the set.

### Examples

The following update processing phase examples incorporate all of the member-level and record-level directives. It is assumed that each data source unit or new unit identified was established in the run-life data base via a preceding CREATE CS, ATTACH CS, or LOAD CS.

Example 1: A revised mode update is to be performed by using run-life unit OLDJET as the primary old unit. Unit OLDJET contains 20 members named MEM1 through MEM20.

Unit NEWJET is the run-life unit to be built (or rebuilt). NEWJET is to receive the following members:

MEM1 through MEM6	copied unchanged from unit OLDJET
MEM7	rebuilt, using OLDJET Unit MEM7 as its primary source member. MEM7 contains eight records; records 6 and 7 are to be replaced by user input records.
MEM8	to be generated by using new input records
MEM9	to be a renamed copy of run-life unit UNITFAN resident member UFMEM1
MEM10 through MEM15	copied unchanged from unit OLDJET
MEM16	to be a rebuilt version of run-life unit UNITFAN resident member UFMEM8. Member UFMEM8 consists of 10 records; records 4 and 9 are not to be copied to new member MEM16.
MEM17 through MEM20	copied unchanged from unit OLDJET

Thus,

```

UPDATE OLDU=OLDJET NEWU=NEWJET SOURCE=* LIST=ESCA $
  -COPY MEM1, MEM2, MEM3, MEM4, MEM5, MEM6 $
  -CHANGE OLDM=MEM7 $
    -INSERT 5 FROM=* $
      .TRUE. .TRUE. 17.65D27 7278 (7.53 1E320) $
      .FALSE. .FALSE. -1D+322 6920 (375.8E-12,270E+12) $
    -DELETE 6,7 $
    -QUIT 8 $
  -ADDR OLDM=* NEWM=MEM8 FORMAT=8H2I,L*RS$ MNR=10 $
    757 9329 .TRUE. 17.97 -3.18E03 $
    14 723 .TRUE. 1E216 275E+10 18.327 $
    1278 87 .FALSE. $
  -ADDR OLDM=UNITFAN(UFMEM1), NEWM=MEM9 $
  -COPY MEM10, MEM11, MEM12, MEM13, MEM14, MEM15 $
  -CHANGE OLDM=UNITFAN(UFMEM8) NEWM=MEM16 $
    -DELETE 4 $
    -DELETE 9 $
    -QUIT 10 $
  -COPY MEM17, MEM18, MEM19, MEM20 $
END* $

```

Example 2: A revised mode update is to be performed by using run-life unit STAT1 as the primary old unit. STAT1 contains 15 members named MEM1 through MEM15.

Unit NEWSTAT is the run-life unit to be built and is to receive the following members:

MEM1 through MEM4           copied unchanged from unit STAT1

MEM5                       rebuilt by using system unit DATA member MEMB as the primary source member. MEMB is a card-image formatted member containing seven cards; cards 2 and 7 are to be replaced by user input records.

MEM6                       to be generated by using new input records

MEM7 and MEM8             renamed copies of unit STAT1 members MEM14 and MEM15

MEM9 through MEM13       copied unchanged from unit STAT1

Thus,

```
UPDATE OLDU=STAT1 NEWU=NEWSTAT ALL SOURCE=* $
  -CHANGE OLDM=DATA(MEMB) NEWM=MEM5 $
    -INSERT 1 FROM=* $
    THIS CARD IMAGE WILL BE NEW MEM5 RECORD 2
    -DELETE 2 $
    -INSERT 6 FROM=* $
    THIS CARD IMAGE WILL BE NEW MEM5 RECORD 7
  -ADDR OLDM=* NEWM=MEM6 FORMAT=16HA7,L,I,RS,RD,CSS $
    6HHEIGHT .TRUE. 89 + 7.93 17.16D27 (9.75,1E279) $
    5HSPEED .FALSE. 94757 -1.33E05 1.36D-03 (-1.7E02 0.92) $
  -OMIT MEM14, MEM15 $
  -ADDR OLDM=MEM14 NEWM=MEM7 $
  -ADDR OLDM=MEM15 NEWM=MEM8 $
END* $
```

NOTE: The ALL keyword specification on the UPDATE CS above results in all primary old-unit STAT1 members that were not specified on a -OMIT directive, and that do not possess a name that is the same as a new unit member generated via a -CHANGE or -ADDR directive, being copied unchanged to the new unit NEWSTAT during the final update processing phase step.

### 3.3.2.25 Conditional Processing CS

#### Overview

The normal processing flow occurring in an ANOPP run is the sequential execution of each control statement in the user's input streams. Following execution of the last CS in a called secondary stream, the system executes the control statement that follows the CALL CS in the calling stream. Execution of the ENDCS control statement, marking the end of the user's primary input stream, terminates the ANOPP run.

The user, however, has the capability to deviate from normal sequential CS processing through error handling and branch control statements.

Controlling subsequent CS processing following the occurrence of a system "nonfatal" error.- Prior to execution of each control statement, the system sets an internal nonfatal error flag to "off." If during the execution of the CS or FM an abnormal but nonfatal condition occurs, the flag is set to "on." Documentation of error diagnostics of specific functional module and ANOPP system identifies conditions that result in nonfatal errors.

Following the execution of a functional module or a control statement, the system performs this sequence of steps:

Test the value of the internal nonfatal error flag.

If "off," continue processing with next CS in stream.

If "on" (error occurred), test value of system parameter JCON:

If JCON=.TRUE. (set by user via SETSYS CS), continue processing with next CS in stream.

If JCON=.FALSE. (default value), search forward sequentially seeking a PROCEED CS:

If a PROCEED CS is found, continue processing with the execution of the CS following it.

If the ENDCS ending the user's primary input stream is found first, terminate the job.

If, at the time of the error, control was within a secondary input stream and JCON=.FALSE., the search for a PROCEED CS will continue through the rest of the secondary stream. The search continues with the remainder of the calling stream beginning with the CS following the CALL CS until either a PROCEED CS or the run terminating ENDCS is found.

Determining subsequent CS processing flow based on the current value of a user parameter.- The IF control statement allows the user to specify a logical condition of several types. The user also specifies a name that must be the label field value of another control statement in the same input stream. If the specified condition test is .TRUE., processing will continue with the execution of the CS having the specified label name; if the condition test is .FALSE., processing continues normally with the next sequential CS in the stream.

The GOTO CS performs the single function of transferring processing control unconditionally to the control statement having a specific label name. The CONTINUE CS is a control statement that may contain a label-name field and is used as the object of IF CS or GOTO CS processing, but it causes no other action to occur.

#### 3.3.2.25.1 GOTO CS

Purpose: The GOTO control statement allows for an unconditional deviation from the sequential processing within an input stream. Processing will continue with the execution of the control statement having the label-field value specified.



Format: [label<sub>d</sub>]GOTO<sub>d</sub>labname \$

**label** the optional label name of this control statement

**labname** the required label name of another control statement within the same input stream where processing will continue

Examples:

```
ENDST1 GOTO STEP3 $
GOTO STEP2 $
```

Restriction: Labname must be the label name of a control statement within the same input stream as the GOTO CS.

### 3.3.2.25.2 IF CS

Purpose: The IF control statement permits an alteration in the flow of processing if a specified condition exists. The conditional statement is specified within parentheses. In the first form, the value of a user parameter (or parameter array) is logically compared with the value of either another user parameter (array) or user-specified constant(s). In the second form, the existence or nonexistence of a unit member is determined. If the condition within the parentheses is true, then processing continues with the control statement having the specified label; otherwise processing continues with the next control statement.

Formats:

$$\text{label}_d \text{IF} \left( \text{paramname}_1 \text{ } \underset{d}{\text{operator}} \left( \underset{d}{\text{paramname}_2} \underset{d}{\text{value}} \right) \right) \text{GOTO}_d \text{labnam} \$$$
$$\text{label}_d \text{IF} \left( \text{unit}(\text{mem}) \left( \begin{array}{c} \text{AS} \\ \text{NAS} \end{array} \right) \right) \text{GOTO}_d \text{labnam} \$$$

**label** label name

**paramname<sub>1</sub>** name of a user parameter whose value(s) is to be compared with the value(s) following the logical operator

**logical operator** a logical operator used in comparison of the two values. Any logical operator is valid for comparing values which are type integer, real single precision, or real double precision. The operators .EQ. and .NE. are the only valid operators for logical, complex single, and character string data types. For the .NE. operator, the results of element comparisons are combined with an OR condition. For all other operators, the results of element comparisons are combined with an AND condition. For the .NE. operator, only one element of the first array need be unequal to the corresponding element of the second array in order for the IF CS to yield a true

condition. However, for all other operators, every element of the first array compared to corresponding elements of the second array must yield a true condition in order for the IF CS to yield a true condition.

**value** array of one or more elements of type numerical constant (I, RS, RD, CS), logical constant, or string constant. All elements must be of the same type.

**paramname<sub>2</sub>** name of an existing user parameter array to be used in comparison

**unit** name of data unit on which data member, mem, resides

**mem** name of data member

**AS** flag indicating condition is true if unit (mem) exists. If unit (mem) exists as a data unit member and flag AS is indicated, a true condition results. If unit (mem) does not exist and AS is indicated, a false condition results.

**NAS** flag indicating condition is true if unit (mem) does not exist. If unit (mem) does not exist as a data unit member and flag NAS is indicated, a true condition results. If unit (mem) exists and NAS is specified, a false condition results.

**labnam** the label of the control statement at which processing should continue if the comparison of the two values results in a true condition.

Examples:

```
LABEL1  IF (A .GE. B) GOTO LABEL2 $
        IF (D .EQ. .TRUE.) GOTO LABEL1 $
        IF (F .EQ. 6HFVALUE) GOTO LABEL2 $
        IF (G .GT. 10 20 30) GOTO LABEL1 $
        IF (H .EQ. .FALSE. .TRUE.) GOTO LABEL1 $
        IF (PROC (SOURCE) AS) GOTO LABEL3 $
        IF (PROC (SOURCE) NAS) GOTO LABEL2 $
        .
        .
LABEL2  CONTINUE $
        .
        .
LABEL3  CONTINUE $
```

Restrictions:

Labname must be in the label field of a control statement within the same input stream as the IF CS.

The data type of the second value array must agree with the value type of paramname<sub>1</sub>.

When comparing string constants of unequal length, trailing blanks are ignored.

The number of elements of the second-value array should agree with the number of elements of paramname<sub>1</sub>; if not, the condition is assumed false.

### 3.3.2.25.3 CONTINUE CS

Purpose: The CONTINUE control statement provides a no-action step within the input stream. It is used mainly with the IF CS and/or GOTO CS to allow for an alteration in the flow of control-stream processing. The processing continues with the execution of the CS following the CONTINUE CS.

Format: [label<sub>1</sub>]CONTINUE \$

label label field

Example:

LABEL1 CONTINUE \$

Restriction: None.

### 3.3.2.25.4 PROCEED CS

Purpose: The PROCEED control statement marks the location within the input stream where ANOPP execution will continue if a nonfatal error condition occurs during the processing of a preceding CS or functional module and if the current value of system parameter JCON is .FALSE.. The PROCEED CS is a no-operation statement when encountered during normal sequential CS processing.

Format:

[label<sub>1</sub>]PROCEED \$

label label field

Examples

ERROR1 PROCEED \$  
PROCEED \$

Restrictions:

During the forward search in the input stream of the system for a PROCEED CS, any CALL statements encountered are not processed. (The procedure-member resident input stream is not searched or expanded.)

If the nonfatal error condition initiating the search for the PROCEED CS occurred during processing within a CALL'ed secondary input stream, and no PROCEED CS is found to exist in the remainder of that stream, the search will continue in the input stream that contained the CALL, beginning with the control statement following the CALL CS.

If the ENDCS control statement ending the user's primary input stream is found before any PROCEED CS, the run will be terminated.

### Examples

The following examples illustrate conditional processing flow within input streams.

Example 1: The following input stream illustrates the processing flow that will occur if a system nonfatal error occurs during the execution of a functional module:

```
STARTCS $
:
SETSYS JCON=.TRUE. $
EXECUTE FM1 $
SETSYS JCON=.FALSE. $
EXECUTE FM2 $
ATTACH FM3UNIT/EFN3/ $
EXECUTE FM3 $
DETACH FM3UNIT $
GOTO LASTSTEP $
PROCEED $
ATTACH FM4UNIT/EFN4/ $
EXECUTE FM4 $
DETACH FM4UNIT $
LASTSTEP EXECUTE FM5 $
:
ENDCS $
```

Since system parameter JCON equals true when functional module FM1 is executed, processing will continue with the SETSYS CS that follows it, regardless if a nonfatal error occurs during FM1 execution. However, JCON=.FALSE. when functional module FM2 is executed. If no nonfatal error occurs in FM2, processing continues with the following ATTACH CS. If a nonfatal error does occur in FM2, processing will continue with the ATTACH CS following the PROCEED CS.

Example 2: The following input stream illustrates conditional processing flow that occurs following the test of a logical condition based on the value of a user parameter:

```
STARTCS $
:
PARAM EXCOUNT=0 $
REPEAT PARAM COUNT=EXCOUNT + 1 $
EXECUTE JET $
PARAM EXCOUNT=COUNT $
IF (EXCOUNT .NE. 5) GOTO REPEAT $
IF (JETRSLT .EQ. .TRUE.) GOTO JETT $
EXECUTE JET9 $
GOTO ENDJET $
JETT EXECUTE JET7 $
ENDJET CONTINUE $
```

### 3.3.2.26 Checkpoint/Restart CS

#### Overview

The ANOPP system provides the capability to establish checkpoints within an input stream(s) of a run and to restart that run in a subsequent job without reprocessing the input stream statements that preceded a specific checkpoint in the initial run. The intent of checkpoint/restart is to save the run environment in the event of either a planned or unplanned interruption in an ANOPP execution. This capability may be used to protect against system crashes or to halt a run's execution for examination of intermediate data without losing execution time or data base characteristics. Depending on the results of that examination, the user may specify that the operating environment that existed at any specific checkpoint be reestablished and execution continued.

The checkpoint run.- A checkpoint run is any ANOPP run during which a CKPNT CS is executed. During the execution of CKPNT CS, the system writes to a checkpoint file the necessary internal system and run-life data base information required to recreate this run's current operating environment for use in a subsequent restart run.

The checkpoint/restart file.- During each checkpoint run the user must ensure that an external sequential checkpoint file is assigned to the job. This checkpoint file must be permanently retained in the external system following job termination if it is to be used as the restart file in an ANOPP restart run application. The checkpoint/restart file may reside on tape or rotating mass storage.

The checkpoint/restart file structure is divided into primary divisions called cycles. Each cycle is referenced by the integer number of its sequential position on the file. Cycle 1 would be the first cycle written to the file resulting from the first CKPNT CS executed in the checkpoint run. Any subsequent CKPNT control statement execution in the run would generate another cycle on the checkpoint/restart file. Whenever a checkpoint-file cycle is generated during a run, a report is printed identifying the cycle number, the current run-life data units, and other checkpoint status information that will assist the user in selecting his future restart options. In a subsequent restart run the user specifies at what position (checkpoint) he wishes to restart his run within the initial processing stream of the checkpoint run.

Each cycle of the checkpoint file contains the necessary information to preserve the current operating environment that existed at the time its

generating CKPNT CS was executed. The cycle information contains the following:

- All required internal system tables, including those that contain:
  - User parameter names and values
  - Run-control (initialization and system) parameter values
  - Run-life data base directories

- A copy of each nonarchived run-life unit and its members, including system unit DATA

- A list of the archived run-life units

The external name under which the sequential checkpoint file is assigned to a user's job must also be established within the run. The user may specify the external name of the checkpoint file via the ANOPP CS or RSTRT CS specification of initialization parameter CKPNT. If not specified in this manner, the system uses the default name of CPFILE.

The CKPNT CS.- The CKPNT CS is used to establish a checkpoint within any input stream in a user's run. The CKPNT CS is not automatically executed when encountered during the processing phases of an ANOPP run. The CKPNT CS is treated as inactive (a no-operation CS) if logical system parameter JCKPNT is set to system default value .FALSE. at the time that the CKPNT CS is processed. This allows the user to insert CKPNT control statements at potential restart locations within the input streams and control activation and deactivation via system parameter JCKPNT. A CKPNT CS may contain an optional specification of the keyword STOP. If specified on an executed CKPNT CS, the ANOPP run will be terminated following generation of the checkpoint-file cycle.

System parameter JCKPNT.- The value of JCKPNT controls all checkpoint (CKPNT CS) processing in ANOPP runs. Only when the current value of JCKPNT equals .TRUE. is a CKPNT CS executed, resulting in the generation of a new checkpoint-file cycle. If the user specifies the name of an external checkpoint file via initialization parameter CKPNT on an ANOPP CS or RSTRT CS, the initial value of JCKPNT is set to .TRUE.. The user may also activate and deactivate checkpoint processing during his runs via SETSYS CS specifications of parameter JCKPNT values.

The restart run.- Each ANOPP restart run is initiated via the special RSTRT control statement. The RSTRT CS must be the first card in the run input deck and provides run-initialization capabilities similar to those provided by the optional ANOPP CS in nonrestart runs. The RSTRT CS performs the following:

- Identifies the run as an ANOPP restart application
- Indicates which previous checkpoint run is to be restarted at what checkpoint
- Allows for the change of previous initialization or system parameter values

The optional RSTRT= keyword parameter of the RSTRT control statement is used to specify the name under which the user's external restart file is

assigned to his job. If omitted, the system assumes that the user's restart file has the name RSFILE.

The RSCYCL= keyword parameter of the RSTRT CS is used to specify the integer number of the cycle on the restart file to be used to restart the previous checkpoint run. If omitted, the last cycle generated on the file will be used.

Restart-run initialization.- The initialization phase of a restart run begins when the RSTRT CS is encountered. The RSTRT CS specified restart file and cycle are accessed. The operating environment preserved by the specified cycle is reestablished. The initialization steps are as follows:

(1) System and initialization parameters are assigned their checkpointed values or are assigned new values if specified on the RSTRT CS.

(2) The checkpointed run-life data base is reestablished. Each non-archived data unit and its members will be copied from the restart file to an external file with the same name as the one it resided on at checkpoint time. If the user has an external file assigned to the restart job that matches a file name of a checkpointed unit stored on the restart file, that external file will receive the copy; otherwise, the system will use a scratch (temporary) file of the same name. The system unit DATA is regenerated by using checkpoint cycle copies. No archived data unit file that existed in the checkpointed run-life data base will be regenerated. The user has the responsibility of ensuring that the external files on which archived units reside are assigned to his restart job. The user must ensure that any library files required by subsequent restart-run processing are externally assigned to his job.

(3) The user parameter tables and data unit directory are reestablished with checkpointed values.

(4) System parameter JRSTRT is set to .TRUE., indicating that ANOPP restart run processing is to occur.

Following initialization, the next processing step depends on whether the user has supplied a new primary input stream in his run input deck. If the RSTRT CS was the only card image in the input deck, run processing resumes with the execution of the control statement within the initial-run input stream which followed the CKPNT CS that generated the applicable checkpoint (restart) cycle.

An inserted primary input stream can be processed either prior to or instead of the resumption of checkpointed input stream processing as described above. An inserted primary input stream may not contain a CKPNT control statement.

The value of logical system parameter JRSTRT upon completion of the inserted input stream (ENDCS) processing determines whether the run will terminate or resume processing the checkpointed input stream. The JRSTRT is set to .TRUE. during restart-run initialization. If the user has changed its value to .FALSE., and the ENDCS is processed, the run will be terminated at that point. If JRSTRT is equal to .TRUE. at ENDCS process time, run

processing will continue with the execution of the control statement that follows the applicable CKPNT CS within the checkpointed input streams.

### 3.3.2.26.1 CKPNT CS

**Purpose:** The CKPNT control statement establishes a potential checkpoint within a user's run. A CKPNT CS is executed only when system parameter JCKPNT=.TRUE.; each execution generates a new cycle on a predesignated checkpoint file. The checkpoint cycle contains the internal system and run-life data base information required to preserve the current operating environment for use in subsequent restart runs.

**Format:** [label<sub>a</sub>]CKPNT[<sub>a</sub>STOP] \$

label label name

**STOP** optional keyword. If specified on an executed CKPNT CS, the run will terminate following the generation of the checkpoint file cycle preserving the current-run operating environment.

**Examples:**

LABEL1 CKPNT \$

CKPNT STOP \$

**Restriction:** The CKPNT CS is valid within any input stream except for a primary input stream inserted in an ANOPP restart run.

### 3.3.2.26.2 RSTRT CS

**Purpose:** The RSTRT control statement identifies a restart run and reestablishes a checkpointed operating environment. It identifies, through the specification or omission of keyword parameters, the name of the external restart file and the checkpoint cycle on the file where the operating environment to be reestablished is preserved.

The RSTRT CS also allows the user to specify new values for initialization and system parameters. Table 3.3-3 in this section presents those parameters subject to RSTRT CS specification.

**Format:**

RSTRT[<sub>a</sub>RSTRT=sefn][<sub>a</sub>RSCYCL=cn][<sub>a</sub>paramname<sub>1</sub>=value<sub>1</sub>....<sub>a</sub>paramname<sub>n</sub>=value<sub>n</sub>] \$

**RSTRT=sefn** optional keyword specification of the external file name (sefn) by which the restart file must be currently assigned to the user's job. If omitted, RSTRT=RSFILE is assumed.



**RSCYCL=cn** optional keyword specification of the integer checkpoint cycle number on the restart file from which the operating environment is to be reestablished at the initiation of this run. If omitted, the last cycle on the restart file is used.

**paramname** optional specification of the name of a run-control initialization or system parameter whose checkpointed value is to be changed. (See Table 3.3-3.)

**value** the new value to be assigned the specified paramname at initialization of the restart run

Examples:

```
RSTRT $
RSTRT RSTRT=RFILEIN LENG1=30000 CKPNT=NEWRFLE $
RSTRT RSCYCLE=3, JECHO=.TRUE. $
```

Restrictions:

RSTRT CS is valid only as the first CS and, optionally, the only CS in a restart run input deck.

The specified or assumed external file name (sefn) of the restart file may not be the same as the external file name of any sequential library file or run-life data-unit file used within the run.

Examples

Checkpoint run example 1: For the following example, the run is initialized with checkpoint processing active via the initialization parameter CKPNT specification on the ANOPP CS; system parameter JCKPNT is automatically initialized as .TRUE.; and external sequential file JETFILE is identified as the checkpoint file:

```
ANOPP CKPNT=JETFILE $
STARTCS $
:
CP1 CKPNT $
:
IF (KEYITEM .EQ. .TRUE.) GOTO CP2 $
SETSYS JCKPNT=.FALSE. $
CP2 CKPTN STOP $
:
SETSYS JCKPNT=.TRUE. $
CP3 CKPNT STOP $
:
ENDCS $
```

Table 3.3-3. RSTRT CS Initialization and System Parameters

Parameter name	Parameter type	Description	Value		Default value
			Type	Range	
ACCOUNT	System <sup>a</sup>	Controls accumulation of cost-accounting data associated with each functional module or executive processor  ACCOUNT=.TRUE. - accumulate accounting data ACCOUNT=.FALSE. - do not accumulate accounting data	Logical	.TRUE. .FALSE.	.FALSE.
CKPNT	Initiali- zation	Identifies name of external checkpoint file to be generated during run. Its specification also results in system parameter JCKPNT being initialized as .TRUE., automatically activating checkpoint processing in the run	Alphanumeric name of seven characters or less		CPFILE
JECHO	System <sup>a</sup>	Controls printing of CS card image upon validation in primary edit phase (primary input stream editing) and secondary edit phases (procedure member-editing)  JECHO=.TRUE. - print CS card images JECHO=.FALSE. - do not print CS card images	Logical	.TRUE. .FALSE.	.FALSE.
JLOG	System <sup>a</sup>	Control printing of CS card images upon execution in executive processing phases  JLOG=.TRUE. - print CS card images JLOG=.FALSE. - do not print CS card images	Logical	.TRUE. .FALSE.	.TRUE.
LENGL	Initiali- zation	Controls size (number of computer words within user's job field length) assigned to global dynamic stores (GDS) for this ANOPP run	Integer	≥3000	12000
NAETD	Initiali- zation	Controls number of table-directory entries initially allocated for this ANOPP run	Integer	≥1	10
NAEUD	Initiali- zation	Controls number of data unit directory entries initially allocated for this ANOPP run	Integer	≥5	25
NLPPM	Initiali- zation	Controls number of lines per page to be used for ANOPP printed output during this run	Integer	>15	48
NOGO	Initiali- zation	Determine if ANOPP run is to be limited to primary-edit-phase execution only  NOGO=.TRUE. - primary-edit-phase-only run NOGO=.FALSE. - normal ANOPP run	Logical	.TRUE. .FALSE.	.FALSE.

<sup>a</sup>Usage discussions for each of the initialization parameters are presented in Table 3.3-1.

Checkpoint CP1 will be executed. However, the next checkpoint to be executed depends on the result of the IF CS test. Checkpoint CP2 will be executed if logical user param KEYITEM=.TRUE. followed by run termination via the STOP keyword specification on the CKPNT CS. Otherwise, the SETSYS CS preceding checkpoint CP2 will deactivate checkpoint processing and CP2 CKPNT will be ignored. In this case, checkpoint CP3 would subsequently be executed and terminate the run.

Checkpoint run example 2: For the following example, the run would be initialized with checkpoint processing inactive (system param JCKPNT set to default value .FALSE.). Checkpoint CKP1 would not be executed.

```

STARTCS $
:
CKP1  CKPNT $
:
SETSYS JCKPNT=.TRUE. $
CKP2  CKPNT $
:
ENDCS $

```

The SETSYS CS specification of JCKPNT=.TRUE. would initialize checkpoint processing in the run. The checkpoint file used will have the default external name of CPFILE. Checkpoint CKP2 will be executed, followed by continued run processing since the CKPNT CS keyword STOP was not specified.

Restart run example 1: Assume that the following CS is the only card image in the user's input stream, that this is a restart of the run shown in checkpoint run example 1 above, and that the CKPNT CS labeled CP2 was the last checkpoint executed in that run:

```
RSTRT RSTRT=JETFILE CKPNT=NEWCPF $
```

The operating environment preserved by CKPNT CP2 will be reestablished since RSTRT CS keyword RSCYCL was not specified and the last cycle found on the restart file is used by default. Since the RSTRT CS also specifies initialization parameter CKPNT, checkpoint processing will also be initially active during this restart run. Processing will resume with the execution of the control statement in the input stream of the checkpointed run that follows the CKPNT CS labeled CP2. Subsequent execution of CKPNT control statements will result in new checkpoint cycles being generated on checkpoint file NEWCPF.

Restart run example 2: Again, assume that the following is a restart of the run shown in checkpoint run example 1 above. Since the RSTRT CS keyword RSTRT= is not specified, the system assumes that the user's restart file is assigned to this job by the default external file name RSFILE. Keyword RSCYCL specified that cycle 1 on the file, generated by the CKPNT CS labeled CP1 in the checkpointed run, will be used to reestablish the operat-

ing environment. The user has also chosen to reset the checkpointed value of system parameter JECHO to .FALSE..

```
RSTRT RSCYCL=1 JECHO=.FALSE. $
STARTCS $
:
EXECUTE FMJET $
IF (FMRESULT .EQ. .TRUE.) GOTO ENDIT $
SETSYS JRSTRT=.FALSE. $
ENDIT ENDCS $
```

Following initialization, processing will resume with the execution of the inserted primary input stream above. Following the execution of the ENDCS ending the inserted stream, the system will check the current value of system parameter JRSTRT:

If JRSTRT=.FALSE., restart run processing terminates.

If JRSTRT is still set at restart-run initial value .TRUE., processing will continue with the execution of the control statement following the CKPNT CS labeled CP1 in the input stream of the checkpointed run.

### 3.4 INPUT/OUTPUT REQUIREMENT FOR FUNCTIONAL MODULES

This section describes the input/output requirements for each functional module in GAPAS. The required I/O is either in the form of user parameters or data units. Data units required by users are described in detail, including record formats. Data units created internally within functional modules are described briefly. Functional modules are implemented by using the EXECUTE control statement.

ATMOSPHERIC PROPERTIES MODULE (ATM)

PURPOSE: TO CALCULATE THE ATMOSPHERIC PROPERTIES AT A SPECIFIED  
ALTITUDE

INPUT:

USER PARAMETER

ALTUDE Altitude of aircraft in feet.

OUTPUT:

USER PARAMETERS

VISC Viscosity in slug/sec-ft  
PRESS Ambient pressure in lbf/ft\*\*2  
TEMP Ambient temperature in degree F  
RHOA Ambient density in slug/ft\*\*3  
CO Speed of sound in ft/sec

## GEOMETRY GENERATOR MODULE (GEOGEN)

PURPOSE: TO CALCULATE PROPELLER GEOMETRY DATA FOR THE  
AERODYNAMIC STRUCTURAL AND ACOUSTIC MODULES

### INPUT

#### USER PARAMETERS:

IGLOBE Control flag, =1 for input sections given on global nicely-spaced coordinates. =0 for sections given in local coordinates. (default = 0)

ISPACE Control flag, =0 for section coordinate with "nice" spacing; =1 with arbitrary spacing (DEF = 1)

ISTART Control flag, =0 when GAPAS is being executed the first time through (DEF = 0)

IOPT Options for geometry output, -10 output for sections normal to PCA, 1/4-chord line and elastic axis is required. (DEF = 10)

NIN Number of input sections

BETA3Q Delta blade angle at 3/4 radius of propeller in DEG. (DEF = 0.0)

IROT Index indicating rotational direction. =0 for clockwise rotation (i.e., SR5) =1 for counter-clockwise rotation (i.e., H101/16) (DEF = 0)

INPRNT Control flag for debug print: =0 no debug print. (DEF = 0)

MAXIT Maximum number of iterations for shear center calculation. (DEF = 10)

EPSSHR Convergence criterion for shear center iterations (DEF = .01)

EMOD Young's modulus for propeller material. PSI/IN (DEF = 1.5E+7)

POISS Poisson ratio (DEF = 0.3)

DENS Density of propeller material in LBM/(CUB.FT) (DEF = .17)

#### DATA UNITS (MEMBER):

GEOM (RSLICE) - Input member containing locations along PCA where calculations are to be performed and output for aerodynamic, structural and acoustic modules. This member contains 2 records. The first record is the number of slices and the second record is an array of output locations. (in inch)

GEOM (XCOORD) - Input member containing chordwise locations where calculations are to be performed and output. This is a 2-record member; the first is the number of output stations and second record is an array of output stations. (dimensionless)

GEOM (MSLICE) - Input member containing coordinates of spanwise sections. The format of this member is as follows:

RECORD	DESCRIPTION
1	Number of chordwise station for 1st section.
2	(x,y,z) coordinate
.	.
.	.
.	(x,y,z) and ply angle of last chordwise station
NSLICE+2	Leading edge alignment, horizontal ref. length, chord length, blade angle (DEG.) spanwise location (in inch) number of chordwise stations for 2nd section.
.	(x,y,z) coordinate
.	.
.	.
.	.

(The total number of sections depends on user parameter NIN.)

GEOM(ARBLINE) - This is a multi-record member containing the global coordinates of an arbitrary line upon which input sections are based.

RESTART(XYZPLY) - This is a multi-record member created by the BEAM module for the operation of GAPAS multiple-pass mode. This member contains information of a deflected elastic axis. Specifically, it contains the three coordinates and the ply angle of each peripheral point of a section normal to the elastic axis. It also contains the leading edge alignment, horizontal length, chord length and twist angle for each section.

RESTART(DELXYZ) - This is a multi-record member created by the BEAM module for the multiple-pass mode operation. It contains the deflection (in three directions) and the twisting of the elastic axis for each spanwise station.

#### OUTPUT

##### DATA UNITS:

STRUC(BPROP) - This is a multi-record member created for the use of the BEAM module. It contains structural properties for each section normal to elastic axis.

STRUC(SHRCNTR) - This is a multi-record member created by the shear center routines embedded in GEOGEN module. It contains the three coordinates of the calculated shear center for each spanwise station.



GEOM(BEAMXLE) - This is a multi-record member created for the use of PROPCHG module. It is used for the determination of surface loading to be input in the structure BEAM module. It contains the leading edge coordinates of sections normal to the elastic axis.

GEOM(BMATRIX) - This is a multi-record member to be used by the performance module (PROPCHG). It is used for the determination of surface loading to be input in the BEAM module. It contains the coordinate transformation matrix of sections normal to the elastic axis.

GEOM(BLADE) - This is a multi-record member created for the use of acoustic modules. It contains the local coordinates, blade angle and nose radius of each output section normal to the pitch change axis.

GEOM(BLADEG) - This is a transient multi-record member created for the internal use of GEOGEN module. It contains the global coordinates of each peripheral points of sections normal to the pitch change axis.

GEOM(BLADEL) - this is a multi-record member created for the internal use of a pressure loading module (PLD). It contains the local coordinates of each peripheral point of sections normal to the pitch change axis.

GEOM(DERIV) - This is a multi-record member which contains the first derivative of airfoil sections at various peripheral points. This member is created for the use of TRANSEP module.

GEOM(SCRATCH) - This is a multi-record member generated in GEOGEN module. It contains the spline fit coefficients for internal use within the GEOGEN module. This is a transient data member and therefore not used in any other part of GAPAS.

GEOM(X14CHD) - This is a multi-record member created by GEOGEN module and used in PROPCHG module. It contains the coordinates of mid-points between vortices along the 1/4-chord line.

GEOM(X34CHD) - This is a multi-record member for the use of PROPCHG module. It contains the coordinates and the normal vectors of control points along the 3/4-chord line.

GEOM(RNOSE) - This is a multi-record member created for the use of PROPCHG module. It contains the nose radius, thickness/chord ratio, chord length, blade angle, area of airfoil and camber, for sections normal to the pitch change axis.

GEOM(MATRIX) - This is multi-record member created for the use of JUMPER. It contains the coordinate transformation matrix for points in the propeller plane.

GEOM(VCHORD) - This is a multi-record created for the PROPCHG module. It contains the vector components of the chord of each section.

GEOM(AIRFOIL) - This is a multi-record member created for the use of TRANSEP MODULE. It contains the coordinates of each peripheral points for sections normal to the 1/4-chord line.

GEOM(XVORTX) - This is a multi-record member created by the GEOGEN module and used in the PROPCHG module. It contains the coordinates of the 1/4-chord line for each spanwise station.

#### JUMPER PROPELLER FLOWFIELD MODULE (JUMPER)

PURPOSE: TO CALCULATE FLOWFIELD OF AIRCRAFT AND PROPELLER PLANE

#### INPUT

##### USER PARAMETERS:

VCOM	Free stream velocity (IN FT/SEC)
ALPHA	Aircraft angle of attack (IN DEG.)
YAW	Aircraft angle of YAW (IN DEG.)
XREF	Reference length for output purpose (INCH)
R	Propeller blade radius (INCH)
ERR	Exponent for the value of largest relative error between solution iteration (DEF = -4.0)
RPM	Rotational speed (REV/MIN)
NSYMET	Control flag for symmetry. =0 body and wing are symmetric about X-Z plane. =1 not symmetric. (DEF = 0)
NLIST	Output flag. =0, print out body panel. (DEF = 1)
NCALC	Flow solution flag. =0, perform flow solution; =1, no solution (this is for test run) (DEF = 0)
NWING	Control flag. =0, wing is not modeled. =1, wing is modeled. (DEF = 0)
ITMAX	Maximum number of iterations (DEF = 25)
NAZIMU	Number of azimuthal stations. (DEF = 1)
NSLICE	Number of spanwise stations where velocity on propeller plane is to be determined. (DEF = 11)
NINFLO	Number of inlet or outlet panels on aircraft body (DEF = 0)
NPOINT	Control flag. =0, velocity on propeller plane not to be determined. =1, flowfield required. (DEF = 1)

DATA UNITS (MEMBER):

AIRCRAFT (BODY) - INPUT MULTI-RECORD MEMBER CONTAINING AIRCRAFT BODY PANELING DATA.

RECORD	DESCRIPTION
1	Total number of body cross sections (NSECTO)
2	Cross section sequence number, number of periphery points on the 1st section (NIP), control flag indicating the section is to be repeated (=0), or not to be repeated (=1).
3	(x,y,z) coordinates of the 1st periphery point on 1st section.
4	(x,y,z) coordinates of the 2nd periphery point on 1st section.
5	.....
6	.....
.	.....
.	.....
.	.....
NIP+2	(x,y,z) coordinates of the NIP-TH periphery point on 1st section.
.	(x,y,z) coordinates of the 1st periphery point on 2nd section.
.	.....
.	.....
.	.....
(RECORD 2,3,....NIP+ repeat for each section, up to NSECTO)	aircraft (WING)-data member containing information related to the wing. This member is required only if NWING = 1.

RECORD	DESCRIPTION
1	(x,y,z) coordinate of wing root quarter chord location.
2	Wing span, wing root chord length, wing dihedral angle, (DEG.) sweep angle (DEG.), wing lift coefficient.

AIRCRAFT (PROP) - DATA MEMBER CONTAINING INFORMATION RELATED TO PROPELLER PLANE LOCATION AND ORIENTATION. THIS MEMBER IS REQUIRED ONLY IF NPOINT = 1.

RECORD	DESCRIPTION
1	(x,y,z) coordinate of hub of center point of propeller plane in reference aircraft coordinate system
2	Built-in vertical tilt angle of propeller plane (DEG), built-in sidway tilt angle of propeller plane.

## DATA UNITS:

AIRCRFT(INLET) - This is a one or multi-record member containing flow rates for the inlets of an aircraft. This data member is not required for input if user parameter NINFLO = 0.

AIRCRFT(WING) - This is a multi-record member containing the coordinates of the 1/4-chord point of the root section of wing, and the span, chord length, dihedral angle and sweep angle of wing. This data member is not required for input if user parameter NWING=0

GEOM(MATRIX) - This member is created by GEOGEN module. Refer to GEOGEN input/output for description.

GEOM(X14CHD) - This member is created by GEOGEN module. Refer to GEOGEN input/output for description.

## OUTPUT

AIRCRFT(SCRATCH) - This is a transient multi-record member for the internal use of JUMPER module. It contains coordinate transformation matrices for points on the propeller plane.

AIRCRFT(COEFF) - This is a transient multi-record member containing elements of a matrix related to each panel.

AIRCRFT(ATTACK) - This is a multi-record member containing the angle of attack of airfoil at each spanwise station. This is used in PROPCHG module.

AIRCRFT(VHN) - This is a multi-record member containing the three vector components of inflow. It is used in TRANSEP and PROPCHG modules.

AIRCRFT(FLOWQRT) - This is a multi-record member containing the three velocity components of inflow normal to airfoil sections of each spanwise station.

## TRANSEP AIRFOIL LOADING MODULE

### INPUT

#### USER PARAMETER:

XIBDLY - the x-location (x/c) at which transition is assumed to occur on upper surface. The relationship to percent chord is:  $XIBDLY = (\%chord - 50.) / 100.$

XLBDLY - the x-location at which transition is assumed to occur on lower surface.

XSEP - x-location after which the Nash-Macdonald separation parameter can assume its calculated value. Used only in the viscous interaction case.

XLSEP - location at which the trailing edge correction procedure begins. Use only if ITEUPC and/or ITELWC = 1. Default 0.5.

XPC - location after which the lower surface displacement thickness is required to continue decreasing once it has started to decrease. Upstream of XPC, the displacement thickness is to be monotonically increasing. Default 0.1.

ITACT - viscous interaction control parameter. It should be set to 0 for analysis mode without interaction. Default = 0.

ISKP2 - airfoil update control parameter. It is 0 if on grid 2 an update is desired every 10 iterations. Default = 0.

ISKP3 - same as ISKP2 but for grid 3 (medium grid)

ISKP4 - same as ISKP2 but for grid 4 (fine grid)

IREAD - starting solution parameter. If IREAD=0 the initial perturbation solution is assumed to be everywhere zero. If IREAD=1 an initial solution is input. Default = 0.

ITEUPC - upper surface trailing edge correction control parameter. If ITEUPC=0 correction is not desired. Default = 0.

ITELWC - same as ITEUPC but for lower surface. Default = 0.

W - relaxation factor for subsonic points. Default = 1.7.

EPS - subsonic damping factor to match difference equations at sonic line if needed. Default = 0.0.

EPSS - supersonic damping factor for iterative stability. Default = 0.4.

CONV - convergence criteria control value. Default =  
1.E-05

A1 - stretching constant for the Y direction. Default =  
.246

A2 - first stretching constant for the X direction.  
Default = .15

A3 - second stretching constant for X direction.  
Default =3.87

X4 - the positive X location where coordinate stretching  
changes. Default. 0.49

S4 - the positive value in computational plane where  
stretching changes. Default = 2.0

IMAX - number of vertical grid lines in horizontal di-  
rection. Default = 13

JMAX - number of horizontal grid lines in vertical direc-  
tion. Default = 7

MITER - Maximum number of iterations allowed on first  
grid. Default = 800

NHALF - number of grid refinements to be done. Default = 2

LP - relaxation cycle interval at which boundary layer,  
coordinates are printed. Default = 1000 (no printout)

RN - Reynold's number based on chord length

ALP - angle of attack (DEG.)

## DATA UNITS:

AIRCRAFT(VHN) - member containing the inflow velocity components. It is created by the JUMPER module. Refer to JUMPER for detailed description.

PROP(CLZM) - member containing the free stream Mach number for all mid sections between two adjacent airfoil sections normal to the 1/4-chord line.

GEOM(DERIV), GEOM(AIRFOIL), GEOM(RNOSE) - These are member created by the geometry generator, GEOGEN. Refer to GEOGEN for detailed description.

INIT(SOLUTION) - a multi-record member containing an initial guess of the flowfield around an airfoil. This is not a required input if user parameter IREAD=0.

## OUTPUT

COEFF(CDR1), COEFF(CDR2),....members containing lift, drag and moment coefficients for each section. They are created for performance calculations in PROPCHG module. The structure of the member is the same as that described in NACA16 module.

COEFF(CP) - multi-record member containing the pressure coefficient on upper and lower surface of an airfoil. It includes all sections at specified spanwise stations.

AIRFOIL DATABANK (NACA16)

PURPOSE: TO CALCULATE LIFT AND DRAG AT SPECIFIED ANGLE OF  
ATTACK

USER PARAMETERS:

ALP        Angle of attack (DEG.)  
RN         Reynold's number based on chord length  
MACH       Free stream MACH number.    Default = 0.5  
ALPHAO     Angle of zero lift (DEG.)    Default = -3.5  
CLD        Design lift coefficient  
VISC       Viscosity of ambient air.    (FT\*\*2/SEC).  
           Default = 1.59E-04  
VCOM       Free stream velocity (FT/SEC).    Default - 880  
ICL        Lift coefficient compressibility check factor.  
           ICL=0, gives incompressible lift coeff.    ICL=1,  
           gives compressible lift coeff.    Default = 1.  
ICD        Same as ICL, but for DRAG coefficient.    Default = 1  
MP         Moment coefficient parameter.    MP=0, moment coeff  
           not desired.    MP=1, moment coeff.    About the leading  
           edge, MP=2, moment coeff.    about the 1/4-chord.  
           Default = 2  
IFORM      Output control flag.    Default = 1.  
TC         Thickness to chord ratio



GEOM(RNOSE) - multi-record member containing section properties. Refer to GEOGEN for detailed description.  
AIRCRAFT(ATTACK) - multi-record member containing angle of attack for airfoil sections at each spanwise station. Refer to JUMPER module for detailed description.

PROP(CLD) - multi-record member containing the design lift coefficients for airfoil sections at each specified spanwise station.

#### OUTPUT

COEFF(CDR1), COEFF(CDR2),.... - multi-record member containing angle of attack, Reynold's number, free stream Mach number, and lift and drag coefficients, and moment coefficients about 1/4-chordpoint and leading edge. CDR1, CDR2, ... are members for each spanwise station.

## CLARK-Y DATABANK (CLARKY)

PURPOSE: TO CALCULATE LIFT, DRAG AND MOMENT COEFFICIENTS AT SPECIFIED ANGLE OF ATTACK AND SECTION PROPERTIES.

### INPUT

#### USER PARAMETERS:

ALP - Angle of attack in degree.  
RN - Reynold's number.  
MACH - Free stream Mach number.  
TC - Thickness/chord ratio.  
CHORD - Chord length in feet.  
ALTUDE - Altitude of aircraft in feet.

### OUTPUT

#### DATA UNIT:

COEFF(CDR1), COEFF(CDR2),... multi-record member containing the lift, drag and moment coefficients. The structure of this member is same as that of NACA16. Refer to NACA16 for detailed discussion.

## PROPELLER PERFORMANCE MODULE (PROPCHG)

PURPOSE: TO CALCULATE PROPELLER PERFORMANCE AND AERODYNAMIC  
LOADING FOR STRUCTURE AND ACOUSTICS MODULE

### INPUT

#### USER PARAMETERS:

NB	Number of blades
NSLICE	Number of vortex elements used for calculations
MPROP	The index number of vortex element where proplet is located
INDUC	Control flag to output induced velocity at desig- nated locations. INDUC=1, output is desired. Default = 0.
ID2	ID2=0, inflow velocity calculated in Jumper module is used. ID2=1, uniform inflow. Default = 0
ID	ID=0, normal vector of 3/4 chord line calculated internally is used. ID=1, use normal vector calculated in GEOGEN module. Default = 1
LWRITE	If LWRITE=1 DEBUG print is desired. Default = 0
BETAW	Proplet cant angle (DEG.) Default = 0.0
MACH	Free stream MACH number. Default = 0.5
AJ	Advance ratio
R	Propeller radius (INCH)
RHOA	Ambient density (SLUG/FT**3). Default = .002378
VCOM	Free stream velocity (FT/SEC)

#### DATA UNITS:

GEOM(RNOSE), GEOM(BCHORD), GEOM(XVORTX), GEOM(X34CHD),  
GEOM(X14CHD), GEOM(MATRIX), GEOM(BEAMXLE), GEOM(BMATRIX)

- These are data members created in the Geometry Generator module. They contain information related to airfoil sections. Refer to GEOGEN module for detailed description.

COEFF(CDR1), COEFF(CDR2),... data members containing lift, drag and moment coefficients of each airfoil section. Refer to airfoil loading module, for example, NACA 16 for detailed description.

AIRCRAFT(FLOWQRT) - This is a multi-record member containing the three velocity components of inflow.

#### OUTPUT

##### DATA UNIT:

STRUC(SURFLOAD) - This is a multi-record member containing the surface loading on the 1/4-chord line at locations where structural modules require the aerodynamic loading. This is to be used in the structure BEAM module.

STRUC(X14CHD) - This is a multi-record member which contains the global coordinates of points on the 1/4-chord line intercepting the sections normal to elastic axis. It is created for the use of the BEAM module.

## PRESSURE LOADING MODULE (PLD)

PURPOSE: TO INTERPOLATE THE SURFACE PRESSURE FOR ACOUSTIC CALCULATIONS

### INPUT

#### USER PARAMETERS:

B Blade radius in meters  
VF Free stream velocity in m/sec  
MZ Free stream Mach number  
OMEGA Rotational speed in rad/sec  
ATRAN(I) Constant A used to characterize the pressure profile on each airfoil section (Figure 3.4-1). (def. =.02)  
BTRAN(I) Constant B used to characterize the pressure profile on each airfoil section. (def. =.999)

#### DATA UNITS:

COEFF(CP) - multi-record member containing the pressure coefficient for each airfoil section. This member is created by airfoil loading module TRANSEP. Detailed description can be found in the I/O discussion of TRANSEP.

GRID(PST) - multi-record member containing the psi function of each peripheral points on each airfoil section. This member is created by the rotating blade module (RBS). Detailed description can be found in RBS I/O discussion.

GRID(XI2) - multi-record member containing the chordwise stations for a section in the elliptic coordinate system. The values should range from 0.0 to  $2\pi$ . This is a required input specified by users. For detailed description, refer to SPN module I/O discussion.

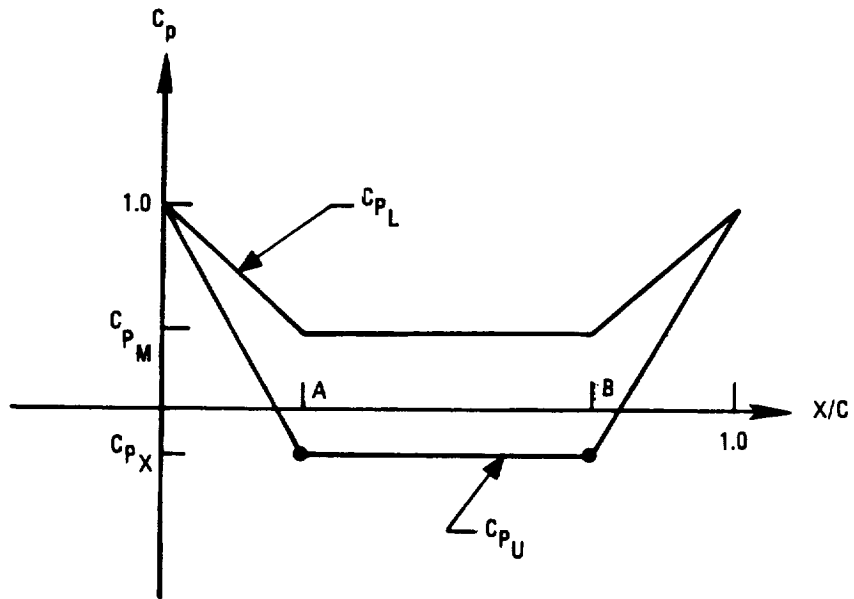
GEOM(RSLICE), GEOM(BLADE), GEOM(BLADEL) - These are data member created by GEOGEN module. Refer to the GEOGEN for detailed description.

PROP(CLD) - multi-record member containing the design lift coefficients for each airfoil section.

PROP(CLZM) - multi-record member containing the lift coefficient and Mach number for each mid-section of airfoil calculated in PROPCHG module.

### OUTPUT

PLD(LOADS) - multi-record table created for the use of acoustic modules. It contains the pressure loading on each section along the propeller span.



A = ATRAN

B = BTRAN

$$C_{PM} = \frac{C_{LD}}{1 + B - A}$$

$$C_{PX} = C_{PM} \cdot \frac{C_{LD}}{1 + B - A}$$

Figure 3.4-1. Definition of ATRAN(I) and BTRAN(I)

#### DATA UNITS:

BEAM(SPINP) - This is a three record member containing the index and the three unit vectors which defines the spin plane.

BEAM(BOUNDRY) - This is a one record member containing the control flags which specify the boundary conditions for each spanwise station.

STRUC(BPROP), STRUC(SHRCNTR) - These are the two members created by the Geometry Generator module (GEOGEN) which contain section properties and locations of shear centers. Refer to GEOGEN for detailed description.

STRUC(X14CHD), STRUC(SURFLOAD) - These are the two members created by the Performance module (PROPCHG). They contain locations of the 1/4-chord line that intercepts the structure sections, and the surface loading. Refer to PROPCHG for detailed description.

#### OUTPUT

#### DATA UNITS:

RESTART(DELXYZ) - This is a multi-record member which contains the deflection and change in twisting angle for each section considered. This member is output for the purpose of finding a converged blade shape in multiple-pass mode operation.

TRANSN2(NBKFL) - This is a transient multi-record member which contains information of the stiffness matrix, force array, etc. This member is only used internally.

TRANSN2(NBM) - This is a transient multi-record member which contains the mass matrix. This member is only used internally.

# TRANSONIC PROPELLER NOISE MODULE (TPN)

PURPOSE - TO CALCULATE THE PERIODIC ACOUSTIC PRESSURE SIGNATURE AND SPECTRUM OF A PROPELLER WITH TRANSONIC TIP SPEED

## INPUT

### USER PARAMETERS:

Z Aircraft altitude (RS), M (FT) (required if IATM=1)  
Default = 0

B Blade length from axis to tip (RS), M (FT)  
Default = 1.

CA Ambient speed of sound (RS), M/S (FT/S)  
(required if IATM=0) Default = 340.294 M/S

RHOA Ambient density (RS), KG/M\*\*3 (SLUGS/FT\*\*3)  
(required if IATM=0) Default = 1.225

ERRTOL Error tolerance for root of retarded time EQ.  
(I) (Default is recommended) Default = 1.E-8

GLIM Tolerance for F(X) = 0. In Newton's method  
(I) (Default is recommended) Default = 1.E-5

FRACOT Fraction of time step to use in observer time  
differentiation (RS) (Default is recommended)  
Default = 1.0

IATM Atmospheric conditions selector (I)  
0 C0 and RHO0 are input parameters  
1 - ATM (TMOD) has ambient conditions  
Default = 0

IPRINT Print option (I)  
0 - no print desired  
1 - input print only  
2 - output print only  
3 - input and output print  
4 - full print + plot tape  
Default = 3

ITMAX Maximum no. of iterations for Newton's method (I)  
Default is recommended) Default = 10

IUNITS System of units indicator (A2 or A7)  
2HSI SI UNITS  
7HENGLISH ENGLISH UNITS  
Default = 2HSI

NBLADE Number of blades on propeller (I). Default = 4

NHARM No. of spectral levels output (I). Default = 16

NTIME No. of points used for time signature.  $NTIME \geq 2 * NHARM \geq 4$   
Default = 100

OMEGA Angular velocity of the blade (RS), Radians/sec.  
Default = 225.

PSIO Initial azimuth angle of blade (RS), radians  
Default = 0.

PLTID 8 letter ID written on 1st record of plot tape  
(A8) Default = 3HTPN



THETAR Root pitch change (RS), radians Default = 0.  
RX Source Radius for output, RE B (RS), (Required if IOU =  
1) Default = 10.  
MZ Aircraft Mach number (RS), Default = .3

## SUBSONIC PROPELLER NOISE MODULE (SPN)

PURPOSE: TO CALCULATE THE PERIODIC ACOUSTIC PRESSURE SIGNATURE  
AND SPECTRUM OF A PROPELLER WITH SUBSONIC TIP SPEED.

### INPUT

#### USER PARAMETERS:

ALT Aircraft altitude M (FT) Default = 0.  
BTIP Blade radius; length from axis to tip M (FT)  
Default = 1.  
C0 Ambient speed of sound M/S (FT/S) Default = 340.294 M/S  
ERRTOL Error criteria; root of retarded time EQ.- Default  
is recommended Default = 1.E-8  
GLIM Tolerance check for Newton's Method - Default  
is recommended Default = 1.E-5  
IATM Atmospheric conditions selector  
0 - C0 and RHO0 are input parameters  
1 - ATM(TMOD) has ambient conditions Default = 0  
IDPDT Blade loading selector  
0 - Blade loading is steady  
1 - Blade loading is time dependent Default = 0  
IPRINT Print option  
0 - No print desired  
1 - input print only  
2 - output print only  
3 - input and output print  
4 - full print + plot tape  
Default = 3  
ITMAX Maximum no. of iterations for Newton's method -  
Default is recommended Default = 10  
IUNITS System of units indicator  
2HSI SI Units  
7ENGLISH English Units  
Default = 2HSI  
NB number of blades on propeller  
Default = 4  
NS No. of spectral levels output  
Default = 16  
NT No. of points used for time signature  
 $4 < 2*NS < NT$   
Default = 100  
METHOD Computational method selector  
1 - full blade formulation  
2 - mean surface approximation  
3 - compact source approximation  
Default = 1  
OMEGA Angular velocity of the blade Radians/Sec  
Default = 225.

PSIO Initial azimuthal angle of blade in  
degrees Default = 0.  
RHO0 Ambient Density  
KG/M\*\*3 (slugs/FT\*\*3)  
Default = 1.225  
RPC Root pitch change  
degrees  
Default = 0.  
RX Source radius for output, re BTIP  
Default = 10.  
VF Aircraft velocity M/S (FT/S)  
Default = 100.

### 3.5 SYSTEM RUNNING PROCEDURE

This section outlines the procedure required to run the GAPAS system. The procedure can be separated into three steps: program compilation, object modules linkage, and execution.

Figure 3.5-1 shows the job control statements required to compile all the functional modules. It should be pointed out that GAPAS is developed to run on the CDC CYBER 175 or 750, NOS operating system. Any attempt to run the system under another operating procedure is likely to result in system errors. Furthermore, it is assumed that a FORTRAN IV compiler is to be used. The six cards from /JOB to DELIVER.B31, are job control cards, and are therefore required only if the NASA Langley CDC computer system is used. The following three GET cards are to retrieve all the functional module source code stored on disk as permanent files. Permanent file XM is the main program driver necessary when the overall system is linked. Next, the FIN and SAVE cards are used to compile each module, and the object module thus obtained is saved as permanent file. On the FTN card, L=0 means that no compiler output listing is desired, and B=ATML means that the object module generated resides on a file named ATML. Finally, /EOF is the end-of-file card.

Figure 3.5-2 shows the input control statement stream required to link the overall GAPAS system. Again, the first six cards are job cards for the NASA Langley computer system. GET,EXECLIB is the control card that retrieves the ANOPP executive system currently residing in the user library. GEOL, JUMPL,.... are object modules for each functional module. GET,SEGDIR retrieves the segment loader directive file for the construction of an overlay executable program (Figure 3.5-3). GAPAS is a large program and requires overlay techniques to meet the main core memory limitation. The current version of GAPAS has employed the segment overlay for reasons of flexibility. SEGDIR contains loader directives which instruct the loader as to how the overall program should be overlaid. The LDSET card sets the load options. LIB=EXECLIB specifies that EXECLIB file is a user library, PRESET=ZERO pre-sets core everywhere to zero; and MAP=SB/MAPLIST puts the load map in a file named MPLIST. The SEGLOAD card invokes the segment loader which takes directives from the file SEGDIR. The

executable, thus constructed, will reside on a file named ANOPTMP. The actual loading is done by invoking the LOAD(XML,ATML,....) command. The NOGO statement allows the loader to complete the loading. Finally, the temporary executable ANOPTMP is saved as a direct file by using the DEFINE and COPY statements.

Figure 3.5-4 shows the job control statements required to execute GAPAS. ATTACH, GAPASX retrieves the GAPAS executable which is a direct file. GET,SR3DAT retrieves the input file for the NASA Lewis SR-3 propeller, which contains a GAPAS control statement stream specifying the type of run to be made. LDSET presets core to zero. GAPASX(SR3DAT) executes the program with input file residing in SR3DAT. Upon completion, the output file which resides on TAPE4 is saved as a permanent file.

```
/JOB
/NOSEQ
GAPAS, 7777.
USER, 977603, UMIRLQ3
CHARGE, 103267, LRC.
DELIVER. B31 FRANKLIN
GET XM.
GET, ATM, GEOGEN, JUMPER, NACA16, TRANSEP.
GET, PROPCHG, BEAM, EIGPAC, PLD, TPN, SPN.
FTN, I=XM, L=0, B=XML.
SAVE, XML.
FTN, I=ATM, L=0, B=ATML
SAVE, ATML.
FTN, I=GEOGEN, L=0, B=GEOL.
SAVE, GEOL.
FTN, I=JUMPER, L=0, B=JUMPL.
SAVE, JUMPL.
FTN, I=NACA16, L=0, B=NACAL.
SAVE, NACAL.
FTN, I=TRANSEP, L=0, B=TRANL.
SAVE, TRANL.
FTN, I=PROPCHG, L=0, B=PROPL.
SAVE, PROPL.
FTN, I=BEAM, L=0, B=BEAML.
SAVE, BEAML.
FTN, I=EIGPAC, L=0, B EIGL.
SAVE, EIGL
FTN, I=PLD, L=0, B=PLDL.
SAVE, PLDL.
FTN, I=TPN, L=0, B=TPNL.
SAVE, TPNL.
FTN, I=SPN, L=0, B=SPNL.
SAVE, SPNL.
/EOF
```

Figure 3.5-1. Compilation of GAPAS Functional Modules

```
/JOB
/NOSEQ
GAPAS,T777.
USER,977603C,UMIRLQ3.
CHARGE,103267,LRC.
DELIVER.B31 FRANKLIN
GET,EXECLIB.
GET,GEOL,JUMPL,NACAL,PROPL,BEAML,EIGL.
REWIND,GEOL,JUMPL,NACAL,PROPL,BEAML,EIGL.
GET,XML,ATML,TRANL,PLDL,SPNL,TPNL.
REWIND,XML,ATML,TRANL,PLDL,SPNL,TPNL.
GET,SEGDIR.
REWIND,SEGDIR.
LDSET(LIB=EXECLIB,PRESET=ZERO,MAP=SB/MAPLST).
SEGLOAD,B=ANOPTMP,LO=DT,I=SEGDIR.
LOAD(XML,EIGL,ATML,GEOL,JUMPL,NACAL,TRANL,PROPL,BEAML,PLDL,SPNL,TPNL
NOGO.
REWIND,ANOPTMP.
REWIND,MAPLST.
DEFINE,GAPASX.
COPY,ANOPTMP,GAPASX.
REPLACE,MAPLST.
```

Figure 3.5-2. Linking GAPAS

```

*****
*
*   THE FOLLOWING INCLUDES AND GLOBALS ARE FOR LEVEL 0 (ROOT)
*
*****
      INCLUDE   DECODE=
      INCLUDE   ENDFILE
      INCLUDE   GETP
      INCLUDE   MNGETU
      INCLUDE   MNOPUS
      INCLUDE   MNPWTU
      INCLUDE   MMREU
      INCLUDE   PUTP
      INCLUDE   REWIND
      INCLUDE   SKIP
      INCLUDE   XCA
      INCLUDE   XCATRA
      GLOBAL    CMNC
      GLOBAL    UCON
      GLOBAL    XANOPP
      GLOBAL    XBSC
      GLOBAL    XCAC
      GLOBAL    XCKRS
      GLOBAL    XCRC
      GLOBAL    XCS
      GLOBAL    XCSFM
      GLOBAL    XCSPC
      GLOBAL    XCVT
      GLOBAL    XDBMC
      GLOBAL    XDSMC
      GLOBAL    XDTMC
      GLOBAL    XPOTH
      GLOBAL    XROOT
      GLOBAL    XSPT
MAIN   INCLUDE  XM
XCO    INCLUDE  XIFCRD
XCO    INCLUDE  XIFCRS
XCO    INCLUDE  XPACX
XCO    INCLUDE  XPAINT
XCO    INCLUDE  XPARD
XCO    INCLUDE  XPARS
XCO    INCLUDE  XPR
XUFMG2 INCLUDE  XSEVM2
*****
*
*   THE FOLLOWING INCLUDES AND GLOBALS ARE FOR TREE
*   SEGMENTS IN LEVEL 1.
*
*****

```

Figure 3.5-3. Overlay Structure



FMP1	INCLUDE	FMP2
FMP1	INCLUDE	FMP3
FMP1	INCLUDE	FMP4
FMP1	INCLUDE	FMP5
FMP1LA	GLOBAL	FMP1LA
FMP1LA	EQUAL	FMP2LA
FMP1LA	EQUAL	FMP3LA
FMP1LA	EQUAL	FMP4LA
FMP1LA	EQUAL	FMP5LA
FM1	INCLUDE	FM2
FM1	INCLUDE	FM3
FM1	INCLUDE	FM4
FM1	INCLUDE	FM5
SPLS	INCLUDE	ALEV
SPLS	INCLUDE	VPWL
SPLS	INCLUDE	BBPNL
FM1LA	GLOBAL	FM1LA
FM1LA	EQUAL	FM2LA
FM1LA	EQUAL	FM3LA
FM1LA	EQUAL	FM4LA
FM1LA	EQUAL	FM5LA
HRNLA	GLOBAL	HRNLA
TPNLA	GLOBAL	TPNLA
RBALA	GLOBAL	RBALA
RBSLA	GLOBAL	RBSLA
SPNLA	GLOBAL	SPNLA
PRTL	GLOBAL	PRTL
XBSLA	GLOBAL	XBSLA
XCG	INCLUDE	XML
XCG	INCLUDE	XTB
XCG	INCLUDE	XTL
XCGFUL	INCLUDE	XDUHDR
XCGFUL	INCLUDE	XFMTQ
XCK	INCLUDE	XEN
XCL	INCLUDE	XCABST
XCL	INCLUDE	XCACLO
XCL	INCLUDE	XCAI
XCL	INCLUDE	XCAMXX
XCL	INCLUDE	XCANCS
XCL	INCLUDE	XEVTBD
XCL	INCLUDE	XRTAMU
XCL	INCLUDE	XRTBAD
XCL	INCLUDE	XRTBCS
XCL	INCLUDE	XRTBLR
XCL	INCLUDE	XRTCAL
XCL	INCLUDE	XRTCSS
XCL	INCLUDE	XRTDAT
XCL	INCLUDE	XRTEND
XCL	INCLUDE	XRTLRF
XCL	INCLUDE	XRTLSA

Figure 3.5-3. Overlay Structure (Continued)

XCL	INCLUDE	XRTLSE
XCL	INCLUDE	XRTODB
XCL	INCLUDE	XRTPIN
XCL	INCLUDE	XRTSAT
XCL	INCLUDE	XRTSAR
XCL	INCLUDE	XRTSCA
XCL	INCLUDE	XRTSCG
XCL	INCLUDE	XRTSCK
XCL	INCLUDE	XRTSCO
XCL	INCLUDE	XRTSCR
XCL	INCLUDE	XRTSDA
XCL	INCLUDE	XRTSDR
XCL	INCLUDE	XRTSDT
XCL	INCLUDE	XRTSDU
XCL	INCLUDE	XRTSEN
XCL	INCLUDE	XRTSER
XCL	INCLUDE	XRTSEV
XCL	INCLUDE	XRTSEX
XCL	INCLUDE	XRTSFN
XCL	INCLUDE	XRTSGO
XCL	INCLUDE	XRTSIF
XCL	INCLUDE	XRTSLD
XCL	INCLUDE	XRTSL1
XCL	INCLUDE	XRTSL2
XCL	INCLUDE	XRTSNL
XCL	INCLUDE	XRTSPA
XCL	INCLUDE	XRTSPL
XCL	INCLUDE	XRTSPR
XCL	INCLUDE	XRTSPU
XCL	INCLUDE	XRTSRE
XCL	INCLUDE	XRTSSS
XCL	INCLUDE	XRTSTA
XCL	INCLUDE	XRTSTI
XCL	INCLUDE	XRTSTL
XCL	INCLUDE	XRTSUL
XCL	INCLUDE	XRTSUP
XCL	INCLUDE	XRTSYN
XCL	INCLUDE	XRTTC
XCL	INCLUDE	XRTU
XCL	INCLUDE	XRTVCS
XCL	INCLUDE	XSEVBD
XCL	INCLUDE	XSEVCK
XCL	INCLUDE	XSEVLE
XCL	INCLUDE	XSEVSC
XCL	INCLUDE	XSEVSM
XCL	INCLUDE	XSEVSR
XCSLA	GLOBAL	XCSLA
XPL	INCLUDE	XAR
XPL	INCLUDE	XAT
XPL	INCLUDE	XDA
XPL	INCLUDE	XDT

Figure 3.5-3. Overlay Structure (Continued)

XPL	INCLUDE	XPU
XPL	INCLUDE	XSS
XPL	INCLUDE	XTI
XRT	INCLUDE	XEVTBD
XRT	INCLUDE	XRTAMU
XRT	INCLUDE	XRTBAD
XRT	INCLUDE	XRTBCS
XRT	INCLUDE	XRTBLR
XRT	INCLUDE	XRTCAL
XRT	INCLUDE	XRTCSS
XRT	INCLUDE	XRTDAT
XRT	INCLUDE	XRTEND
XRT	INCLUDE	XRTTC
XRT	INCLUDE	XRTLRF
XRT	INCLUDE	XRTLSA
XRT	INCLUDE	XRTLSE
XRT	INCLUDE	XRTODB
XRT	INCLUDE	XRTPIN
XRT	INCLUDE	XRTSAR
XRT	INCLUDE	XRTSAT
XRT	INCLUDE	XRTSCA
XRT	INCLUDE	XRTSCG
XRT	INCLUDE	XRTSCK
XRT	INCLUDE	XRTSCO
XRT	INCLUDE	XRTSCR
XRT	INCLUDE	XRTSDA
XRT	INCLUDE	XRTSDR
XRT	INCLUDE	XRTSDT
XRT	INCLUDE	XRTSDU
XRT	INCLUDE	XRTSEN
XRT	INCLUDE	XRTSER
XRT	INCLUDE	XRTSEV
XRT	INCLUDE	XRTSEX
XRT	INCLUDE	XRTSFN
XRT	INCLUDE	XRTSGO
XRT	INCLUDE	XRTSIF
XRT	INCLUDE	XRTSLD
XRT	INCLUDE	XRTSL1
XRT	INCLUDE	XRTSL2
XRT	INCLUDE	XRTSML
XRT	INCLUDE	XRTSPA
XRT	INCLUDE	XRTSPL
XRT	INCLUDE	XRTSPR
XRT	INCLUDE	XRTSPU
XRT	INCLUDE	XRTSRE
XRT	INCLUDE	XRTSSS
XRT	INCLUDE	XRTSTA
XRT	INCLUDE	XRTSTI
XRT	INCLUDE	XRTSTL
XRT	INCLUDE	XRTSUL
XRT	INCLUDE	XRTSUP

Figure 3.5-3. Overlay Structure (Continued)

```

XRT      INCLUDE  XRTSYN
XRT      INCLUDE  XRTU
XRT      INCLUDE  XRTVCS
XRT      INCLUDE  XSEVBD
XRT      INCLUDE  XSEVCK
XRT      INCLUDE  XSEVLE
XRT      INCLUDE  XSEVSC
XRT      INCLUDE  XSEVSM
XRT      INCLUDE  XSEVSR
XRTLA    GLOBAL   XRTLA
XSLFBD   GLOBAL   XSLF
XSLFBD   INCLUDE  XDR
XSLFBD   INCLUDE  XLD
XSLFBD   INCLUDE  XUN
XUP      GLOBAL   XUPC
NACA16   INCLUDE  ATNCON
*****
*
*   THE FOLLOWING TREE DEFINES THE ONLY SEGMENT IN LEVEL 0   *
*
*****
      TREE  MAIN-(TMERR2,MMERR2,XXNMG2,XXFMG2,XUFMG2,FMNMG2,DSMER2,
.XCO)
*****
*
*   THE FOLLOWING TREE SEGMENTS ARE FOR LEVEL 1.           *
*   (NOTE, THERE IS ONLY ONE SEGMENT IN LEVEL 0.)         *
*
*****
      LEVEL
FMP11    TREE  FMP1-FMP1LA
FM11     TREE  FM1-FM1LA
HRN1     TREE  HRN-HRNLA
PROP1    TREE  PROP
TPN1     TREE  TPN-TPNLA
RBA1     TREE  RBA-RBALA
RBS1     TREE  RBS-RBSLA
SPN1     TREE  SPN-SPNLA
PRT1     TREE  PRT-PRTLA
PROPP    TREE  SPLS-(FMP11,HRN1,PROP1,TPN1,RBA1,RBS1,SPN1,PRT1)
      TREE  GEODEN
      TREE  JUMPER
      TREE  NEUMANN
      TREE  EPPLER
      TREE  TRANSEP
      TREE  CLARKY
      TREE  NACA16
      TREE  NACA44
      TREE  NACA65
      TREE  PROPCHG
      TREE  PROPALJ

```

Figure 3.5-3. Overlay Structure (Continued)

```

TREE STRUC-(BEAM,EIGPAC)
TREE PLD
TREE ATM
  TREE XBS-XBSLA
  TREE XCGFUL-(XUP-(XUPCHG.XUPXFR.XUPSYN),XCL.XCG.XSLFBD)
  TREE XCK
  TREE XEV
  TREE XMERR
  TREE XPL
  TREE XRT-XRTL
*****
*
* THE FOLLOWING LEVEL IS FOR THOSE LDSA COMMON BLOCKS USED *
* BY FUNCTIONAL MODULES OR EXECUTIVE MODULES THAT RESIDE *
* IN THE ROOT LEVEL 0. *
*
*****
LEVEL
TREE XCSLA
END XM

```

Figure 3.5-3. Overlay Structure (Continued)

```
/JOB
/NOSEQ
GAPAS,T2000.
USER,977603C,UMIRLQ3
CHARGE,103267,LRC.
DELIVER.B31      FRANKLIN
ATTACH,GAPASX.
GET,SR3DAT.
LDSET(PRESET = ZERO)
GAPASX(SR3DAT)
REWIND,TAPE4.
REPLACE,TAPE4
SRUCOMP.
DAYFILE,DFILE.
REPLACE,DFILE.
EXIT.
SRUCOMP.
DAYFILE,DFILE.
REPALCE,DFILE.
REWIND,TAPE4.
REPLACE,TAPE4
/EOF
```

Figure 3.5-4. Execution of GAPAS

### 3.6 INPUT FILE DESCRIPTION

GAPAS input file contains a set of internally supported control statements that are processed sequentially by the executive part of the system. Different statements perform different tasks by activating various system routines. These statements can be placed in any way relative to each other, as long as they represent a meaningful sequence to the users. To prepare a primary input stream, the following statements must be included.

```
ANOPP  $
STARTCS $
...    $
...    $
ENDCS  $
```

Figure 3.6-1 shows the input file for NASA Lewis SR-3 propeller operating at Mach number = 0.8, advance ratio = 3.06, and sea level altitude. The first two cards (ANOPP, STARTCS) and the last card (ENDCS) must be included. These cards are necessary to set up a primary input stream. In ANOPP, the statement CKPNT=CPNT13 option specifies that a subsequent restart run is desired and that the run environment is saved in a file named CPNT13. Just prior to the EXECUTE ATM, PARAM ALTUDE=0.0 \$ is inserted, which specifies the altitude of flight for atmospheric properties calculations. This statement is followed by three PARAM statements that specify various input parameters for the geometry generator. UPDATE is used to create a data unit named GEOM which possesses three data members named RSLICE, XCOORD, and MSLICE. These members specify the output spanwise stations in GEOM(RSLICE), chordwise stations in GEOM(XCOORD), and coordinates of peripheral points of sections normal to the pitch change axis in GEOM(MSLICE). To close out the data unit, an END\* statement is used. These are all the input required for the geometry generator. With these data, the module is executed by issuing EXECUTE GEOGEN \$.

The next module to be executed is the Aircraft Flowfield module, JUMPER. The required input is either satisfied by data units already created by the GEOGEN, or by user parameters and data units to be created by users. Here a mixture of PARAM and EVALUATE are used to establish values for user parameters. PARAM is used to input values to user

```

ANOPP CKPNT=CPNT13 JECHO=.TRUE. $
STARTCS $
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ INPUT FILE FOR NASA SR-3 PROPELLER OPERATED $
$ UNDER FOLLOWING CONDITIONS MACH=0.8 $
$ ADVANCE RATIO=3.06 ALTITUDE=0.0 $
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ EXECUTE ATMOSPHERIC MODULE $
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
PARAM ALTUDE=0.0 $
EXECUTE ATM $
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ EXECUTE GEOMETRY GENERATOR $
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
PARAM IGLOBE=0, ISPACE=1, NIN=15, IOPT=10, MAXIT=7, ISTART=0 $
PARAM BETA3Q=-.61 EPSSHR=.0005 EMOD=1.59E+7 POISS=.334 DENS=0.16 $
PARAM R=12.25 $
UPDATE NEWU=GEOM, SOURCE=* $
-ADDR OLDM=*, NEWM=RSLICE, FORMAT=0 $
13 $
3.00 3.85 4.65 5.45 6.20 7.00 7.75
8.50 9.20 9.85 10.50 11.20 11.90 $
-ADDR OLDM=*, NEWM=XCOORD, FORMAT=0 $
25 $
1.0 0.95 0.9 0.85 0.8 0.7 0.6 0.5
0.4 0.3 0.25 0.2 0.15 0.1 0.075 0.05 0.03
0.025 0.02 0.015 0.01 0.006 0.003 0.001 0.0 $
-ADDR OLDM=*, NEWM=MSLICE, FORMAT=0 $
53 $
1.000000 0.000000 0.000000 0.000000 $
.958110 .014973 0.000000 0.000000 $
.917160 .028795 0.000000 0.000000 $
.876240 .041627 0.000000 0.000000 $
.835230 .054010 0.000000 0.000000 $
.
.
.
.

```

Figure 3.6-1. Input File for NASA Lewis SR-3 Propeller





```

-ADDR OLDM=*, NEWM=CLD, FORMAT=5H1RS $ $
-0.20 $
-0.01 $
 0.09 $
 0.14 $
 0.17 $
 0.19 $
 0.21 $
 0.23 $
 0.24 $
 0.24 $
 0.23 $
 0.22 $
  END* $
  PARAM MACH=0.8, IAIR23=1 $
  PARAM NOUT = 1 $
L2 CONTINUE $
  PARAM CLL=0.2 $
  PARAM NALPHA=4 $
L1 CONTINUE $
  EXECUTE NACA16 $
  PARAM CLL = CLL + 0.11 $
  PARAM NALPHA = NALPHA - 1 $
  IF(NALPHA .NE. 0) GOTO L1 $
  PARAM NOUT = NOUT + 1 $
  IF(NOUT .LT. NSLICE) GOTO L2 $
#####
$
$ EXECUTE CHANG-SULLIVAN MODULE $
$
#####
PARAM NB=8, NUMEL=12, MPROP=13, INDUC=1 $
PARAM ID2=0, LWRITE=0, ID=0 $
PARAM AJ=3.06, R=12.25 $
EXECUTE PROPCHG $
#####
$
$ EXECUTE STRUCTURE BEAM MODULE $
$
#####

```

Figure 3.6-1. Input File for NASA Lewis SR-3 Propeller (Continued)





parameters directly. EVALUATE is used to establish the values by using existing user parameter values. Following is a data unit named AIRCRFT to be created by using an UPDATE statement. Only one member is required to be input. This member contains the coordinates describing the geometry of the nacelle housing the propeller. At the end, END\* signals the termination of the input data. This is followed by an execution command, namely, EXECUTE JUMPER \$

After GEOGEN and JUMPER are executed, the drag polar is constructed by executing NACA16 data bank repeatedly at various angles of attack for each airfoil section. This requires conditional/branching control statements, IF(.....) GOTO LABEL. In normal cases when a subsonic part of NACA16 is used, angle of attack is varied to obtain the drag polar. For the SR-3 case, however, since the free-stream condition calls for transonic lift/drag data, AIR23 within NACA16 is employed. When AIR23 is used, the lift coefficient is varied to obtain the drag polar. The following is an excerpt from the input file.

```

..... $
..... $
L2 CONTINUE $
PARAM CLL=0.2 $
PARAM NALPHA=4 $
L1 CONTINUE $
EXECUTE NACA16 $
PARAM CLL=CLL + 0.11 $
PARAM NALPHA=NALPHA - 1 $
IF(NALPHA .NE. 0) GOTO L1 $
PARAM NOUT = NOUT + 1 $
IF(NOUT .LT. NSLICE) GOTO L2 $
..... $
..... $

```

There are two conditional loops specified by L2 and L1 CONTINUE \$. The first loop, starting from "L2 CONTINUE" to "IF (.....) GOTO L2," generates a drag polar for each airfoil section. The second loop, starting from "L1 CONTINUE" to "IF (.....) GOTO L1," generates a drag polar for an airfoil at four different lift coefficients.

After the drag polars are constructed, performance calculations are performed by EXECUTE PROPCHG. Finally, deflection due to aerodynamic loading and centrifugal force is determined by EXECUTE STRUC. Following

the structural calculation, CP1 CKPNT STOP \$ is issued. It signals the creation of a checkpoint which effectively saves the run environment in the checkpoint file named CPNT13. The STOP option simply instructs the executive system to stop processing after the checkpoint is created.

Up to this point, a single pass of GAPAS is completed without acoustic calculations. The following block of control statements simply rerun all the modules already discussed except NACA16. These statements constitute the multiple pass of GAPAS processing in which the deflected propeller is iterated until a converged solution is obtained, or until a specified number of iterations is expired.

After either the single-pass or multiple-pass mode of operation is completed, acoustics calculations are ready to be performed. This requires the execution of two other functional modules: the rotating blade shape (RBS) and pressure loading (PLD) modules, which generate data to be input to the Acoustics modules. RBS recasts the airfoil profiles from a Cartesian coordinate system into an Elliptic coordinate system. PLD determines pressure loading on airfoil sections normal to the pitch change axis. Upon completion of RBS and PLD processing, the transonic propeller noise module is executed (EXECUTE TPN \$).

The Hartzell 101/16 propeller which is designed to operate in subsonic speed has also been analyzed for the verification of GAPAS. Figure 3.6-2 shows the input file for a single-pass mode operation. The input control statement stream is similar to the SR-3 case, and is not repeated here.

```

ANOPP CKPNT=CHK317 JECHO=.TRUE. $
STARTCS $
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ INPUT FILE FOR HARTZELL-101/16 PROPELLER $
$ OPERATED UNDER FOLLOWING CONDITIONS $
$ MACH=0.317 ADV RATIO=1.52 ALTUDE=10000.0 $
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ EXECUTE ATMOSPHERIC MODULE $
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
PARAM ALTUDE=3048.00 $
EXECUTE ATM $
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ EXECUTE GEOGNETRY GENERATOR MODULE $
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
PARAM IGLOBE=0, ISPACE=1, MIN=7, IOPT=10, MAXIT=6, ISTART=0 $
PARAM BETA3Q=20.0, EPSSHR=.0005 EMOD=1.03E+7 POISS=0.33 DENS=0.0966 $
PARAM R=53.03 IROT=0 $
UPDATE NEWM=GEOM, SOURCE=* $
-ADDR OLDM=*, NEWM=RSLICE, FORMAT=0 $
13 $
12.19177 15.17773 18.16369 21.14965 24.13561 27.12157,
30.10753 33.0935 36.07946 39.06542 42.05138 45.03734,
48.02330 $
-ADDR OLDM=*, NEWM=XCOORD, FORMAT=0 $
25 $
1.0 0.95 0.9 0.85 0.8 0.7 0.6 0.5
0.4 0.3 0.25 0.2 0.15 0.1 0.075 0.05
0.03 0.025 0.02 0.015 0.01 0.006 0.003 0.001
0.0 $
-ADDR OLDM=*, NEWM=MSLICE, FORMAT=0 $
53 $
1.000000 0.000000 0.000000 60.000000 $
0.995000 0.020000 0.000000 60.000000 $

0.990000 0.000800 0.000000 30.000000 $
0.995000 0.000000 0.000000 30.000000 $
1.000000 0.002500 0.000000 30.000000 $
2.559900 0.848000 7.165600 21.300000 0.000000 0.000000 48.023300 $
END* $
EXECUTE GEOGEN $

```

Figure 3.6-2. Input File for A Single-Pass Mode Operation

```

#####
$
$ EXECUTE JUMPER MODULE $
$
#####
PARAM MACH=.317 AJ=1.52 PI=3.14159 $
EVALUATE VCOM=MACH*CO $
EVALUATE RPM=VCOM*60.*12./(AJ*2.*R) $
PARAM NSYMET=0, NHING=0, NINFLO=0, NAZINU=1, NSLICE=13 $
PARAM NPOINT=1, NCALC=0, NLIST=1, ITMAX=20 $
PARAM ALPHA=0.0, YAU=0.0 $
PARAM XREF=1.0, ERR=-3.0 $
UPDATE NEU=AIRCRAFT, SOURCE=* $
-ADDR OLDN=*, NEUM=BODY, FORMAT=0 $
30 $

1 9 0 $
0.0000 0.0000 0.0000 $
0.0000 0.0000 0.0000 $
0.0000 0.0000 0.0000 $
0.0000 0.0000 0.0000 $
0.0000 0.0000 0.0000 $
0.0000 0.0000 0.0000 $
0.0000 0.0000 0.0000 $
0.0000 0.0000 0.0000 $
0.0000 0.0000 0.0000 $
2 9 0 $
-0.0389 0.0000 -0.4207 $
-0.0389 -0.1610 -0.3887 $
....
....
....
-69.2943 -8.2309 -3.4093 $
-69.2943 -8.9090 0.0000 $
-69.2943 -8.2309 3.4093 $
-69.2943 -6.2996 6.2996 $
-69.2943 -3.4093 8.2309 $
-69.2943 0.0000 8.9090 $
-ADDR OLDN=*, NEUM=PROP, FORMAT=0 $
-14.6398 0.0 0.0 $
0.0 0.0 $
END* $
EXECUTE JUMPER $
#####
$
$ EXECUTE TRANSEP $
$
#####
PARAM NOUT=1 NHALF=1 INOISE=0 ITACT=1 IBERIV=0 ILAM=1 $
L2 CONTINUE $
PARAM ALP=2.0 $
PARAM NALPHA=3 $
L1 CONTINUE_.$
EXECUTE TRANSEP $
PARAM ALP = ALP + 1.20 $
PARAM NALPHA = NALPHA - 1 $
IF(NALPHA .NE. 0) GOTO L1 $
PARAM NOUT = NOUT + 1 $
IF(NOUT .LT. NSLICE) GOTO L2 $

```

Figure 3.6-2. Input File for A Single-Pass Mode Operation (Continued)



```

.....
$
$ EXECUTE CHANG-SULLIVAN MODULE
$
$
$
PARAM NB=3, NUHEL=12, NPROP=13, INDUC=1 $
PARAM ID2=0, LWRITE=0, ID=0 IAIR23=0 $
EXECUTE PROPCBG $
$
$ EXECUTE STRUCTURE BEAM MODULE
$
$
$
PARAM IPROB=1 PL=0.0 PU=10000000.0 NR=6 $
PARAM EPS=.0001 EPS1=.001 B=0.0 EIGEN=.TRUE. INDEX=1 $
PARAM ISTDYN=1, IOPLIN=0, ITERMX=20, ZDELTA=.0001 $
UPDATE NEUU=BEAM, SOURCE=* $
-ADDR OLDN=*, NEUM=SPINP, FORMAT=6HI,3RS $ $
9000 0.0 0.0 0.0 $
9001 0.0 1.0 0.0 $
9002 1.0 0.0 0.0 $
-ADDR OLDN=*, NEUM=BOUNDRY, _FORMAT=4H6I-$...$
1 1 1 1 1 1 $
END* $
EXECUTE STRUC $
PURGE TRANSNT,TRANSN2 $
CPI CKPNT STOP $
PURGE BEAM,RESTART,STRUC $
PARAM B=1.3462, IUNITS=2HSI, IPRINT=3, ZSLOPE=.FALSE., UHEIGHT=.FALSE. $
UPDATE NEUU=GRID SOURCE=* $
-ADDR NEUM=XI2 OLDN=* FORMAT=4H*RS$ MNR=1 $
0.0 .2618 .5236 .7854 1.0472 1.309 1.5708
1.8326 2.0944 2.35619 2.61799 2.87797 3.1415926
3.40339 3.66519 3.92699 4.18879 4.45059 4.71239
4.97419 5.23599 5.49779 5.75959 6.02139 6.28319 $
-ADDR NEUM=PSI OLDN=* FORMAT=4H*RS$ MNR=1 $
0. $
END* $
EXECUTE RBS $
PARAM IDBANK=0 $
PARAM NBLADE=3, NZ=.317, Z=3048.0 $
EVALUATE RH00=RHOA*515.240 $
EVALUATE CO=CO*.3048 $
EVALUATE VF=VCOM*.3048 $
EVALUATE OMEGA=RPH*PI/30.0 $
PARAM IPRINT=4, INFO=1 $
PARAM BTIP=B, NS=30, IATH=0, NB=NBLADE, RPC=0.0 $
PARAM ALT*Z CA=CO NHARM=NS RHOA=RH00 $
EVALUATE POUREF= RH00 * B**5 * OMEGA**3$
EVALUATE POWER = 4. *CP *POUREF /(PI**3) $
EVALUATE PKU=POWER*0.001/NBLADE $
EXECUTE PLD $
UPDATE NEUU=OBSERV SOURCE=* $
-ADDR OLDN=*, NEUM=COORD FORMAT=4H3RS$ $
0.0 -2.024 -0.279 $
0.0 -2.024 -0.406 $
0.0 -2.024 -0.559 $
END* $
EXECUTE SPN $
ENDCS $

```

Figure 3.6-2. Input File for A Single-Pass Mode Operation (Continued)

### 3.7 COMPUTER RESOURCES REQUIREMENTS

System verifications have focused on the NASA Langley SR-3 propeller operating at  $M_\infty = 0.8$  and the Hartzell 101/16 propeller which operates at subsonic speeds. Computer resource requirements, namely CPU time and main storage, were determined based on these studies. Table 3.7-1 shows the system requirements. As noted previously, GAPAS is developed to operate on the CDC computer under the NOS operating system. The total memory requirement is 340K octal words which includes the executive system routines, functional module, system routines, intrinsic functions, etc. Processing the SR-3 and H101/16 test cases under similar conditions (i.e., single pass, 13 spanwise stations and same number of observers for acoustics calculations) requires approximately 400 and 800 seconds of execution, respectively. The basic difference in CPU time is due to the fact that the airfoil data bank is used to build the drag polar for the SR-3 performance calculations while for the H101/16, TRANSEP is utilized to construct the drag polar. It should be pointed out that the estimate for SR-3 is based on the assumption that TPN runs without retarded time equation problems.

Table 3.7-1. GAPAS Limitations and Requirements

COMPUTER SYSTEM:

- CDC - NOS OPERATING SYSTEM
- MEMORY ~340,000 OCTAL WORDS
- CPU TIME (SEC) ~400 (SR3, SINGLE PASS)  
13 STATIONS
- ~800 (H101/16, SINGLE PASS)  
13 STATIONS

FUNCTIONAL MODULE REQUIREMENT:

<u>MODULE</u>	<u>CPU(SEC)</u>	<u>MEMORY (OCTAL WORDS)</u>
ATM	~1.0	~1,000
GEOGEN	113.0	101,000
JUMPER	70.0	50,000
NACA16	~2.0	41,000
TRANSEP	11.0	73,000
PROPCHG	30.0	36,000
STRUC (EIGEN)	48.0	67,000 (40,000)
PLD	3.0	40,000
RBS	4.0	40,000
SPN	40.0	47,000
TPN	20 - 120	43,000



APPENDIX A  
FINITE ELEMENT MATRICES ASSOCIATED WITH THE BEAM  
MODEL DEVELOPMENT

All of the finite elements that are required for doing a static and/or structural dynamic analysis of a rotating blade using a beam model are presented. These element matrices were obtained by using a cubic Hermite interpolation polynomial,  $\{\phi_c\}$ , for the planar deflections (v,w) and a quadratic Hermite polynomial,  $\{\phi_q\}$ , for the axial deflection (u) and the torsional rotation ( $\phi$ ). Each of the element matrices and arrays can be written in the partitioned form as follows:

$$[A] = \left[ \begin{array}{c|c|c|c} \left[ \begin{array}{c} A_{11} \\ A_{21} \\ A_{31} \\ A_{41} \end{array} \right] & \left[ \begin{array}{c} A_{12} \\ A_{22} \\ A_{32} \\ A_{42} \end{array} \right] & \left[ \begin{array}{c} A_{13} \\ A_{23} \\ A_{33} \\ A_{43} \end{array} \right] & \left[ \begin{array}{c} A_{14} \\ A_{24} \\ A_{34} \\ A_{44} \end{array} \right] \\ \hline \left[ \begin{array}{c} A_{21} \\ A_{31} \\ A_{41} \end{array} \right] & \left[ \begin{array}{c} A_{22} \\ A_{32} \\ A_{42} \end{array} \right] & \left[ \begin{array}{c} A_{23} \\ A_{33} \\ A_{43} \end{array} \right] & \left[ \begin{array}{c} A_{24} \\ A_{34} \\ A_{44} \end{array} \right] \\ \hline \left[ \begin{array}{c} A_{31} \\ A_{41} \end{array} \right] & \left[ \begin{array}{c} A_{32} \\ A_{42} \end{array} \right] & \left[ \begin{array}{c} A_{33} \\ A_{43} \end{array} \right] & \left[ \begin{array}{c} A_{34} \\ A_{44} \end{array} \right] \\ \hline \left[ \begin{array}{c} A_{41} \end{array} \right] & \left[ \begin{array}{c} A_{42} \end{array} \right] & \left[ \begin{array}{c} A_{43} \end{array} \right] & \left[ \begin{array}{c} A_{44} \end{array} \right] \end{array} \right] \quad (A.1)$$

$$\{C\} = \left\{ \begin{array}{l} \left\{ \begin{array}{c} C_1 \\ C_2 \\ C_3 \\ C_4 \end{array} \right\} \\ \left\{ \begin{array}{c} C_2 \\ C_3 \\ C_4 \end{array} \right\} \\ \left\{ \begin{array}{c} C_3 \\ C_4 \end{array} \right\} \\ \left\{ \begin{array}{c} C_4 \end{array} \right\} \end{array} \right\} \quad (A.2)$$

where the indices 1, 2, 3, and 4 correspond to the planar deflections, v and w, the torsional rotation,  $\phi$ , and the axial deflection u, respectively. Since the finite element matrix and array expressions are complicated, they will be expressed in terms of their submatrices and subarrays.

### A.1. FINITE ELEMENT DEFINITION OF THE STRAIN ENERGY VARIATION

The finite element matrices and arrays that are associated with the strain energy variation are the linear stiffness matrix, the nonlinear stiffness matrix, and the nonlinear Jacobian matrix.

To express the foregoing finite elements in terms of their submatrix definition, the following constants are defined:

$$\overline{EA\eta_A} = EA\eta_A \cos\beta - EA\zeta_A \sin\beta$$

$$\overline{EA\zeta_A} = EA\zeta_A \cos\beta + EA\eta_A \sin\beta$$

$$\overline{EAC_1} = EAC_1 \cos\beta - EAC_2 \sin\beta$$

$$\overline{EAC_2} = EAC_2 \cos\beta + EAC_1 \sin\beta$$

$$\overline{EAD_1} = EAD_1 \cos\beta - EAD_2 \sin\beta$$

$$\overline{EAD_2} = EAD_2 \cos\beta + EAD_1 \sin\beta$$

$$\overline{EAD'_1} = EAD'_1 \cos\beta - EAD'_2 \sin\beta$$

$$\overline{EAD'_2} = EAD'_2 \cos\beta + EAD'_1 \sin\beta$$

$$\overline{EI_{\eta\eta}} = (EI_{\eta\eta} \cos^2\beta + EI_{\zeta\zeta} \sin^2\beta) + 2 EI_{\eta\zeta} \cos\beta \sin\beta$$

$$\overline{EI_{\zeta\zeta}} = (EI_{\zeta\zeta} \cos^2\beta + EI_{\eta\eta} \sin^2\beta) - 2 EI_{\eta\zeta} \cos\beta \sin\beta$$

$$\overline{EI_{\eta\zeta}} = ((EI_{\zeta\zeta} - EI_{\eta\eta}) \cos\beta \sin\beta + EI_{\eta\zeta} (\cos^2\beta - \sin^2\beta))$$

The linear stiffness matrix is defined with the following submatrices

$$\left[ K_{11}^L \right] = \int_0^L \overline{EI_{\zeta\zeta}} \{\phi_c''\}^T \{\phi_c''\} dx$$

$$\left[ K_{12}^L \right] = \int_0^L \overline{EI_{\eta\zeta}} \{\phi_c''\}^T \{\phi_c''\} dx$$

$$[k_{13}^L] = \int_0^L -\overline{EAD}_1 \{\phi_c''\}^T \{\phi_q''\} - \overline{EAD}_1 \{\phi_c''\}^T \{\phi_q'\} dx$$

$$[k_{14}^L] = \int_0^L -\overline{EA}\eta_A \{\phi_c''\}^T \{\phi_q'\} dx$$

$$[k_{21}^L] = [k_{12}^L]^T$$

$$[k_{22}^L] = \int_0^L \overline{EI} \frac{1}{\eta\eta} \{\phi_c''\}^T \{\phi_c''\} dx$$

$$[k_{23}^L] = \int_0^L -\overline{EAD}_2 \{\phi_c''\}^T \{\phi_q''\} - \overline{EAD}_2 \{\phi_c''\}^T \{\phi_q'\} dx$$

$$[k_{24}^L] = \int_0^L -\overline{EAS}_A \{\phi_c''\}^T \{\phi_q'\} dx$$

$$[k_{31}^L] = [k_{13}^L]^T$$

$$[k_{32}^L] = [k_{23}^L]^T$$

$$[k_{33}^L] = \int_0^L (GJ + \overline{EAD}_3) \{\phi_q'\}^T \{\phi_q'\} + \overline{EAD}_3 \{\phi_q''\}^T \{\phi_q''\}$$

$$+ \overline{EAD}_5 \left[ \{\phi_q''\}^T \{\phi_q'\} + \{\phi_q'\}^T \{\phi_q''\} \right] dx$$

$$[K_{34}^L] = \int_0^L EA D_0 \{\phi_q''\} \{\phi_q'\}^T + EA D_0' \{\phi_q'\} \{\phi_q'\}^T dx$$

$$[K_{41}^L] = [K_{14}^L]^T$$

$$[K_{42}^L] = [K_{24}^L]^T$$

$$[K_{43}^L] = [K_{34}^L]^T$$

$$[K_{44}^L] = \int_0^L EA \{\phi_q'\}^T \{\phi_q'\} dx$$

The nonlinear stiffness matrices are defined with the following sub-matrices:

$$\begin{aligned} [K_{11}^{NL}] &= \int_0^L \frac{EA}{2} \{\phi_c'\}^T [2 \{\phi_q'\} \{U\}^T + \{\phi_c'\} \{V\} \{\phi_c'\}^T \{V\}^T + \{\phi_c'\} \{W\} \{\phi_c'\}^T \{W\}^T] \{\phi_c'\} \\ &+ \{\phi_c'\}^T [EA D_0 \{\phi_q''\} \{\phi\}^T - \overline{EA} \eta_A [\{\phi_c''\} \{V\}^T + \{\phi_q\} \{\phi\}^T \{\phi_c''\} \{W\}^T]] \{\phi_c'\} \\ &+ \{\phi_c'\}^T [ - \overline{EA} \zeta_A [\{\phi_c''\} \{W\}^T - \{\phi_q\} \{\phi\}^T \{\phi_c''\} \{V\}^T]] \{\phi_c'\} \\ &- \frac{1}{2} \{\phi_c''\}^T [ \overline{EA} \eta_A \{\phi_c'\} \{V\}^T - \overline{EA} \zeta_A \{\phi_q\} \{\phi\} \{\phi_c'\} \{V\}^T ] \{\phi_c'\} \\ &- 2 \{\phi_c''\}^T [ \overline{EI} \eta_{\zeta} \{\phi_q\} \{\phi\}^T ] \{\phi_c''\} \end{aligned}$$



$$+ \{\phi_{1'}^i\}^T [EAD_0^i \{\phi_{1'}^i\} \{\phi\}^T + \frac{1}{2} \overline{EAC_0} \{\phi_{1'}^i\} \{\phi\}^T \{\phi_q^i\} \{\phi_q^i\}^T] \{\phi_{1'}^i\} \quad dx$$

$$[K_{12}^{NL}] = \int_0^L -\frac{1}{2} \overline{E\eta_A} \{\phi_{1'}^i\} \{\phi_{1'}^i\}^T \{W\} \{\phi_{1'}^i\} + (\overline{EI}_{SS} - \overline{EI}_{\eta\eta}) \{\phi_{1'}^i\} \{\phi_{1'}^i\}^T \{\phi_q^i\} \{\phi_q^i\}^T \{\phi_{1'}^i\}$$

$$[K_{13}^{NL}] = \int_0^L \frac{1}{2} \overline{EAS_A} \{\phi_{1'}^i\} \{\phi_{1'}^i\}^T [\{\phi_{1'}^i\} \{W\} \{\phi_{1'}^i\} \{W\}^T + 2 \{\phi_{1'}^i\} \{U\} \{\phi_{1'}^i\}^T] \{\phi_q^i\} \\ + GJ \cos\beta \{\phi_{1'}^i\} \{\phi_{1'}^i\}^T [\{\phi_{1'}^i\} \{W\} \cos\beta - \{\phi_{1'}^i\} \{V\} \sin\beta] \{\phi_q^i\} \\ - GJ \sin\beta \{\phi_{1'}^i\} \{\phi_{1'}^i\}^T [\{\phi_{1'}^i\} \{V\} \cos\beta + \{\phi_{1'}^i\} \{W\} \sin\beta] \{\phi_q^i\} \\ + \frac{\{\phi_{1'}^i\} \{\phi_{1'}^i\}^T [\overline{EAD}_2^i \{\phi_q^i\} \{\phi\}^T - \frac{1}{2} \overline{EAC}_1 \{\phi_q^i\} \{\phi\}^T] \{\phi_q^i\}}{\overline{EAD}_2} \\ + \frac{\overline{EAD}_2 \{\phi_{1'}^i\} \{\phi_{1'}^i\}^T \{\phi\} \{\phi_q^i\} \{\phi_q^i\}^T}{\overline{EAD}_2} \quad dx$$

$$[K_{14}^{NL}] = 0$$

$$[K_{21}^{NL}] = \int_0^L -\frac{1}{2} \overline{EAS_A} \{\phi_{1'}^i\} \{\phi_{1'}^i\}^T \{V\} \{\phi_{1'}^i\} + (\overline{EI}_{SS} - \overline{EI}_{\eta\eta}) \{\phi_{1'}^i\} \{\phi_{1'}^i\}^T \{\phi_q^i\} \{\phi_q^i\}^T \{\phi_{1'}^i\} \quad dx$$

$$\begin{aligned}
[ K_{22}^{NL} ] &= \int_0^L \frac{EA}{2} \{ \phi_{,1}^T \} [ 2 \{ \phi_{,1}^T \} \{ U \} + \{ \phi_{,1}^T \} \{ V \} \{ \phi_{,1}^T \} \{ V \} + \{ \phi_{,1}^T \} \{ W \} \{ \phi_{,1}^T \} \{ W \} ] \{ \phi_{,1}^T \} \\
&+ \{ \phi_{,1}^T \} [ EAD_0 \{ \phi_{,1}^T \} \{ \phi \} - E\overline{A}\eta_A [ \{ \phi_{,1}^T \} \{ V \} + \{ \phi_{,1}^T \} \{ \phi \} \{ \phi_{,1}^T \} \{ W \} ] ] \{ \phi_{,1}^T \} \\
&+ \{ \phi_{,1}^T \} [ \quad - E\overline{A}\zeta_A [ \{ \phi_{,1}^T \} \{ W \} - \{ \phi_{,1}^T \} \{ \phi \} \{ \phi_{,1}^T \} \{ V \} ] ] \{ \phi_{,1}^T \} \\
&- \frac{1}{2} \{ \phi_{,1}^T \} [ E\overline{A}\zeta_A \{ \phi_{,1}^T \} \{ W \} + E\overline{A}\eta_A \{ \phi_{,1}^T \} \{ \phi \} \{ \phi_{,1}^T \} \{ W \} ] \{ \phi_{,1}^T \} \\
&+ 2 \{ \phi_{,1}^T \} [ E\overline{I}\eta_{\zeta} \{ \phi_{,1}^T \} \{ \phi \} ] \{ \phi_{,1}^T \} \\
&+ \{ \phi_{,1}^T \} [ EAD_0 \{ \phi_{,1}^T \} \{ \phi \} + \frac{1}{2} EAC_0 \{ \phi_{,1}^T \} \{ \phi \} ] \{ \phi_{,1}^T \} \quad dx
\end{aligned}$$

$$\begin{aligned}
[ K_{23}^{NL} ] &= \int_0^L \frac{1}{2} E\overline{A}\eta_A \{ \phi_{,1}^T \} [ \{ \phi_{,1}^T \} \{ V \} \{ \phi_{,1}^T \} \{ V \} + 2 \{ \phi_{,1}^T \} \{ U \} ] \{ \phi_{,1}^T \} \\
&+ GJ \sin \rho \{ \phi_{,1}^T \} [ \{ \phi_{,1}^T \} \{ W \} \cos \rho - \{ \phi_{,1}^T \} \{ V \} \sin \rho ] \{ \phi_{,1}^T \} \\
&+ GJ \cos \rho \{ \phi_{,1}^T \} [ \{ \phi_{,1}^T \} \{ V \} \cos \rho + \{ \phi_{,1}^T \} \{ W \} \sin \rho ] \{ \phi_{,1}^T \} \\
&- \frac{\{ \phi_{,1}^T \} [ E\overline{AD}_1 \{ \phi_{,1}^T \} \{ \phi \} + \frac{1}{2} E\overline{AC}_2 \{ \phi_{,1}^T \} \{ \phi \} ] \{ \phi_{,1}^T \}}{\quad} \\
&- \frac{E\overline{AD}_1 \{ \phi_{,1}^T \} \{ \phi_{,1}^T \} \{ \phi \} \{ \phi_{,1}^T \}}{\quad} \quad dx
\end{aligned}$$

$$[ K_{24}^{NL} ] = 0$$

$$\begin{aligned}
[ K_{31}^{NL} ] = & \int_0^L \frac{1}{2} \overline{EAS}_A \{ \phi_q \}^T [ \{ \phi_c' \} \{ V \}^T \{ \phi_c' \} \{ V \}^T + \{ \phi_c' \} \{ W \}^T \{ \phi_c' \} \{ W \}^T + 2 \{ \phi_c' \} \{ U \}^T ] \{ \phi_c'' \} \\
& + \{ \phi_q \}^T [ \overline{EI}_{SS} \{ \phi_c'' \} \{ W \}^T - \overline{EI} \eta_S \{ \phi_c'' \} \{ V \}^T ] \{ \phi_c'' \} \\
& - GJ \sin \beta \{ \phi_q \}^T [ \{ \phi_c'' \} \{ V \}^T \cos \beta + \{ \phi_c'' \} \{ W \}^T \sin \beta ] \{ \phi_c' \} \\
& + \frac{1}{2} EAD_0 \{ \phi_q'' \}^T \{ \phi_c' \} \{ V \}^T \{ \phi_c' \} + \frac{1}{2} EAD_0 \{ \phi_q' \}^T \{ \phi_c' \} \{ V \}^T \{ \phi_c' \} \quad dx
\end{aligned}$$

$$\begin{aligned}
[ K_{32}^{NL} ] = & \int_0^L \frac{1}{2} \overline{EA} \eta_A \{ \phi_q \}^T [ \{ \phi_c' \} \{ V \}^T \{ \phi_c' \} \{ V \}^T + \{ \phi_c' \} \{ W \}^T \{ \phi_c' \} \{ W \}^T + 2 \{ \phi_c' \} \{ U \}^T ] \{ \phi_c'' \} \\
& - \{ \phi_q \}^T [ \overline{EI} \eta \eta \{ \phi_c'' \} \{ V \}^T - \overline{EI} \eta_S \{ \phi_c'' \} \{ W \}^T ] \{ \phi_c'' \} \\
& + GJ \cos \beta \{ \phi_q \}^T [ \{ \phi_c'' \} \{ V \}^T \cos \beta + \{ \phi_c'' \} \{ W \}^T \sin \beta ] \{ \phi_c' \} \\
& + \frac{1}{2} EAD_0 \{ \phi_q'' \}^T \{ \phi_c' \} \{ W \}^T \{ \phi_c' \} + \frac{1}{2} EAD_0 \{ \phi_q' \}^T \{ \phi_c' \} \{ W \}^T \{ \phi_c' \} \quad dx
\end{aligned}$$

$$\begin{aligned}
[ K_{33}^{NL} ] = & \int_0^L \{ \phi_q \}^T [ \overline{EAD}_2 \{ \phi_c'' \} \{ V \}^T - \overline{EAD}_1 \{ \phi_c'' \} \{ W \}^T ] \{ \phi_q \} \\
& + \{ \phi_q \}^T [ \overline{EAD}_2 \{ \phi_c'' \} \{ V \}^T - \overline{EAD}_1 \{ \phi_c'' \} \{ W \}^T ] \{ \phi_q \} \\
& - \{ \phi_q \}^T [ \overline{EAC}_1 \{ \phi_c'' \} \{ V \}^T + \overline{EAC}_2 \{ \phi_c'' \} \{ W \}^T - \overline{EAD}_4 \{ \phi_c'' \} \{ \phi \}^T ] \{ \phi_q \}
\end{aligned}$$

$$\begin{aligned}
& + \frac{1}{2} EAC_0 \{\phi'_q\}^T \left[ 2\{\phi'_q\}^T \{U\} + \{\phi'_c\} \{V\} \{\phi'_c\}^T \{V\} + \{\phi'_c\} \{W\} \{\phi'_c\}^T \{W\} \right] \{\phi'_q\} \\
& + \frac{1}{2} \{\phi'_q\}^T \{\phi'_q\} \{\phi\}^T \left[ 2EAD'_4 + (EAC_3 + EAD'_4) \{\phi'_q\} \{\phi\} \right] \{\phi'_q\} \\
& + \{\phi''_q\}^T \left[ \overline{EAD}_2 \{\phi''_c\} \{V\} - \overline{EAD}_1 \{\phi''_c\} \{W\} \right] \{\phi''_q\} \\
& + \{\phi''_q\}^T \left[ \overline{EAD}_2 \{\phi''_c\} \{V\} - \overline{EAD}_1 \{\phi''_c\} \{W\} \right] \{\phi''_q\} \\
& + \frac{1}{2} EAD_4 \{\phi''_q\}^T \{\phi'_q\} \{\phi\} \{\phi'_q\}^T \quad dx
\end{aligned}$$

$$[K_{34}^{NL}] = 0$$

$$[K_{41}^{NL}] = \int_0^L \frac{1}{2} EA \{\phi'_q\}^T \{\phi'_c\} \{V\} \{\phi'_c\}^T + \overline{EAS}_A \{\phi'_q\}^T \{\phi_q\} \{\phi\} \{\phi'_c\}^T \quad dx$$

$$[K_{42}^{NL}] = \int_0^L \frac{1}{2} EA \{\phi'_q\}^T \{\phi'_c\} \{W\} \{\phi'_c\}^T - \overline{EA}\eta_A \{\phi'_q\}^T \{\phi_q\} \{\phi\} \{\phi'_c\}^T \quad dx$$

$$[K_{43}^{NL}] = \int_0^L \frac{1}{2} EAC_0 \{\phi'_q\}^T \{\phi'_q\} \{\phi\} \{\phi'_q\}^T \quad dx$$

$$[K_{44}^{NL}] = 0$$

The nonlinear Jacobian matrix is defined with the following sub-matrices:

$$\begin{aligned}
 [J_{11}^{NL}] = & \int_0^L EA \{\phi_c'\}^T [ \{\phi_q'\}^T \{U\} + \frac{3}{2} \{\phi_c'\}^T \{V\} \{\phi_c'\}^T \{V\} + \frac{1}{2} \{\phi_c'\}^T \{W\} \{\phi_c'\}^T \{W\} ] \{\phi_c'\} \\
 & - \{\phi_c''\}^T [ \overline{EA\eta_A} \{\phi_c'\}^T \{V\} - \overline{EAS_A} \{\phi_q'\} \{\phi\}^T \{\phi_c'\}^T \{V\} ] \{\phi_c'\} \\
 & - \{\phi_c'\}^T [ \overline{EA\eta_A} \{\phi_c'\}^T \{V\} - \overline{EAS_A} \{\phi_q'\} \{\phi\}^T \{\phi_c'\}^T \{V\} ] \{\phi_c'\} \\
 & - \{\phi_c'\}^T [ \overline{EA\eta_A} [ \{\phi_c''\}^T \{V\} + \{\phi_q'\} \{\phi\}^T \{\phi_c''\}^T \{W\} ] - EAD_0 \{\phi_c''\}^T \{\phi\} ] \{\phi_c'\} \\
 & - \{\phi_c'\}^T [ \overline{EAS_A} [ \{\phi_c''\}^T \{W\} - \{\phi_q'\} \{\phi\}^T \{\phi_c''\}^T \{V\} ] ] \{\phi_c'\} \\
 & - 2 \{\phi_c''\}^T [ \overline{EI} \eta_S \{\phi_q'\} \{\phi\}^T ] \{\phi_c''\} \\
 & - GJ \cos \beta \sin \beta [ \{\phi_c'\}^T \{\phi_q'\} \{\phi\}^T \{\phi_c''\} + \{\phi_c''\}^T \{\phi_q'\} \{\phi\}^T \{\phi_c'\} ] \\
 & + \{\phi_c'\}^T [ EAD_0 \{\phi_q'\} \{\phi\}^T + \frac{1}{2} \overline{EAC_0} \underbrace{\{\phi_q'\} \{\phi\}^T \{\phi_q'\} \{\phi\}^T}_{\{\phi_c'\}^T} ] \{\phi_c'\} \, dx
 \end{aligned}$$

$$\begin{aligned}
 [J_{12}^{NL}] = & \int_0^L EA \{\phi_c'\}^T \{\phi_c'\}^T \{V\} \{\phi_c'\}^T \{W\} \{\phi_c'\}^T + (\overline{EI}_{SS} - \overline{EI} \eta \eta) \{\phi_c''\}^T \{\phi_q'\} \{\phi\}^T \{\phi_c'\}^T \\
 & - \{\phi_c''\}^T [ \overline{EA\eta_A} \{\phi_c'\}^T \{W\} - \overline{EAS_A} \{\phi_q'\} \{\phi\}^T \{\phi_c'\}^T \{W\} ] \{\phi_c'\}
 \end{aligned}$$

$$\begin{aligned}
& - \{\phi'_1\}^T \left[ \overline{EA}_S \{\phi'_1\} \{V\} + \overline{EA} \eta_A \{\phi_q\} \{\phi\} \{\phi'_1\} \{V\} \right] \{\phi''_2\} \\
& + GJ \left[ \cos^2 \beta \{\phi''_2\} \{\phi'_1\} \{\phi\} \{\phi''_2\} - \sin^2 \beta \{\phi'_1\} \{\phi_q\} \{\phi\} \{\phi''_2\} \right] dx \\
\left[ J_{13}^{NL} \right] = & \int_0^L \overline{EA}_S \{\phi''_2\} \left[ \{\phi'_1\} \{U\} + \frac{1}{2} \{\phi'_1\} \{V\} \{\phi'_1\} \{V\} + \frac{1}{2} \{\phi'_1\} \{W\} \{\phi'_1\} \{W\} \right] \{\phi_q\} \\
& + \{\phi'_1\} \{\phi''_2\} \{V\} \left[ \overline{EA}_S \{\phi''_2\} \{V\} - \overline{EA} \eta_A \{\phi''_2\} \{W\} \right] \{\phi_q\} \\
& - \{\phi''_2\} \left[ 2 \overline{EI} \eta_S \{\phi''_2\} \{V\} + (\overline{EI} \eta \eta - \overline{EI} \eta_S) \{\phi''_2\} \{W\} \right] \{\phi_q\} \\
& + GJ \cos \beta \{\phi''_2\} \left[ \{\phi'_1\} \{W\} \cos \beta - \{\phi'_1\} \{V\} \sin \beta \right] \{\phi_q\} \\
& - GJ \sin \beta \{\phi'_1\} \left[ \{\phi''_2\} \{V\} \cos \beta + \{\phi''_2\} \{W\} \sin \beta \right] \{\phi_q\} \\
& + EAD_0 \{\phi'_1\} \{\phi''_2\} \{V\} \{\phi_q\} + EAD_0 \{\phi'_1\} \{\phi''_2\} \{V\} \{\phi_q\} \\
& + \{\phi''_2\} \left[ \overline{EAD}'_2 \{\phi_q\} \{\phi\} - \overline{EAC}'_1 \{\phi_q\} \{\phi\} \right] \{\phi_q\} \\
& + \{\phi''_2\} \left[ \overline{EAD}'_2 \{\phi''_2\} \{\phi\} + \overline{EAD}'_2 \{\phi'_1\} \{\phi\} \right] \{\phi_q\} \\
& + \overline{EAD}'_2 \{\phi''_2\} \{\phi_q\} \{\phi''_2\} + EAC_0 \{\phi'_1\} \{\phi''_2\} \{V\} \{\phi_q\} \{\phi_q\} \quad dx
\end{aligned}$$

$$\left[ J_{14}^{NL} \right] = \int_0^L EA \{\phi'_1\} \{\phi''_2\} \{V\} \{\phi_q\} + \overline{EA}_S \{\phi''_2\} \{\phi_q\} \{\phi\} \{\phi_q\} \quad dx$$

$$[ J_{21}^{NL} ] = [ J_{12}^{NL} ]^T$$

$$\begin{aligned}
[ J_{22}^{NL} ] = & \int_0^L EA \{ \phi_1' \}^T [ \{ \phi_1' \} \{ U \} + \frac{1}{2} \{ \phi_1' \} \{ V \} \{ \phi_1' \}^T \{ V \} + \frac{3}{2} \{ \phi_1' \} \{ W \} \{ \phi_1' \}^T \{ W \} ] \{ \phi_1' \} \\
& - \{ \phi_2'' \}^T [ \overline{EASA} \{ \phi_1' \} \{ W \} + \overline{EA\eta_A} \{ \phi_1' \} \{ \phi \} \{ \phi_1' \} \{ W \} ] \{ \phi_1' \} \\
& - \{ \phi_1' \}^T [ \overline{EASA} \{ \phi_1' \} \{ W \} + \overline{EA\eta_A} \{ \phi_1' \} \{ \phi \} \{ \phi_1' \} \{ W \} ] \{ \phi_1' \} \\
& - \{ \phi_1' \}^T [ \overline{EA\eta_A} \{ \phi_1' \} \{ V \} + \{ \phi_1' \} \{ \phi \} \{ \phi_1' \} \{ W \} ] - \overline{EAD_0} \{ \phi_1' \} \{ \phi \} \{ \phi_1' \} \\
& - \{ \phi_1' \}^T [ \overline{EASA} \{ \phi_2'' \} \{ W \} - \{ \phi_1' \} \{ \phi \} \{ \phi_1' \} \{ V \} ] \{ \phi_1' \} \\
& + 2 \{ \phi_2'' \}^T [ \overline{EI} \eta_S \{ \phi_1' \} \{ \phi \} ] \{ \phi_1' \} \\
& + GJ \cos \beta \sin \beta [ \{ \phi_1' \} \{ \phi_1' \} \{ \phi \} \{ \phi_2'' \} + \{ \phi_2'' \} \{ \phi_1' \} \{ \phi \} \{ \phi_1' \} ] \\
& + \{ \phi_1' \}^T [ \overline{EAD_0}' \{ \phi_1' \} \{ \phi \} + \frac{1}{2} \overline{EAC_0} \{ \phi_1' \} \{ \phi \} \{ \phi_1' \} ] \{ \phi_1' \} \quad dx
\end{aligned}$$

$$\begin{aligned}
[ J_{23}^{NL} ] = & \int_0^L \overline{EA\eta_A} \{ \phi_2'' \}^T [ \{ \phi_1' \} \{ U \} + \frac{1}{2} \{ \phi_1' \} \{ V \} \{ \phi_1' \}^T \{ V \} + \frac{1}{2} \{ \phi_1' \} \{ W \} \{ \phi_1' \}^T \{ W \} ] \{ \phi_1' \} \\
& + \{ \phi_1' \} \{ \phi_1' \} \{ W \} [ \overline{EASA} \{ \phi_2'' \} \{ V \} - \overline{EA\eta_A} \{ \phi_2'' \} \{ W \} ] \{ \phi_1' \} \\
& + \{ \phi_1' \}^T [ 2 \overline{EI} \eta_S \{ \phi_2'' \} \{ W \} + (\overline{EI}_{SS} - \overline{EI}_{\eta\eta}) \{ \phi_2'' \} \{ V \} ] \{ \phi_1' \}
\end{aligned}$$

$$\begin{aligned}
& - GJ \sin \theta \{\phi''\}^T [ \{\phi'\} \{V\} \sin \theta - \{\phi'\} \{W\} \cos \theta ] \{\phi'_q\}^T \\
& + GJ \cos \theta \{\phi''\}^T [ \{\phi''\} \{W\} \sin \theta + \{\phi''\} \{V\} \cos \theta ] \{\phi'_q\}^T \\
& + EAD_0 \{\phi'\}^T \xi \{W\} \{\phi''\}^T + EAD'_0 \{\phi'\}^T \xi \{\phi'\}^T \xi \{W\} \{\phi'_q\}^T \\
& - \frac{\{\phi''\}^T [ \overline{EAD}_1 \{\phi''\} \{\phi\}^T + \overline{EAC}_2 \{\phi'\} \{\phi\}^T ] \{\phi'_q\}^T}{\{\phi''\}^T [ \overline{EAD}_1 \{\phi''\} \{\phi\}^T + \overline{EAD}'_1 \{\phi'\} \{\phi\}^T ] \{\phi'_q\}^T} \\
& - \frac{\overline{EAD}_1 \{\phi''\}^T \{\phi''\} \{\phi\}^T \{\phi''\}^T + EAC_0 \{\phi'\}^T \{\phi''\} \{W\} \{\phi'_q\}^T \{\phi\} \{\phi'_q\}^T}{\{\phi''\}^T [ \overline{EAD}_1 \{\phi''\} \{\phi\}^T + \overline{EAD}'_1 \{\phi'\} \{\phi\}^T ] \{\phi'_q\}^T} dx
\end{aligned}$$

$$\left[ J_{24}^{NL} \right] = \int_0^L EA \{\phi'\}^T \{\phi''\} \{W\} \{\phi'_q\}^T - \overline{EA} \eta_A \{\phi''\}^T \{\phi''\} \{\phi\} \{\phi'_q\}^T dx$$

$$\left[ J_{31}^{NL} \right] = \left[ J_{13}^{NL} \right]^T$$

$$\left[ J_{32}^{NL} \right] = \left[ J_{23}^{NL} \right]^T$$



$$\begin{aligned}
[ J_{33}^{NL} ] &= \int_0^L \{ \Phi_q \}^T \left[ \underline{\underline{\overline{\overline{EAD}}_2}} \{ \Phi_c'' \} \{ V \}^T - \underline{\underline{\overline{\overline{EAD}}_1}} \{ \Phi_c'' \} \{ W \}^T \right] \{ \Phi_q' \} \\
&+ \{ \Phi_q' \}^T \left[ \underline{\underline{\overline{\overline{EAD}}_2}} \{ \Phi_c'' \} \{ V \}^T - \underline{\underline{\overline{\overline{EAD}}_1}} \{ \Phi_c'' \} \{ W \}^T \right] \{ \Phi_q \} \\
&- \{ \Phi_q' \}^T \left[ \overline{\overline{EAC}}_1 \{ \Phi_c'' \} \{ V \}^T + \overline{\overline{EAC}}_2 \{ \Phi_c'' \} \{ W \}^T - \underline{\underline{\overline{\overline{EAD}}_4}} \{ \Phi_q'' \} \{ \phi \}^T \right] \{ \Phi_q' \} \\
&+ \underline{\underline{\frac{1}{2} \overline{\overline{EAC}}_0 \{ \Phi_q' \}^T \left[ 2 \{ \Phi_q' \} \{ U \}^T + \{ \Phi_c' \} \{ V \} \{ \Phi_c' \} \{ V \}^T + \{ \Phi_c' \} \{ W \} \{ \Phi_c' \} \{ W \}^T \right] \{ \Phi_q' \}}} \\
&+ \underline{\underline{\{ \Phi_q \}^T \left[ \overline{\overline{EAD}}_2 \{ \Phi_c'' \} \{ V \}^T - \overline{\overline{EAD}}_1 \{ \Phi_c'' \} \{ W \}^T \right] \{ \Phi_q'' \}}} \\
&+ \underline{\underline{\{ \Phi_q'' \}^T \left[ \overline{\overline{EAD}}_2 \{ \Phi_c'' \} \{ V \}^T - \overline{\overline{EAD}}_1 \{ \Phi_c'' \} \{ W \}^T \right] \{ \Phi_q \}}} \\
&+ \underline{\underline{\{ \Phi_q' \} \{ \Phi_q' \} \{ \phi \}^T \left[ 2 \overline{\overline{EAD}}_4 + \frac{3}{2} (\overline{\overline{EAC}}_3 + \overline{\overline{EAD}}_4) \{ \Phi_q' \} \{ \phi \}^T \right] \{ \Phi_q' \}}} \\
&+ \underline{\underline{\overline{\overline{EAD}}_4 \left[ \{ \Phi_q'' \} \{ \Phi_q' \} \{ \phi \} \{ \Phi_q' \} + \{ \Phi_q' \} \{ \Phi_q' \} \{ \phi \} \{ \Phi_q'' \} \right]}} \quad dx
\end{aligned}$$

$$\begin{aligned}
[ J_{34}^{NL} ] &= \int_0^L \{ \Phi_q \}^T \left[ \overline{\overline{EAC}}_A \{ \Phi_c'' \} \{ V \}^T - \overline{\overline{EA\eta}}_A \{ \Phi_c'' \} \{ W \}^T \right] \{ \Phi_q' \} \\
&+ \underline{\underline{\overline{\overline{EAC}}_0 \{ \Phi_q' \} \{ \Phi_q' \} \{ \phi \} \{ \Phi_q' \}}} \quad dx
\end{aligned}$$

$$\left[ J_{41}^{NL} \right] = \left[ J_{14}^{NL} \right]^T$$

$$\left[ J_{42}^{NL} \right] = \left[ J_{24}^{NL} \right]^T$$

$$\left[ J_{43}^{NL} \right] = \left[ J_{34}^{NL} \right]^T$$

$$\left[ J_{44}^{NL} \right] = 0$$

#### A.2. FINITE ELEMENT DEFINITION OF THE KINETIC ENERGY VARIATION

The finite element matrices and arrays that are associated with the kinetic energy variation are the mass matrix, the Coriolis damping matrix, the centrifugal force matrix, and the centrifugal force array.

To express the foregoing finite elements in terms of their sub-matrix definition, the following constants are defined:

$$\overline{m}_{\eta m} = m_{\eta m} \cos \beta - m_{\zeta m} \sin \beta$$

$$\overline{m}_{\zeta m} = m_{\zeta m} \cos \beta + m_{\eta m} \sin \beta$$

$$\overline{I}_{\eta \eta} = (I_{\eta \eta} \cos^2 \beta + I_{\zeta \zeta} \sin^2 \beta) + 2 I_{\eta \zeta} \cos \beta \sin \beta$$

$$\overline{I}_{\zeta \zeta} = (I_{\zeta \zeta} \cos^2 \beta + I_{\eta \eta} \sin^2 \beta) - 2 I_{\eta \zeta} \cos \beta \sin \beta$$

$$\overline{I}_{\eta \zeta} = ((I_{\zeta \zeta} - I_{\eta \eta}) \cos \beta \sin \beta + I_{\eta \zeta} (\cos^2 \beta - \sin^2 \beta))$$

$$\overline{h}_{\Omega_x} = (\Omega_y^2 + \Omega_z^2) h_x - \Omega_x \Omega_y h_y - \Omega_x \Omega_z h_z$$

$$\begin{aligned} \overline{h\Omega}_y &= -\Omega_x \Omega_y h_x + (\Omega_x^2 + \Omega_z^2) h_y - \Omega_y \Omega_z h_z \\ \overline{h\Omega}_z &= -\Omega_x \Omega_z h_x - \Omega_y \Omega_z h_y + (\Omega_x^2 + \Omega_y^2) h_z \end{aligned}$$

The linear mass matrix is defined with the following submatrices.

All the submatrices are zero except:

$$[M_{11}] = \int_0^L m \{\Phi_c\} \{\Phi_c\}^T dx$$

$$[M_{13}] = -\int_0^L m \xi_m \{\Phi_c\} \{\Phi_q\}^T dx$$

$$[M_{22}] = \int_0^L m \{\Phi_c\} \{\Phi_c\}^T dx$$

$$[M_{23}] = \int_0^L m \overline{\eta}_m \{\Phi_c\} \{\Phi_q\}^T dx$$

$$[M_{31}] = [M_{13}]^T$$

$$[M_{32}] = [M_{23}]^T$$

$$[M_{33}] = \int_0^L (Im_{\eta\eta} + Im_{SS}) \{\Phi_q\} \{\Phi_q\}^T dx$$

$$[M_{44}] = \int_0^L m \{\Phi_q\} \{\Phi_q\}^T dx$$

The linear Coriolis damping matrix is defined with the following submatrices. All the submatrices are equal to zero except:

$$[ M_{11}^C ] = 2 \int_0^L \Omega_z \overline{m\eta}_m [ \{\phi_c'\}^T \{\phi_c\} - \{\phi_c\}^T \{\phi_c'\} ] dx$$

$$[ M_{12}^C ] = -2 \int_0^L \Omega_x m \{\phi_c\}^T \{\phi_c\} + \Omega_y \overline{m\eta}_m \{\phi_c'\}^T \{\phi_c\} + \Omega_z \overline{m\zeta}_m \{\phi_c\}^T \{\phi_c'\} dx$$

$$[ M_{13}^C ] = \int_0^L -2 \Omega_x \overline{m\eta}_m \{\phi_c\}^T \{\phi_q\} + (\Omega_y h_z - \Omega_z h_y) \overline{m\zeta}_m \{\phi_c'\}^T \{\phi_q\} dx$$

$$[ M_{14}^C ] = 2 \int_0^L \Omega_z m \{\phi_c\}^T \{\phi_q\} dx$$

$$[ M_{21}^C ] = - [ M_{12}^C ]^T$$

$$[ M_{22}^C ] = 2 \int_0^L \Omega_y \overline{m\zeta}_m [ \{\phi_c\}^T \{\phi_c'\} - \{\phi_c'\}^T \{\phi_c\} ] dx$$

$$[ M_{23}^C ] = \int_0^L -2 \Omega_x \overline{m\zeta}_m \{\phi_c\}^T \{\phi_q\} + (\Omega_z h_y - \Omega_y h_z) \overline{m\eta}_m \{\phi_c'\}^T \{\phi_q\} dx$$

$$[ M_{24}^C ] = -2 \int_0^L \Omega_y m \{\phi_c\}^T \{\phi_q\} dx$$

$$[ M_{31}^C ] = - [ M_{13}^C ]^T$$

$$[M_{32}^C] = - [M_{23}^C]^T$$

$$[M_{41}^C] = - [M_{14}^C]^T$$

$$[M_{42}^C] = - [M_{24}^C]^T$$

The centrifugal stiffening matrix is defined with the following sub-matrices:

$$[K_{11}^{CF}] = \int_0^L \bar{m} (\Omega_x^2 + \Omega_z^2) \{\Phi_C\} \{\Phi_C\}^T - \bar{m} \bar{\eta}_m \Omega_x \Omega_y [ \{\Phi_C\} \{\Phi_C\}^T + \{\Phi_C\} \{\Phi_C\}^T ] dx$$

$$[K_{12}^{CF}] = \int_0^L \bar{m} \Omega_y \Omega_z \{\Phi_C\} \{\Phi_C\}^T - \bar{m} \bar{\zeta}_m \Omega_x \Omega_y \{\Phi_C\} \{\Phi_C\}^T - \bar{m} \bar{\eta}_m \Omega_x \Omega_z \{\Phi_C\} \{\Phi_C\}^T dx$$

$$[K_{13}^{CF}] = \int_0^L [ \bar{m} \zeta_m (\Omega_x^2 + \Omega_z^2) + \bar{m} \bar{\eta}_m \Omega_y \Omega_z ] \{\Phi_C\} \{\Phi_C\}^T - \bar{m} \bar{\zeta}_m (\bar{h} \bar{\alpha}_x + x (\Omega_y^2 + \Omega_z^2)) \{\Phi_C\} \{\Phi_C\}^T dx$$

$$[K_{14}^{CF}] = \int_0^L \bar{m} \Omega_x \Omega_y \{\Phi_C\} \{\Phi_C\}^T dx$$

$$[K_{21}^{CF}] = [K_{12}^{CF}]^T$$

$$[K_{22}^{CF}] = \int_0^L -\bar{m} (\Omega_x^2 + \Omega_y^2) \{\Phi_c\}^T \{\Phi_c\} - \bar{m} \bar{S}_m \Omega_x \Omega_z [\{\Phi_c\}^T \{\Phi_c\} + \{\Phi_c\}^T \{\Phi_c\}] dx$$

$$[K_{23}^{CF}] = \int_0^L [ -\bar{m} \bar{\eta}_m (\Omega_x^2 + \Omega_y^2) - \bar{m} \bar{S}_m \Omega_y \Omega_z ] \{\Phi_c\}^T \{\Phi_q\} + \bar{m} \bar{\eta}_m (\bar{h} \Omega_x + x (\Omega_y^2 + \Omega_z^2)) \{\Phi_c\}^T \{\Phi_q\} dx$$

$$[K_{24}^{CF}] = \int_0^L \bar{m} \Omega_x \Omega_z \{\Phi_c\}^T \{\Phi_q\} dx$$

$$[K_{31}^{CF}] = [K_{13}^{CF}]^T$$

$$[K_{32}^{CF}] = [K_{23}^{CF}]^T$$

$$[K_{33}^{CF}] = \int_0^L - [ \bar{m} \bar{\eta}_m \bar{h} \Omega_y + \bar{m} \bar{S}_m \bar{h} \Omega_z - x \Omega_x (\bar{m} \bar{\eta}_m \Omega_y + \bar{m} \bar{S}_m \Omega_z) ] \{\Phi_q\}^T \{\Phi_q\} dx$$

$$[K_{34}^{CF}] = 0$$

$$[K_{41}^{CF}] = [K_{14}^{CF}]^T$$

$$[K_{42}^{CF}] = [K_{24}^{CF}]^T$$

$$[K_{43}^{CF}] = [K_{34}^{CF}]^T$$

$$[K_{44}^{CF}] = \int_0^L -m (\Omega_y^2 + \Omega_z^2) \{\Phi_q^T\} \{\Phi_q\} dx$$

The centrifugal force array is defined with the following subarrays:

$$\begin{aligned} \{F_1^{CF}\} = \int_0^L & - [m (\overline{h\Omega}_y - x \Omega_x \Omega_y) + \overline{m\eta}_m (\Omega_x^2 + \Omega_z^2) - \overline{m\zeta}_m \Omega_y \Omega_z] \{\Phi_c^T\} \\ & + [\overline{m\eta}_m (\overline{h\Omega}_x + x (\Omega_y^2 + \Omega_z^2)) - \overline{I_m}_{SS} \Omega_x \Omega_y - \overline{I_m}_{\eta S} \Omega_x \Omega_z] \{\Phi_c^T\} dx \end{aligned}$$

$$\begin{aligned} \{F_2^{CF}\} = \int_0^L & - [m (\overline{h\Omega}_z - x \Omega_x \Omega_z) + \overline{m\zeta}_m (\Omega_x^2 + \Omega_y^2) - \overline{m\eta}_m \Omega_y \Omega_z] \{\Phi_c^T\} \\ & + [\overline{m\zeta}_m (\overline{h\Omega}_x + x (\Omega_y^2 + \Omega_z^2)) - \overline{I_m}_{\eta\eta} \Omega_x \Omega_z - \overline{I_m}_{\eta S} \Omega_x \Omega_y] \{\Phi_c^T\} dx \end{aligned}$$

$$\begin{aligned} \{F_3^{CF}\} = \int_0^L & [\overline{m\eta}_m (x \Omega_x \Omega_z - \overline{h\Omega}_z) + \overline{m\zeta}_m (\overline{h\Omega}_y - x \Omega_x \Omega_y)] \{\Phi_q^T\} \\ & - [m_y \Omega_z (\overline{I_m}_{\eta\eta} - \overline{I_m}_{SS}) + (\Omega_y^2 - \Omega_z^2) \overline{I_m}_{\eta S}] \{\Phi_q^T\} dx \end{aligned}$$

$$\{F_4^{CF}\} = \int_0^L - [m (\overline{h\Omega}_x + x (\Omega_y^2 + \Omega_z^2)) - \overline{m\eta}_m \Omega_x \Omega_y - \overline{m\zeta}_m \Omega_x \Omega_z] \{\Phi_q^T\} dx$$

### A.3. FINITE ELEMENT DEFINITION OF THE EXTERNAL WORK VARIATION

The finite elements that are associated with the external work variation are the externally applied moment stiffening matrix and the force array.

The submatrices associated with the twisting moments going through a moderate deflection are defined. All the submatrices are equal to zero except:

$$\left[ K_{12}^I \right] = \int_0^L q_x \{\phi_c'\}^T \{\phi_c'\} dx$$

$$\left[ K_{21}^I \right] = \int_0^L -q_x \{\phi_c'\}^T \{\phi_c'\} dx$$

$$\left[ K_{31}^I \right] = \int_0^L -q_y \{\phi_q'\}^T \{\phi_c'\} dx$$

$$\left[ K_{32}^I \right] = \int_0^L -q_z \{\phi_q'\}^T \{\phi_c'\} dx$$

$$\left[ K_{41}^I \right] = \int_0^L q_z \{\phi_q'\}^T \{\phi_c'\} dx$$

$$\left[ K_{42}^I \right] = \int_0^L -q_y \{\phi_q'\}^T \{\phi_c'\} dx$$



The subarrays associated with the distributed load vector are:

$$\{ F_1^I \} = \int_0^L -p_y \{ \phi_c \}^T - q_z \{ \phi'_c \}^T dx$$

$$\{ F_2^I \} = \int_0^L -p_z \{ \phi_c \}^T + q_y \{ \phi'_c \}^T dx$$

$$\{ F_3^I \} = \int_0^L -q_x \{ \phi_q \}^T dx$$

$$\{ F_4^I \} = \int_0^L -p_x \{ \phi_q \}^T dx$$



## REFERENCES

- 2.2-1. Jumper, S. J., "Computer Prediction of Three-Dimensional Potential Flow Fields in Which Aircraft Propellers Operate," Master of Science Thesis, Pennsylvania State University, 1982.
- 2.2-2. Jumper, S. J., "Computer Prediction of Three-Dimensional Potential Flow Fields in Which Aircraft Propellers Operate (Computer Program Description and User's Manual)," Pennsylvania State University, PSU-AERO-R-79/80-25, 1979.
- 2.2-3. Hess, J. L. and Smith, A. M. O., "Calculation of Potential Flow About Arbitrary Bodies," Progress in Aeronautical Sciences Vol. 8, Pergamon Press, Oxford, pp. 1-138, 1966.
- 2.3-1. Chang, L. K., "The Theoretical Performance of High Efficiency Propellers," Ph.D. Thesis, Purdue University, December 1980.
- 2.3-2. Sullivan, J. P., "The Effect of Blade Sweep on Propeller Performance," AIAA Preprint 77-716, June 1977.
- 2.3-3. Bober, L. J. and Chang, L. K., "Factors Influencing the Predicted Performance of Advanced Propeller Designs," AIAA Preprint 81-1564, 1981.
- 2.3-4. Robinson, A. and Laurmann, J. A., Wing Theory, Cambridge University Press, 1965.
- 2.3-5. Lamb, H., "Magnetic Field of a Helix," Proc. Camb. Phil. Soc. 21, p. 477, June 15, 1923.
- 2.3-6. Carnahan, B., Luther, A. A., and Wilkes, J. O, Applied Numerical Methods, Wiley, 1969.
- 2.4-1. Murman, Earl, and Cole, Julian D., "Calculation of Plane Steady Transonic Flows," AIAA J., Vol. 9, No. 1, pp. 114-121, Jan. 1971.
- 2.4-2. Garabedian, P. R., and Korn, D. G., "Analysis of Transonic Airfoils," Comm. on Pure and Applied Math., Vol. 24, pp. 841-851, 1971.
- 2.4-3. Steger, J. L., and Lomax, H., "Transonic Flow about Two Dimensional Airfoils by Relaxation Produres," AIAA J., Vol. 10, No. 1, pp. 49-54, Jan. 1972.
- 2.4-4. Jameson, A., "Iterative Solution of Transonic Flows over Airfoils and Wings, Including Flows at Mach 1," Comm. on Pure and Applied Math., Vol. 27, pp. 283-309, 1974.

- 2.4-5. South, J. C., Jr., and Jameson, A., "Relaxation Solutions for Inviscid Axisymmetric Transonic Flow over Blunt or Pointed Bodies," Proc. AIAA Computational Fluid Dynamics Conf., Palm Springs, Calif., pp. 8-17, July 1973.
- 2.4-6. Carlson, L. A., "Transonic Airfoil Design Using Cartesian Coordinates," NASA CR-2577.
- 2.4-7. Ludford, G. S. S., "The Behavior at Infinity of the Potential Functions of a Two-Dimensional Subsonic Compressible Flow," J. Math. Phys., Vol. 30, No. 3, pp. 117-130, October 1951.
- 2.4-8. Klunker, E. B., "Contribution to Methods for Calculating the Flow About Thin Lifting Wings at Transonic Speeds-Analytic Expressions for the Far Field," NASA TN D-6530, Nov. 1971.
- 2.4-9. Garabedian, P. R., "Estimation of the Relaxation Factor For Small Mesh Size," Math Tables and Aids to Comp., pp. 183-185, Oct. 1956.
- 2.4-10. Nash, J. F. and Macdonald, A. G. J., "The Calculation of Momentum Thickness in a Turbulent Boundary Layer at Mach Numbers up to Unity," Aero. Res. Council C.P. No. 963, 1967.
- 2.4-11. Bauer, F., Garabedian, P., Korn, D., and Jameson, A., "Supercritical Wing Sections II," Springer-Verlag, New York, 1975.
- 2.4-12. Bavitz, P. C., "An Analysis Method for Two-Dimensional Transonic Viscous Flow," NASA TN-D-7718, January 1975.
- 2.4-13. Carlson, L. A., "Transonic Airfoil Flowfield Analysis Using Cartesian Coordinates," NASA CR-2577, Aug. 1975.
- 2.4-14. Carlson, L. A., "Trandes: A FORTRAN Program For Transonic Airfoil Analysis and Design," NASA CR-2821, June 1977.
- 2.4-15. Barnwell, R. W., "A Potential-Flow/Boundary Layer Method for Calculating Subsonic and Transonic Airfoil Flows with Trailing Edge Separation," NASA TM-81850, June 1981.
- 2.4-16. Carlson, L. A., "TRANSEP: A Program for High Lift Separated Flow About Airfoils," NASA CR-3376, 1980.
- 2.4-17. Pinkerton, R. M., Greenberg, H., "Aerodynamic Characteristics of a Large Number of Airfoils Tested in a Variable Density Wind Tunnel," NACA TR 628, 1938.
- 2.4-18. Haines, A. B. and Monaghan, R. J., "High Speed Lift and Drag Data for Propeller Performance Calculations," R&M No. 2036, British A.R.C., 1945.

- 2.4-19. MacDougall, "Revised High-Speed Lift and Drag Data for Propeller Performance Calculations," R&M No. 2474, British A.R.C, 1947.
- 2.4-20. Lindsey, W. F., Stevenson, D. B., and Daley, B. N., "Aerodynamic Characteristics of 24 NACA-16 Series Airfoils at Mach Numbers Between 0.3 and 0.8," NACA TN 1546.
- 2.4-21. Cooper, J. P., "The Linearized Inflow Propeller Strip Analysis," WADC TR TRW 56-615, February 1957.
- 2.5-1. Ffowcs Williams, J. E. and Hawkins, D. L., "Sound Generated by Turbulence and Surfaces in Arbitrary Motion," Phil. Trans Roy. Soc. (London), A264, pp. 321-342, 1969.
- 2.5-2. Farassat, F., "Theory of Noise Generation from Moving Bodies with an Application of Helicopter Rotors," NASA TR R-451, 1975.
- 2.5-3. Farassat, F., "Linear Acoustic Formulas for Calculation of Rotating Blade Noise," AIAA Journal 19, pp. 1122-1130, 1981.
- 2.5-4. Nystrom, P. A. and Farassat, F., "A Numerical Technique for Calculation of the Noise of High-Speed Propellers with Advanced Blade Geometry," NASA TP 1662, 1980.
- 2.5-5. Woan, C. J. and Gregorek, G. M., "The Exact Numerical Calculation of Propeller Noise," AIAA Paper 78-1122, 1978.
- 2.5-6. Lawson, M. V., "The Sound Field for Singularities in Motion," Proc. Roy. Soc. (London) A286, pp. 559-572, 1965.
- 2.5-7. Succi, G. P., "The Design of Quiet Efficient Propellers," SAE Paper 790584, 1979.
- 2.5-8. Gregorek, G. M., "Five Case Studies of the Hartzell Baseline Propeller," Ohio State University Grant NSG-3247, December 1982.
- 2.6-1. Wempner, G., Mechanics of Solids With Applications to Thin Bodies, 1st edition, McGraw-Hill Book Co., New York, N.Y., 1973.
- 2.6-2. Hodges, D.H., and Dowell, E. H., "Nonlinear Equations of Motion for the Elastic Bending and Torsion of Twisted Nonuniform Rotor Blades," NASA TN D-7818, December 1974.
- 2.6-3. Rosen, A., and Friedmann, P. P., "Nonlinear Equations of Equilibrium for Elastic Helicopter or Wind Turbine Blades Undergoing Moderate Deflection", NASA CR-159478, December 1978.
- 2.6-4. Love, A. E. H., A Treatise on the Mathematical Theory of Elasticity, 1st edition, Cambridge University Press, 1927.
- 2.6-5. Timoshenko, S., and Goodier, J. N., Theory of Elasticity, 1st edition, McGraw-Hill Book Co., New York, N.Y., 1951.

- 2.6-6. Wang, C. T., Applied Elasticity, 1st edition, McGraw-Hill Book Co., New York, N.Y., 1953.
- 2.6-7. Oden, J. T., and Ripperger, E. A., Mechanics of Elastic Structures, 2nd edition, McGraw-Hill Book Co., New York, N.Y., 1981
- 2.6-8. Boresi, A. P., Sidebottom, O. M., Seely, F. B., and Smith, J. O., Advanced Mechanics of Materials, 3rd edition, John Wiley and Sons, Inc., New York, N.Y., 1978.
- 2.6-9. Alkire, K., "An Analysis of Rotor Blade Twist Variables Associated With Different Euler Angle Sequences and Pretwist Treatments," NASA TM-84394, May 1984.
- 2.6-10. Green, A.E., and Zerna, W., Theoretical Elasticity, 2nd edition, Oxford Press, 1968.
- 2.6-11. Rosen, A., "Theoretical and Experimental Investigation of the Nonlinear Torsion and Extension of Initially Twisted Bars," ASME Journal of Applied Mechanics, Vol. 50, pp. 321-326, 1983.
- 2.6-12. Rosen, A., "The Effect of Initial Twist on the Torsional Rigidity of Beams - Another Point of View," ASME Journal of Applied Mechanics, Vol. 47, pp. 389-392, June 1980.
- 2.6-13. Hodges, D.H., "Torsion of Pretwisted Beams Due to Axial Loading," ASME Journal of Applied Mechanics, Vol. 47, pp. 393-397, June 1980.
- 2.6-14. Brunelle, E.J., "Dynamical Torsion Theory of Rods Deduced from the Equations of Linear Elasticity," AIAA Journal, Vol. 10, No. 4, pp. 524-526, 1972.
- 2.6-15. Rosen, A., "Discussion of Reference 36 and Author's Closure", ASME Journal of Applied Mechanics, Vol. 48, pp.679-681, 1981.
- 2.6-16. Shield, R.T., "Extension and Torsion of Elastic Bars With Initial Twist," ASME Journal of Applied Mechanics, Vol. 49, pp. 779-786, December 1982.
- 2.6-17. Krenk, S., and Gunneskov, O., "Pretwist and Shear Flexibility in the Vibrations of Turbine Blades," ASME Journal of Applied Mechanics, Vol. 52, pp. 409-415, 1985.
- 2.6-18. Reissner, E., "A Variational Analysis of Small Finite Deformation of Pretwisted Elastic Beams," Int. J. Solids Structures, Vol. 21, No. 7, pp. 773-779, 1985.
- 2.6-19. Kaza, K. K., and Kielb, R. E., "Effects of Warping and Pretwist on Torsional Vibration of Rotating Beams," ASME Journal of Applied Mechanics, July 1983.

- 2.6-20. Tauchert, T. R., Energy Principles in Structural Mechanics, 1st edition, McGraw-Hill Book Co., New York, N.Y., 1974.
- 2.6-21. Oden, J. T., Finite Elements of Nonlinear Continua, 1st edition, McGraw-Hill Book Co., New York, N.Y., 1972.
- 2.6-22. Friedmann, P. P., and Straub, F. K., "Application of the Finite Element Method to Rotary Wing Aeroelasticity", Journal of the American Helicopter Society, Vol. 25, pp. 36-44, January 1980.
- 2.6-23. Straub, F. K., and Friedmann, P. P., "A Galerkin Type Finite Element for Rotary-Wing Aeroelasticity in Hover and Forward Flight," Vertica, Vol. 5, pp. 75-98, 1981.
- 2.6-24. Bathe, K., Finite Element Procedures in Engineering Analysis, 1st edition, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1982.
- 2.6-25. Zienkiewicz, O. C., The Finite Element Method, McGraw-Hill Book Co., New York, N.Y., 1977.
- 2.6-26. Gyan, R., "Reduction of Stiffness and Mass Matrices", AIAA Journal, Vol. 3, No.2, pp. 380, February 1965.
- 2.6-27. Brown, K. M., "Computer Oriented Methods for Fitting Tabular Data in a Linear and Nonlinear Least Squares Sense," Report TR 72-13, University of Minnesota, Dept. of Computer, Information, and Control Sciences, 1972.
- 2.6-28. Sivaneri, N. T., and Chopra, I., "Dynamic Stability of a Rotor Blade Using Finite Element Analysis," AIAA Journal, Vol. 20, No. 5, pp. 716-723, 1984.
- 2.6-29. Tillerson, J. R., Stricklin, J. A., and Haisler, W. E., "Numerical Methods for the Solution of Nonlinear Problems in Structural Analysis," Numerical Solution of Nonlinear Structural Problems, ASME, 1973.
- 2.6-30. Meirovitch, L., "A New Method of Solution of the Eigenvalue Problem for Gyroscopic Systems," AIAA Journal, Vol. 12, No. 10, 1337-1342, October 1974.
- 2.6-31. Love, A. E. H., A Treatise on the Mathematical Theory of Elasticity, 1st edition, Cambridge University, 1927.
- 2.6-32. Timoshenko, S. and Goodier, J. N., Theory of Elasticity, 1st edition, McGraw-Hill Book Company, New York, N.Y., 1951.
- 2.6-33. Gallagher, R. H., Finite Element Analysis Fundamentals, 1st edition, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1972.
- 2.6-34. Zienkiewicz, O. C., The Finite Element Method, McGraw-Hill Book Company, New York, N.Y., 1977.

- 2.6-35. Hodges, D.H., and Dowell, E. H., "Nonlinear Equations of Motion for the Elastic Bending and Torsion of Twisted Nonuniform Rotor Blades," NASA TN D-7818, December 1974.
- 2.6-36. Rosen, A. and Friedmann, P. P., "Nonlinear Equations of Equilibrium for Elastic Helicopter or Wind Turbine Blades Undergoing Moderate Deflections," NASA CR-159478, December 1978.
- 2.6-37. Roark, R. J., and Young, W. C., Formulas for Stress and Strain, 5th edition, McGraw-Hill Book Co., New York N.Y., 1975.
- 2.6-38. Blevins, R. D., Formulas for Natural Frequency and Mode Shape, Van Nostrand Reinhold Co., New York, N.Y., 1979.
- 2.6-39. Rosen, A., and Friedmann, P. P., "The Nonlinear Behavior of Elastic Slender Straight Beams Undergoing Small Strains and Moderate Rotations," Journal of Applied Mechanics, pp. 161-168, March 1979.





# Report Documentation Page

1. Report No. CR-185277		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle Generalized Advanced Propeller Analysis System (GAPAS) - Volume 2 - Computer Program User Manual				5. Report Date December 1986	
				6. Performing Organization Code	
7. Author(s) Leslie Glatt                      Rudy J. Swigart Donald R. Crawford              Eric W. Wong John B. Kosmatka				8. Performing Organization Report No.	
				10. Work Unit No.	
9. Performing Organization Name and Address TRW Space & Defense One Space Park Redondo Beach, CA 90278				11. Contract or Grant No. NAS3-22251	
				13. Type of Report and Period Covered Contractor Report	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration NASA Lewis Research Center Cleveland, Ohio 44135				14. Sponsoring Agency Code	
				15. Supplementary Notes	
16. Abstract <p>This report describes the Generalized Advanced Propeller Analysis System (GAPAS) computer code. GAPAS has been developed to analyze advanced technology multi-bladed propellers which operate on aircraft with speeds up to Mach 0.8 and altitudes up to 40,000 feet. GAPAS includes technology for analyzing aerodynamic, structural and acoustic performance of propellers. The computer code has been developed for the CDC 7600 computer and is currently available for industrial use on the NASA Langley computer.</p> <p>A description of all the analytical models incorporated in GAPAS is included. Sample calculations are also described as well as users requirements for modifying the analysis system. Computer system core requirements and running times are also discussed.</p>					
17. Key Words (Suggested by Author(s)) Advanced Technology Propellers Unified Analysis System Computer Program Aerodynamics Acoustics Structures			18. Distribution Statement		
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of pages 582	22. Price





