NASA/CR—2010-216898/VOL1

Broadband Fan Noise Prediction System for Turbofan Engines

Volume 1: Setup_BFaNS User’s Manual and Developer’s Guide

Bruce L. Morin
Pratt & Whitney, East Hartford, Connecticut

November 2010

NASA/CR—2010-216898/VOL1

Broadband Fan Noise Prediction System for Turbofan Engines

Volume 1: Setup_BFaNS User's Manual and Developer's Guide

Bruce L. Morin
Pratt & Whitney, East Hartford, Connecticut

Prepared under Contract NAS3–27727

National Aeronautics and
Space Administration

Glenn Research Center
Cleveland, Ohio 44135

November 2010

Acknowledgments

This work was performed under Contract NAS3–27727 (AST Task 13) from NASA Glenn Research Center (GRC) with Dennis Huff as contract monitor. The author greatly appreciates the following contributions: Dennis Huff (GRC) and Edmane Envia (GRC) for overall direction and coordination the AST Noise Program; Donald Hanson (P&W, retired) and Stewart Glegg (Florida Atlantic University) for providing the noise prediction routines used in BFaNS; Ramons Reba (United Technologies Research Center) for his work on the cascade-response subroutines; Gary Podboy (GRC), Richard Woodward (GRC) and Christopher Hughes (GRC) for providing noise, performance and flow-field data from the Pratt & Whitney and General Electric 22 in. diameter fan rigs that were tested at GRC; Ulrich Ganz (Boeing), John Premo (Boeing) and Timothy Patten (Boeing) for providing noise, performance and flow-field data from the Boeing 18 in. diameter fan rig that was tested at Boeing; Tony Hoang (United Technologies Research Center) for his assistance with computer programming and running test cases; Jon Gilson (P&W), Aaron Farbo (P&W Intern) and Gary Willett (P&W retired) for their assistance with data reduction and analysis; Clint Ingram (formerly P&W) and Mark Stephens (P&W) for providing CFD predictions for the Pratt & Whitney 22 in. diameter fan rig; and Wesley Lord (P&W), Douglas Mathews (P&W) and David Topol (P&W) for their helpful comments and suggestions regarding this work

Trade names and trademarks are used in this report for identification only. Their usage does not constitute an official endorsement, either expressed or implied, by the National Aeronautics and Space Administration.

This work was sponsored by the Fundamental Aeronautics Program at the NASA Glenn Research Center.

Level of Review: This material has been technically reviewed by expert reviewer(s).

Available from

NASA Center for Aerospace Information
7115 Standard Drive
Hanover, MD 21076–1320

National Technical Information Service
5301 Shawnee Road
Alexandria, VA 22312

Available electronically at http://gltrs.grc.nasa.gov

Summary

Pratt & Whitney has developed a Broadband Fan Noise Prediction System (BFaNS) for turbofan engines. This system computes the noise generated by turbulence impinging on the leading edges of the fan and fan exit guide vane, and noise generated by boundary-layer turbulence passing over the fan trailing edge. BFaNS has been validated on three fan rigs that were tested during the NASA Advanced Subsonic Technology Program (AST). The predicted noise spectra agreed well with measured data. The predicted effects of fan speed, vane count, and vane sweep also agreed well with measurements.

The noise prediction system consists of two computer programs: Setup_BFaNS and BFaNS. Setup_BFaNS converts user-specified geometry and flow-field information into a BFaNS input file. From this input file, BFaNS computes the inlet and aft broadband sound power spectra generated by the fan and FEGV. The output file from BFaNS contains the inlet, aft and total sound power spectra from each noise source.

This report is the first volume of a three-volume set documenting the Broadband Fan Noise Prediction System:

Volume 1: Setup_BFaNS User’s Manual and Developer’s Guide
Volume 2: BFaNS User’s Manual and Developer’s Guide
Volume 3: Validation and Test Cases

The present volume begins with an overview of the Broadband Fan Noise Prediction System, followed by step-by-step instructions for installing and running Setup_BFaNS. It concludes with technical documentation of the Setup_BFaNS computer program.

Contents

List of Figures

List of Tables

Broadband Fan Noise Prediction System for Turbofan Engines
Volume 1: Setup_BFaNS User's Manual and Developer's Guide

Bruce L. Morin
Pratt & Whitney
East Hartford, Connecticut 06108

1.0 Introduction

In early turbofan engines, the fan blade passage tone and higher harmonics dominated the fan noise level when compared to fan broadband noise. However, engine makers have been very successful at reducing fan tone noise by decreasing fan tip speed, providing ample spacing between airfoil rows, carefully selecting the airfoil counts, and acoustically treating the fan duct. In fact, tone noise reduction has been so great that fan broadband noise is now a major contributor to the overall engine noise level.

Under NASA funding, Pratt & Whitney has developed a Broadband Fan Noise Prediction System (BFaNS) for turbofan engines. This prediction system is built around acoustic theories developed by Hanson (Refs. 1 to 4) and Glegg (Refs. 5 to 7). These theories account for noise generated by turbulence impinging on the leading edges of the fan and fan exit guide vane (FEGV), and noise generated by boundary-layer turbulence passing over the fan trailing edge.

Figure 1 shows the fan broadband noise sources that can be calculated with BFaNS. The blade tip-noise and blade self-noise calculations are limited to subsonic blade-tip relative Mach numbers. There are no limitations on the other sources.

This report begins with an overview of the Broadband Fan Noise Prediction System, followed by step-by-step instructions for installing and running Setup_BFaNS. It concludes with technical documentation of the Setup_BFaNS computer program. Volume 2 of this report series provides instructions for running BFaNS, and Volume 3 provides validation and test cases.

2.0 General Information

This section provides general information about the Broadband Fan Noise Prediction System, including the orientation of the coordinate system, the velocity and angle conventions, and the conventions used to describe the flow path and airfoil geometries.

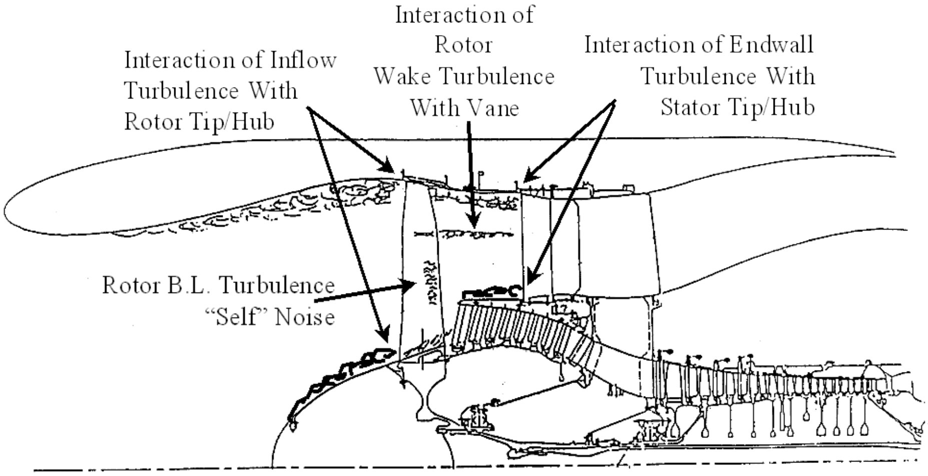


Figure 1.—Fan broadband noise sources predicted by BFaNS.

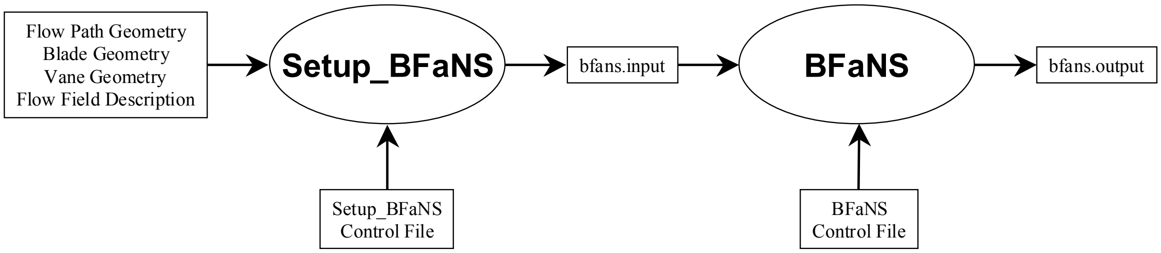
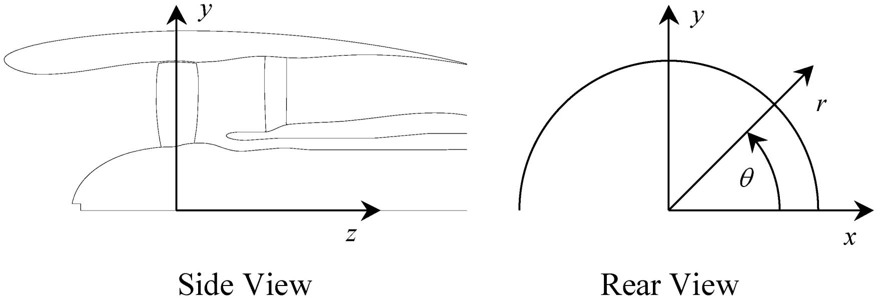


Figure 2.—Organization of BFaNS.



Side View

Rear View

Figure 3.—Engine coordinate system.

2.1 Overview

Figure 2 shows the organization of the Broadband Fan Noise Prediction System, which consists of two computer programs: Setup_BFaNS and BFaNS. Setup_BFaNS converts user-provided geometry and flow-field information into a BFaNS input file (bfans.input). From this input file, BFaNS computes the inlet and aft broadband sound power spectra generated by the fan and FEGV. The output file (bfans.output) contains the inlet, aft and total sound power spectra from each noise source shown in Figure 1, along with the combined sound power spectra from all sources. Both Setup_BFaNS and BFaNS use control files (also referred to as author files) that allow the user to modify the way the programs operate.

2.2 Engine Coordinate System

Figure 3 shows the engine coordinate system used in Setup_BFaNS and BFaNS. The *z*-direction is measured along the engine axis with *z*-values increasing from the inlet toward the exit. The axial location of the origin is arbitrary (for convenience, it is shown at the blade stacking line). When viewed from the rear, the *x*-axis is horizontal with values increasing toward the right. Likewise, the *y*-axis is vertical with values increasing in the upward direction. The corresponding cylindrical coordinate system is defined as follows:

$$r = \sqrt{x^2 + y^2}$$

$$\tan\theta = \frac{y}{x}$$

Defined in this manner, θ increases in the counter-clockwise direction when viewed from the rear.

2.3 Velocity Conventions

Figure 4 shows the velocity conventions used in Setup_BFaNS and BFaNS. The conversion from the stationary reference frame to the rotating reference frame is:

$$\vec{W} = \vec{C} - \vec{U}$$

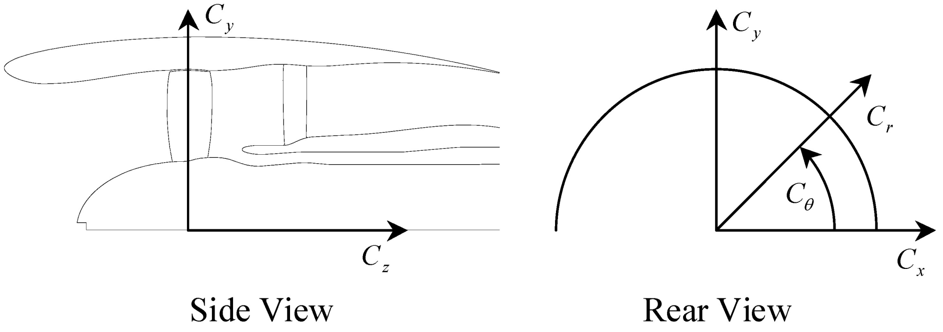


Figure 4.—Velocity conventions.

From this conversion formula, we get the following relationships:

$$W_z = C_z$$
$$W_r = C_r$$
$$W_\theta = C_\theta - U$$

Note that U is positive when the fan rotates counter-clockwise viewed from the rear, and negative when the fan rotates clockwise viewed from the rear. The meridional velocity component (which is the same in both reference frames) is defined as:

$$C_m = W_m = \sqrt{C_z^2 + C_r^2}$$

2.4 Flow-Angle Definitions

The flow angles are computed from:

$$\tan\beta = \frac{W_\theta}{W_m} \qquad \tan\alpha = \frac{C_\theta}{C_m}$$
$$\tan\phi = \frac{C_r}{C_z} = \frac{W_r}{W_z}$$

The α and β angles are measured from the meridional direction to facilitate stripwise application of the acoustic theories used in BFaNS (see Volume 2: BFaNS User's Manual and Developer's Guide).

2.5 Flow-Path Coordinates

Figure 5 shows the side view of a fan stage, along with some of the important terminology used in Setup_BFaNS and BFaNS. The flow path is axisymmetric and consists of the hub surface, shroud surface, upper splitter surface and lower splitter surface. These surfaces must be defined in terms of their (*r,z*) coordinates. Specific points of interest are: the axial location of the hilite, the nose-cone leading edge, the splitter leading edge, the duct exit, and the blade and vane trailing-edge tips. Stations 1, 2, 3, and 4 indicate the locations where flow-field information is available (via experiment or prediction). Flow-field information must be available upstream from the blade leading edge, downstream from the blade trailing edge (but upstream from the splitter leading edge), upstream from the vane leading edge (but downstream from the splitter leading edge), and downstream from the vane trailing edge.

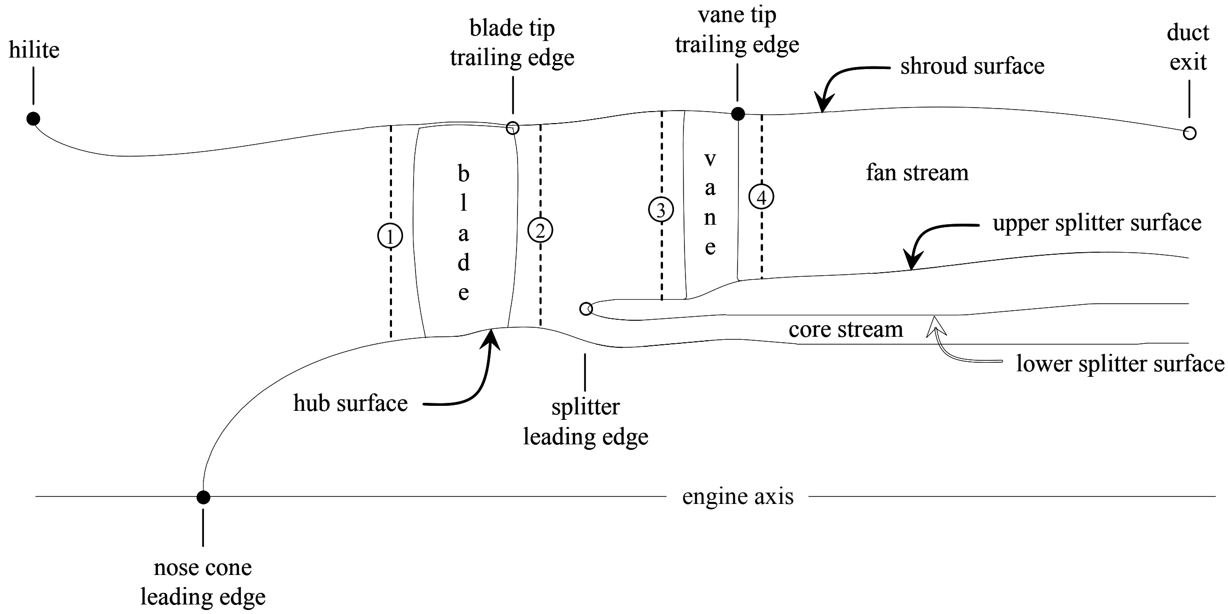


Figure 5.—Flow-path terminology.

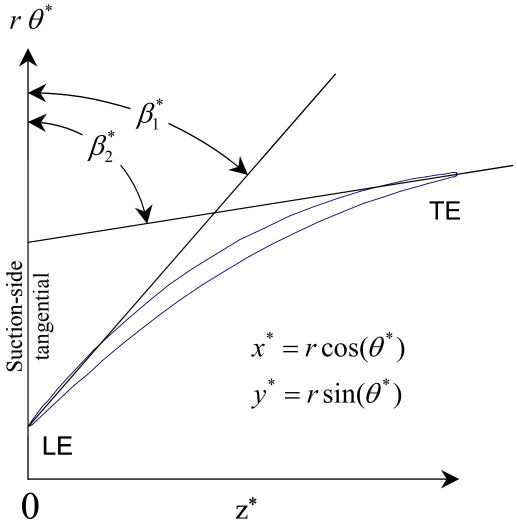


Figure 6.—Airfoil coordinate system.

2.6 Airfoil Coordinates

Figure 6 shows the coordinate system used to define airfoil geometry. The leading and trailing edge metal angles are measured clockwise from the suction-side tangential; therefore, the metal angles are always positive. The asterisks on x, y and θ indicate that the x^*y^*-plane may be rotated relative to the xy-plane shown in Figure 3. Likewise, the asterisk on z indicates that the axial location of the x^*y^*-plane may be translated relative to the xy-plane shown in Figure 3. Defining the airfoil in this manner allows the blade and vane geometry to be defined independently from each other, and independently from the flow path. Section 4.2.2 describes the coordinate transformations used by Setup_BFaNS to ensure that the blade and vane are properly located in the flow path, and are properly oriented relative to the direction of rotation.

2.7 Sweep and Lean Definitions

Figure 7 shows a side view and rear view of an airfoil. In Setup_BFaNS, the leading edge is described parametrically as a function of radius (z_e and θ_{le} versus r). Therefore, the sweep and lean are computed from,

$$\tan\tilde{\psi}_S = \frac{dz_{le}}{dr}$$

$$\tan\tilde{\psi}_L = r\frac{d\theta_{le}}{dr}$$

Both of these quantities depend on radius. The overall sweep and lean are computed from:

$$\tan\tilde{\Psi}_S = \frac{\Delta z_{le}}{\Delta r}$$

$$\tan\tilde{\Psi}_L = \bar{r}\frac{\Delta\theta_{le}}{\Delta r}$$

where the Δ is applied from the hub to the tip, and $\bar{r}$ is the mean radius. Sections 4.8.5 and 4.8.6 provide details of the parametric description for the airfoil leading edge.

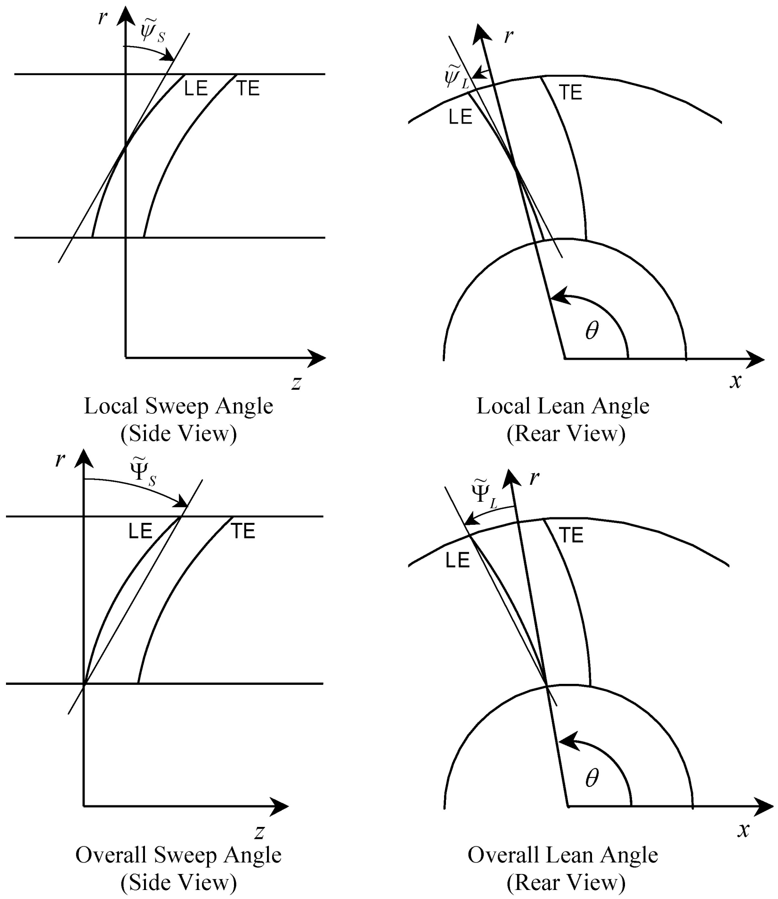


Figure 7.—Sweep and lean conventions.

3.0 Installation and Execution of Setup_BFaNS

This section provides instructions for installing and running Setup_BFaNS. Refer to Section 4.0 for technical documentation of the computer program.

3.1 Installation

0 provides instructions for installing the Broadband Fan Noise Prediction System (Setup_BFaNS and BFaNS) on Unix-based computers. Both computer programs are written in FORTRAN 77. Appendix C explains how to run the Setup_BFaNS test case.

3.2 Run Procedure

Setup_BFaNS converts user-specified geometry and flow-field information into a BFaNS input file (bfans.input). Table 1 shows the step-by-step procedure for running Setup_BFaNS. Sections 3.2.1 through 3.2.10 describe the details of each step.

TABLE 1.—RUN PROCEDURE FOR SETUP_BFANS

Step 1	Create directory structure
Step 2	Create flow-path geometry file
Step 3	Create blade and vane geometry files
Step 4	Create flow-field files
Step 5	Create Setup_BFaNS input file
Step 6	Create Setup_BFaNS author file (optional)
Step 7	Execute Setup_BFaNS
Step 8	Examine boundary-layer characteristics
Step 9	Examine wake characteristics
Step 10	Examine output file

3.2.1 Create Directory Structure

The user must create a directory that contains the following sub-directories:

- A sub-directory that contains the files that describe the geometry of the flow path, the blade, and the vane. The default name of this sub-directory is **bfans_geometry**, but the user can over-ride this name in the Setup_BFaNS author file (see Section 3.2.6, Card 5002).
- (Optional) a sub-directory that contains the files that describe the inviscid streamline-predicted flow field near the blade leading edge, blade trailing edge, vane leading edge, and vane trailing edge for each configuration and operating point. The default name of this sub-directory is **bfans_sline**, but the user can over-ride this name in the Setup_BFaNS author file (see Section 3.2.6, Card 5004).
- (Optional) a sub-directory that contains the files that describe the viscous CFD-predicted flow field near the blade leading edge, blade trailing edge, vane leading edge, and vane trailing edge for each configuration and operating point. The default name of this sub-directory is **bfans_cfd**, but the user can over-ride this name in the Setup_BFaNS author file (see Section 3.2.6, Card 5006).
- (Optional) a sub-directory that contains the files that describe the measured flow field near the blade leading edge, blade trailing edge, vane leading edge, and vane trailing edge for each configuration and operating point. The default name of this sub-directory is **bfans_data**, but the user can over-ride this name in the Setup_BFaNS author file (see Section 3.2.6, Card 5008).
- A sub-directory (or several sub-directories) where Setup_BFaNS will be run. These sub-directories are referred to as working sub-directories. Typically, separate working sub-directories are created for each operating condition (e.g., approach, cutback, and sideline).

3.2.2 Create Flow-Path Geometry File

The user must create a file that contains the (*r,z*) coordinates of the hub, the shroud, the upper splitter, and the lower splitter surfaces (see Figure 5). The axial coordinates can be measured from any reference point, though the same reference point must be used for each surface. The length of the filename must be less than or equal to 30 characters. This file name is specified in the Setup_BFaNS input file (see Section 3.2.5, block 1). The flow-path geometry file must be located in **bfans_geometry** or the directory specified in the Setup_BFaNS author file (see Section 3.2.6, Card 5002).

Table 2 explains the file format for the flow-path geometry file. Appendix D shows a sample file along with a section of FORTRAN code used to read the file. Note that this file contains 27 "blocks" of input. Throughout this report, a "block" refers to one or more variables entered on a single line of input (e.g., block 1), or a single array variable entered on several lines of input (e.g., block 4).

TABLE 2.—FORMAT OF FLOW-PATH GEOMETRY FILE

Block	Variable Name	Type	Format	Description
1	`filedescr`	`character`	`a50`	`description of data in blocks 2 - 6`
2	`nduct_h`	`integer`	`free`	`number of points along hub`
3	`parmname`	`character`	`a50`	`parameter name for block 4`

4	`zhub(i)`
	`i=1,nduct_h`
		`real`	`free`	`hub axial coordinates (in)`

5	`parmname`	`character`	`a50`	`parameter name for block 6`

6	`rhub(i)`
	`i=1,nduct_h`
		`real`	`free`	`hub radial coordinates (in)`

7	`blank`	`character`	`a1`	`blank line`
8	`filedescr`	`character`	`a50`	`description of data in blocks 9 - 13`
9	`nsplt_u`	`integer`	`free`	`number of points along upper splitter`
10	`parmname`	`character`	`a50`	`parameter name for block 11`

11	`zsup(i)`
	`i=1,nsplt_u`
		`real`	`free`	`upper splitter axial coordinates (in)`

12	`parmname`	`character`	`a50`	`parameter name for block 13`

13	`rsup(i)`
	`i=1,nsplt_u`
		`real`	`free`	`upper splitter radial coordinates (in)`

14	`blank`	`character`	`a1`	`blank line`
15	`filedescr`	`character`	`a50`	`description of data in blocks 16 - 20`
16	`nsplt_l`	`integer`	`free`	`number of points along lower splitter`
17	`parmname`	`character`	`a50`	`parameter name for block 18`

18	`zslo(i)`
	`i=1,nsplt_l`
		`real`	`free`	`lower splitter axial coordinates (in)`

19	`parmname`	`character`	`a50`	`parameter name for block 20`

20	`rslo(i)`
	`i=1,nsplt_l`
		`real`	`free`	`lower splitter radial coordinates (in)`

21	`blank`	`character`	`a1`	`blank line`
22	`filedescr`	`character`	`a50`	`description of data in blocks 23 - 27`
23	`nduct_t`	`integer`	`free`	`number of points along shroud`
24	`parmname`	`character`	`a50`	`parameter name for block 24`

25	`ztip(i)`
	`i=1,nduct_t`
		`real`	`free`	`shroud axial coordinates (in)`

26	`parmname`	`character`	`a50`	`parameter name for block 27`

27	`rtip(i)`
	`i=1,nduct_t`
		`real`	`free`	`hub radial coordinates (in)`

3.2.3 Create Blade and Vane Geometry Files

The user must create a file that contains the *(x*, y*,z*)* coordinates of the leading and trailing edges of the blade, along with the corresponding metal angles (see Section 2.6 for details). The user must create a similar file for the vane. The axial coordinates can be measured from an arbitrary reference point (the reference can be different for the blade and the vane, and can also be different than the reference used to describe the flow path). The length of the filenames must be less than or equal to 30 characters. These file names are specified in the Setup_BFaNS input file (see Section 3.2.5, blocks 8 and 11). These files must be located in **bfans_geometry** or the directory specified in the Setup_BFaNS author file (see Section 3.2.6, Card 5002).

Table 3 explains the format of the blade and vane geometry files. Appendix E and Appendix F show sample blade and vane geometry files along with sections of FORTRAN code used to read these files.

TABLE 3.—FORMAT OF BLADE AND VANE GEOMETRY FILES

Block	Variable Name	Type	Format	Description
1	filedesc	character	a50	file description
2	numrle	integer	free	number of points defining blade or vane
3	parmname	character	a50	parameter name for block 4

4	xle(i)
	i=1,numrle
		real	free	l.e. x-coordinates (in)

5	parmname	character	a50	parameter name for block 6

6	yle(i)
	i=1,numrle
		real	free	l.e. y-coordinates (in)

7	desc	character	a50	parameter name for block 8

8	zle(i)
	i=1,numrle
		real	free	l.e. z-coordinates (in)

9	parmname	character	a50	parameter name for block 10

10	betale(i)
	i=1,numrle
		real	free	l.e. metal angle (deg)

11	parmname	character	a50	parameter name for block 12

12	xte(i)
	i=1,numrle
		real	free	t.e. x-coordinates (in)

13	parmname	character	a50	parameter name for block 14

14	yte(i)
	i=1,numrle
		real	free	t.e. y-coordinates (in)

15	parmname	character	a50	parameter name for block 16

16	zte(i)
	i=1,numrle
		real	free	t.e. z-coordinates (in)

17	parmname	character	a50	parameter name for block 18

18	betate(i)
	i=1,numrle
		real	free	t.e. metal angle (deg)

3.2.4 Create Flow-Field Files

The user must create files that describe the flow field at the following four locations (see Figure 5):

- Station no. 1—Upstream from the blade leading edge
- Station no. 2—Downstream from the blade trailing edge, but upstream from the splitter leading edge
- Station no. 3—Upstream from the vane leading edge, but downstream from the splitter leading edge
- Station no. 4—Downstream from the vane trailing edge

The flow-field information can be provided by any combination of predictions (e.g., an inviscid streamline deck, or viscous CFD) or experimental measurements. Table 4 explains the file format used for viscous CFD and measured data, and Appendix G shows the section of FORTRAN code used to read these files.

and 0 show similar information for inviscid streamline predictions.

The length of each filename must be less than or equal to 30 characters. These names are specified the Setup_BFaNS input file (see Section 3.2.5, blocks 13, 15, 17 and 19). The flow-field files must be located in **bfans_sline**, **bfans_data**, **bfans_cfd** or the directories specified in the Setup_BFaNS author file (see Section 3.2.6, Cards 5004, 5006, or 5008).

For viscous CFD predictions and experimental data, the axial coordinates must be measured from the blade trailing-edge tip. The blade trailing-edge tip was chosen as the reference point because of its convenience for experimentally measured data. For inviscid streamline predictions, the axial coordinates must be measured from the same reference that is used when defining the flow path.

The velocity, total pressure and total temperature can be specified at different locations. This feature is built into the program because the total-pressure and total-temperature measurements are often taken at a different axial location than the velocity measurements (e.g., see Volume 3: Validation and Test Cases). The rotational speed must be consistent with the direction of rotation specified in the Setup_BFaNS input file (see Section 3.2.5, block 10). If the fan is rotating counter-clockwise viewed from the rear, then the rotational speed must be positive. Likewise, if the fan is rotating clockwise viewed from the rear, then the rotational speed must be negative. The program will check for this consistency, and will attempt to correct any errors. However, this check is not foolproof. The user should closely examine the velocity triangles that are displayed on the screen when Setup_BFaNS is executed.

The contents of the viscous CFD or measured data files can be viewed with a MATLAB (The MathWorks, Inc.) utility provided with Setup_BFaNS. This utility consists of three files (reduce.m, stress.m, and only_plot.m) located in the directory that contains the Setup_BFaNS source code. At the MATLAB prompt, type "reduce" and respond to the questions.

TABLE 4.—FORMAT OF FLOW-FIELD FILES
[Viscous CFD prediction or measured data]

Block	Variable Name	Type	Format	Description
1	iwaketyp	integer	free	not used
2	omega	real	free	rotor rotational speed (rpm)
	xprtip	real	free	axial distance from the rotor trailing edge tip to the velocity measurement plane tip location (inches)
	rhub	real	free	inner duct radius at the velocity measurement plane (inches)
	rtip	real	free	outer duct radius at the velocity measurement plane (inches)
	npassage	integer	free	not used
	phipr	real	free	not used
3	numr	integer	free	number of radial measurement points for velocities
	numt	integer	free	number of tangential measurement points for velocities

4	z(i)
	i=1,numr
		real	free	axial coordinates for velocities (inches) measured from blade tip trailing edge

5	x(i,j)
	i=1,numr
	j=1,numt
		real	free	x-coordinates for velocities (inches)

6	y(i,j)
	i=1,numr
	j=1,numt
		real	free	y-coordinates for velocities (inches)

7	icz	integer	free	=1 if cz is in file, =0 if not
	icu	integer	free	=1 if cu is in file, =0 if not
	icr	integer	free	=1 if cr is in file, =0 if not
	itke	integer	free	=1 if tke is in file, =0 if not
	ieps	integer	free	=1 if eps is in file, =0 if not
	icpzcpz	integer	free	=1 if cpzcpz is in file, =0 if not
	icpucpu	integer	free	=1 if cpucpu is in file, =0 if not
	icprcpr	integer	free	=1 if cprcpr is in file, =0 if not
	icpzcpu	integer	free	=1 if cpzcpu is in file, =0 if not
	icpzcpr	integer	free	=1 if cpzcpr is in file, =0 if not
	icpucpr	integer	free	=1 if cpucpr is in file, =0 if not
	iscale	integer	free	=1 if scale is in file, =0 if not

8	cz(i,j)
	i=1,numr
	j=1,numt
		real	free	axial component of velocity (ft/sec)

9	cu(i,j)
	i=1,numr
	j=1,numt
		real	free	circumferential component of velocity (ft/sec)

10	cr(i,j)
	i=1,numr
	j=1,numt
		real	free	radial component of velocity (ft/sec)

TABLE 4.—FORMAT OF FLOW-FIELD FILES
[Viscous CFD prediction or measured data]

Block	Variable Name	Type	Format	Description

11	tke(i,j)
	i=1,numr
	j=1,numt
		real	free	turbulent kinetic energy (ft**2/sec**2)

12	eps(i,j)
	i=1,numr
	j=1,numt
		real	free	turbulent dissipation rate (ft**2/sec**3)

13	cpzcpz(i,j)
	i=1,numr
	j=1,numt
		real	free	z-z reynolds stress (ft**2/sec**2)

14	cpucpu(i,j)
	i=1,numr
	j=1,numt
		real	free	θ-θ reynolds stress (ft**2/sec**2)

15	cprcpr(i,j)
	i=1,numr
	j=1,numt
		real	free	r-r reynolds stress (ft**2/sec**2)

16	cpzcpu(i,j)
	i=1,numr
	j=1,numt
		real	free	z-θ reynolds stress (ft**2/sec**2)

17	cpzcpr(i,j)
	i=1,numr
	j=1,numt
		real	free	z-r reynolds stress (ft**2/sec**2)

18	cpucpr(i,j)
	i=1,numr
	j=1,numt
		real	free	θ-r reynolds stress (ft**2/sec**2)

19	scale(i)
	i=1,numr
		real	free	integral length scale (in)

20	ztip2	real	free	axial distance from the rotor trailing edge tip to the PT2,TT2 measurement plane (inches)
	rhub2	real	free	inner duct radius at the PT2,TT2 measurement plane (inches)
	rtip2	real	free	outer duct radius at the PT2,TT2 measurement plane (inches)
21	numr2	integer	free	number of radial measurement points for PT2, TT2

22	r2(i)
	i=1,numr2
		real	free	radial coordinate of radial measurement points for PT2, TT2 (inches)

23	z2(i)
	i=1,numr2
		real	free	axial coordinate relative to the rotor tip t.e. of radial measurement points for PT2, TT2 (inches)

24	pt2(i)
	i=1,numr2
		real	free	circumferentially averaged total pressure (psfa)

25	tt2(i)
	i=1,numr2
		real	free	circumferentially averaged total temperature (deg. R)

TABLE 5.—FORMAT OF FLOW-FIELD FILES
[Inviscid streamline prediction]

Block	Variable Name	Type	Format	Description
1	filedescr	character	a50	file description
2	numr	integer	free	number of radial measurement points
3	parmname	character	a50	parameter name for block 4
4	diam(i), i=1,numr	real	free	local diameter (in)
5	parmname	character	a50	parameter name for block 6
6	z(i), i=1,numr	real	free	axial coordinate (in) measured from flow path reference
7	parmname	character	a50	parameter name for block 8
8	u(i), i=1,numr	real	free	local blade speed (in)
9	parmname	character	a50	parameter name for block 10
10	cz(i), i=1,numr	real	free	axial component of velocity (ft/sec)
11	parmname	character	a50	parameter name for block 12

TABLE 5.—FORMAT OF FLOW-FIELD FILES
[Inviscid streamline prediction]

Block	Variable Name	Type	Format	Description
12	ct(i), i=1,numr	real	free	tangential component of velocity (ft/sec)
13	parmname	character	a50	parameter name for block 14
14	cr(i), i=1,numr	real	free	radial component of velocity
15	parmname	character	a50	parameter name for block 16
16	loss(i), i=1,numr	real	free	loss coefficient
17	parmname	character	a50	parameter name for block 18
18	gapchord(i), i=1,numr	real	free	local gap/chord ratio
19	parmname	character	a50	parameter name for block 20
20	pt2(i), i=1,numr	real	free	total pressure (psfa)
21	parmname	character	a50	parameter name for block 22
22	tt2(i), i=1,numr	real	free	total temperature (deg. r)

3.2.5 Create Setup_BFaNS Input File

The user must create a Setup_BFaNS input file in the working sub-directory (i.e., the sub-directory where Setup_BFaNS will be run). Table 6 explains the format of the Setup_BFaNS input file, and 0 shows a sample file.

When zero is entered in blocks 21, 22, 23 or 24 of the input file, the corresponding boundary-layer thickness will be calculated automatically. For every new case, the author recommends that the user enter zeroes in each of these blocks. After running Setup_BFaNS, the user should then confirm that the computed boundary layers are reasonable (see Section 3.2.8). If not, the boundary layers must be entered manually, and Setup_BFaNS must be run again. See Section 4.10.1 for details on how Setup_BFaNS estimates the boundary-layer thicknesses.

TABLE 6.—FORMAT OF SETUP_BFANS INPUT FILE

Block	Variable name	Type	Format	Description
1	filenam	character	a30	Specifies the name of the file that contains the flow path coordinates. The length of the filename must be less than or equal to 30 characters. This file must be located in the directory specified by Card 5002 of the Setup_BFaNS author file (note: the default geometry directory is **bfans_geometry).**
2	zblade	real	free	Specifies the axial coordinate (in inches) of the blade trailing edge tip. This coordinate must be measured from the same reference point as the axial coordinates of the flow path.
3	zvane	real	free	Specifies the axial coordinate (in inches) of the vane trailing edge tip. This coordinate must be measured from the same reference point as the axial coordinates of the flow path.
4	zhigh	real	free	Specifies the axial coordinate (in inches) of the hilite. This coordinate must be measured from the same reference point as the axial coordinates of the flow path.
5	znose	real	free	Specifies the axial coordinate (in inches) of the nose cone leading edge. This coordinate must be measured from the same reference point as the axial coordinates of the flow path.
6	zcore	real	free	Specifies the axial coordinate (in inches) of the splitter leading edge. This coordinate must be measured from the same reference point as the axial coordinates of the flow path.

TABLE 6.—FORMAT OF SETUP_BFANS INPUT FILE

Block	Variable name	Type	Format	Description
7	zexit	real	free	Specifies the axial coordinate (in inches) of the duct exit. This coordinate must be measured from the same reference point as the axial coordinates of the flow path.
8	filenam	character	a30	Specifies the name of the file that contains the blade geometry. The length of the filename must be less than or equal to 30 characters. This file must be located in the directory specified by Card 5002 of the Setup_BFaNS author file (note: the default geometry directory is **bfans_geometry).**
9	bnum	integer	free	Specifies the number of fan blades.
10	dir	integer	free	Specifies the fan's direction of rotation. Enter +1 if the fan is rotating counter-clockwise viewed from the rear, or −1 if the fan is rotating clockwise viewed from the rear. This notation is consistent with the cylindrical coordinate system shown earlier, where θ increases in the counter-clockwise direction viewed from the rear of the engine.
11	filenam	character	a30	Specifies the name of the file that contains the vane geometry. The length of the filename must be less than or equal to 30 characters. This file must be located in the directory specified by Card 5002 of the Setup_BFaNS author file (note: the default geometry directory is **bfans_geometry).**
12	vnum	integer	free	Specifies the number of vanes.
13	filenam	character	a30	Specifies the name of the file that contains the flow-field description at Station #1 upstream from the fan. The length of the filename must be less than or equal to 30 characters.
14	filetyp1	integer	free	Specifies whether the file specified in Block 13 contains information from an inviscid streamline deck, a viscous cfd deck, or measured data. Enter 1 for streamline, 2 for cfd, 3 for data. If 1, the file specified by Block 13 must be located in the directory specified by Card 5004 of the Setup_BFaNS author file. If 2, the file specified by Block 13 must be located in the directory specified by Card 5006 of the Setup_BFaNS author file. If 3, the file specified by Block 13 must be located in the directory specified by Card 5008 of the Setup_BFaNS author file.
15	filenam	character	a30	Specifies the name of the file that contains the flow-field description at Station #2 downstream from the fan. The length of the filename must be less than or equal to 30 characters.
16	filetyp2	integer	free	Specifies whether the file specified in Block 15 contains information from an inviscid streamline deck, a viscous cfd deck, or measured data. Enter 1 for streamline, 2 for cfd, 3 for data. If 1, the file specified by Block 15 must be located in the directory specified by Card 5004 of the Setup_BFaNS author file. If 2, the file specified by Block 15 must be located in the directory specified by Card 5006 of the Setup_BFaNS author file. If 3, the file specified by Block 15 must be located in the directory specified by Card 5008 of the Setup_BFaNS author file.
17	filenam	character	a30	Specifies the name of the file that contains the flow-field description at Station #3 upstream from the vane. The length of the filename must be less than or equal to 30 characters.

TABLE 6.—FORMAT OF SETUP_BFANS INPUT FILE

Block	Variable name	Type	Format	Description
18	filetyp3	integer	free	Specifies whether the file specified in Block 17 contains information from an inviscid streamline deck, a viscous cfd deck, or measured data. Enter 1 for streamline, 2 for cfd, 3 for data. If 1, the file specified by Block 17 must be located in the directory specified by Card 5004 of the Setup_BFaNS author file. If 2, the file specified by Block 17 must be located in the directory specified by Card 5006 of the Setup_BFaNS author file. If 3, the file specified by Block 17 must be located in the directory specified by Card 5008 of the Setup_BFaNS author file.
19	filenam	character	a30	Specifies the name of the file that contains the flow-field description at Station #4 downstream from the vane. The length of the filename must be less than or equal to 30 characters.
20	filetyp4	integer	free	Specifies whether the file specified in Block 19 contains information from an inviscid streamline deck, a viscous cfd deck, or measured data. Enter 1 for streamline, 2 for cfd, 3 for data. If 1, the file specified by Block 19 must be located in the directory specified by Card 5004 of the Setup_BFaNS author file. If 2, the file specified by Block 19 must be located in the directory specified by Card 5006 of the Setup_BFaNS author file. If 3, the file specified by Block 19 must be located in the directory specified by Card 5008 of the Setup_BFaNS author file.
21	del	real	free	Specifies the boundary layer thickness (in inches) on the shroud surface at Station #1. If zero is entered, the boundary layer thickness will be calculated automatically
22	del	real	free	Specifies the boundary layer thickness (in inches) on the hub surface at Station #1. If zero is entered, the boundary layer thickness will be calculated automatically
23	del	real	free	Specifies the boundary layer thickness (in inches) on the shroud surface at Station #3. If zero is entered, the boundary layer thickness will be calculated automatically
24	del	real	free	Specifies the boundary layer thickness (in inches) on the upper splitter surface at Station #3. If zero is entered, the boundary layer thickness will be calculated automatically
25	title1	character	a30	Specifies the first line of the run title (must be less than or equal to 30 characters)
26	title2	character	a30	Specifies the second line of the run title (must be less than or equal to 30 characters)

3.2.6 Create Setup_BFaNS Author File (Optional)

The author file allows the user to change default values that are hardcoded in Setup_BFaNS. Setup_BFaNS will search the working directory for a file named “setupbfans.author”. If this file exists, Setup_BFaNS will use it to over-ride the internal defaults for various options. Appendix J shows a sample Setup_BFaNS author file. The MATLAB program “sbf_author.m” creates an author-file template. This program is located in the Setup_BFaNS source code directory.

The author file has distinct blocks of input referred to as cards, with each card beginning on a new line. A negative card number indicates that the card is turned off. A positive card number indicates that a card is turned on. This feature allows the user to activate/deactivate various options in Setup_BFaNS. Each card has five input parameters (two integer, two real and one alphanumeric), which are provided at the end of the line. When a card is turned on, its input parameters are accessible to every subroutine within Setup_BFaNS.

TABLE 7.—FORMAT OF SETUP_BFANS AUTHOR FILE

Card	Description
5000	If turned on, this card specifies the file name of the Setup_BFaNS input file, and the user will not be prompted to enter the file name.
5002	If turned on, this card specifies the name of the sub-directory that contains the geometry files (flow path, blade, and vane). The default sub-directory is **bfans_geometry**.
5004	If turned on, this card specifies the name of the sub-directory that contains the flow-field files generated with an inviscid streamline deck. The default sub-directory is **bfans_sline**.
5006	If turned on, this card specifies the name of the sub-directory that contains the flow-field files generated with a viscous cfd deck. The default sub-directory is **bfans_cfd**.
5008	If turned on, this card specifies the name of the sub-directory that contains the flow-field files generated with measured data. The default sub-directory is **bfans_data**.
5010	If turned on, this card specifies the path to the Setup_BFaNS source code. The default path is **~/Codes/fans_nasa/setup_bfans**.
5012 through 5048	Reserved for future use
5050	If turned on, a debug file (debug.dat) will be created in the working directory.
5052	If turned on, the turbulent length scale at Station #1 will be determined in the following manner: 1. if imoda(5052) = 1, the turbulent length scale will be computed from the k-epsilon turbulence model (see cards 5056, 5058, and 5060). 2. if imoda(5052) = 2, the turbulent length scale will be determined from empirical correlations (see cards 5062 and 5064) that relate scale to boundary layer thickness. The default is to use the turbulent length scales specified in the flow-field input file.
5054	If turned on, the turbulent length scale at Station #3 will be determined in the following manner: 1. if imoda(5054) = 1, the turbulent length scale will be computed from the k-epsilon turbulence model (see cards 5056, 5058, and 5060). 2. if imoda(5054) = 2, the turbulent length scale will be determined from empirical correlation (see cards 5066, 5068, and 5070) that relate scale to boundary layer thickness and wake thickness. The default is to use the turbulent length scales specified in the flow-field input file.
5056	If turned on, this card specifies the coefficient that relates the turbulence length scale to the k-epsilon model. The coefficient is specified by the third parameter (real). The default coefficient is one. This card is used only when Card 5052 is on, and imod(5052) = 1.
5058	If turned on, this card will multiply the turbulent kinetic energy (K) by the ratio specified by the third parameter (real). This card only applies to viscous CFD input. This card is used only when Card 5052 is on, and imod(5052) = 1.
5060	If turned on, this card will multiply the turbulent dissipation (epsilon) by the ratio specified by the third parameter (real). This card only applies to viscous CFD input. This card is used only when Card 5052 is on, and imod(5052) = 1.
5062	If turned on, this card specifies the ratio between the turbulent length scale and the hub boundary layer thickness at Station #1. The default ratio is 0.62. This card is used only when Card 5052 is on, and imod(5052) = 2.
5064	If turned on, this card specifies the ratio between the turbulent length scale and the shroud boundary layer thickness at Station #1. The default ratio is 0.62. This card is used only when Card 5052 is on, and imod(5052) = 2.
5066	If turned on, this card specifies the ratio between the turbulent length scale and the upper splitter boundary layer thickness at Station #3. The default ratio is 0.62. This card is used only when Card 5054 is off, and imod(5054) = 2.
5068	If turned on, this card specifies the ratio between the turbulent length scale and the shroud boundary layer thickness at Station #3. The default ratio is 0.62. This card is used only when Card 5054 is off, and imod(5054) = 2.
5070	If turned on, this card specifies the ratio between the turbulent length scale and the wake thickness at Station #3. The default ratio is 0.68. This card is used only when Card 5054 is off, and imod(5054) = 2.
5072 through 5078	Always turn these off. These cards can be used to curve fit the wake centerline.
5080	If turned on, streamtube contraction effects will be included in the calculation of cascade parameters

TABLE 7.—FORMAT OF SETUP_BFANS AUTHOR FILE

Card	Description
5082	If turned on, upwash will be computed from the turbulent kinetic energy assuming isotropic turbulence
5084	If turned on, Setup_BFaNS will not check to see if cfd or experimental covers one full pitch
9999	Indicates end of author file.

3.2.7 Execute Setup_BFaNS

Table 8 shows the procedure for executing Setup_BFaNS.

TABLE 8.—EXECUTING SETUP_BFANS

Step 1	Follow the instruction in Sections 3.2.1 through 3.2.6.
Step 2	Go to the working directory.
Step 3	Execute Setup_BFaNS using the alias sbf.
Step 4	When prompted, enter the name of the Setup_BFaNS author file.
Step 5	When prompted, enter the name of the Setup_BFaNS input file (unless Card 5000 in the Setup_BFaNS author file is turned on, in which case the file name is specified in the author file).
Step 6	A figure will be displayed on the screen showing the blade velocity triangles. Look at the figure and confirm that the velocity triangles are correct relative to the blade orientation. If satisfied, place your cursor inside the figure window, and type the letter x on your keyboard. Typing x will continue program execution.
Step 7	A figure will be displayed on the screen showing the vane velocity triangles. Look at the figure and confirm that the velocity triangles are correct relative to the vane orientation. If satisfied, place your cursor inside the figure window, and type the letter x on your keyboard. Typing x will continue program execution.
Step 8	A figure will be displayed on the screen showing the fan-duct configuration. Look at the figure and confirm that the configuration is correct. If satisfied, place your cursor inside the figure window, and type the letter x on your keyboard. Typing x will continue program execution.
Step 9	A menu will be displayed to allow you to change the geometry. If you wish to change something, type in the number corresponding to your selection and press return. Then provide the requested input. The old and new geometry will now be shown on the screen. Place your cursor inside the figure window, and type the letter x on your keyboard. Typing x will continue program execution, allowing you to make more changes to the geometry.
Step 10	After all geometric changes have been completed, type 99 to finish running Setup_BFANS.
Step 11	Confirm that you have a file named bfans.input in the working directory, along with a sub-directory .

3.2.8 Examine Boundary-Layer Characteristics

For every new case, the author recommends that the user enter zeroes in blocks 21, 22, 23 and 24 of the input file. By entering zeroes, the endwall boundary-layer thicknesses will be calculated automatically. However, the automated calculation procedure is not very robust, so the user should confirm that the computed boundary layers are reasonable. If not, the boundary layers must be entered manually, and Setup_BFaNS must be run again.

Table 9 shows the procedure for checking the endwall boundary layers. This procedure assumes that the user has access to MATLAB.

TABLE 9.—EXAMINING ENDWALL BOUNDARY LAYERS

Step 1	Go to the MATLAB sub-directory inside the working sub-directory. This sub-directory should contain several m-files, including "plot_bl.m", "station1.m", and "station3.m".
Step 2	Launch MATLAB
Step 3	At the MATLAB prompt, type plot_bl.m
Step 4	The boundary-layer profiles at Station #1 and Station #3 will be displayed. If the boundary layers look incorrect, visually estimate the correct boundary layer thickness.
Step 5	Enter the correct boundary-layer thicknesses in the Setup_BFaNS input file (Blocks 21 - 24), and re-run Setup_BFaNS.

3.2.9 Examine Wake Characteristics

Table 10 shows the procedure for checking the fan wake characteristics. This procedure assumes that the user has access to MATLAB. This step is important only when the turbulent length scales at Station no. 3 are being computed from the fan wake thickness (as determined by Card 5054 in the author file).

TABLE 10.—EXAMINING WAKE CHARACTERISTICS

Step 1	Go to the MATLAB sub-directory inside the working directory. This sub-directory should contain several m-files, including "plot_wake.m".
Step 2	Launch MATLAB
Step 3	At the MATLAB prompt, type plot_wake.m.
Step 4	Examine the wake profiles at each radius, focusing on the region away from the endwalls.
Step 5	If the wakes look incorrect, contact the author. Currently, there is no way to manually enter the wake thicknesses.

3.2.10 Examine Output File

Table 11 provides suggestions for examining the Setup_BFaNS output file. These suggestions are based on some of the problems that the author encountered when first running Setup_BFaNS. These problems usually indicated an error in one of the flow-field input files (e.g., in the way that the flow-field predictions were performed).

TABLE 11.—EXAMINING THE OUTPUT FILE

Step 1	Confirm that there are no overflows (i.e *****) in the output file
Step 2	Confirm that there are no reverse flow regions (Cz < 0)
Step 3	Confirm that the swirl is zero at the rotor leading edge (unless an inlet guide vane is present)
Step 4	Confirm that the swirl between the rotor and stator has the same sign as the rotational speed of the rotor
Step 5	Confirm that there is little or no swirl at the vane trailing edge
Step 6	Confirm that the swirl at the hub of the fan leading edge is equal to the hub rotation speed
Step 7	Confirm that the turbulent scales in the endwall regions are less than the boundary-layer thickness; confirm that the turbulent length scales in the wake region are less than the wake thickness

4.0 Program Documentation

This section provides technical documentation for the Setup_BFaNS computer program, which is written in FORTRAN 77. Appendix L shows the subroutine hierarchy.

4.1 Units

Program inputs and outputs are in English units. However, internal calculations are done in SI units.

4.2 Geometry Input

Geometric information is passed into the main program through subroutine “get_geom”. This routine first reads the flow-path geometry, then the blade geometry, and finally the vane geometry.

4.2.1 Flow-Path Geometry

The flow-path geometry is read by the subroutine “get_duct_geom”. This routine reads the (*z*,*r*) coordinates that define the hub surface, the shroud surface, the lower splitter surface and the upper splitter surface. This routine also reads the axial location of the blade tip trailing edge (ZBLADE), the vane tip trailing edge (ZVANE), the hilite (ZHIGH), the nose-cone leading edge (ZNOSE), the splitter leading edge (ZCORE), and the fan-duct trailing edge (ZEXIT). See Figure 5 for a description of the flow-path terminology.

4.2.2 Blade and Vane Geometry

The blade and vane geometries are read by the routine “get_row_geom”, which is called separately for the blade and the vane. This routine reads the *(x*,y*,z*)* coordinates and metal angle of the leading and trailing edges of the airfoil. See Figure 6 for a description of the airfoil coordinate system.

After reading the airfoil geometry, “get_row_geom” applies the following transformations:

- Rotates the airfoil to ensure that the innermost point on the leading edge is located on the +*x* axis,

$$\theta^*_{ID} = \tan^{-1}\left(\frac{y^*_{ID}}{x^*_{ID}}\right)$$

$$x = x^* \cos\theta^*_{ID} + y^* \sin\theta^*_{ID}$$

$$y = -x^* \sin\theta^*_{ID} + y^* \cos\theta^*_{ID}$$

Converts from Cartesian coordinates to cylindrical coordinates, i.e.,

$$r = \sqrt{x^2 + y^2}$$

$$\tan\theta = \frac{y}{x}$$

- Ensures that the airfoil is oriented properly relative to the direction of rotation (this process is referred to as “flipping” the airfoil). “Flipping” is required because Setup_BFaNS internally uses metal angles that are measured from the +θ direction, whereas the airfoils are defined relative to their suction-side tangential.
- If the fan is rotating counter-clockwise (as determined by Block 10 in the Setup_BFaNS input file), Setup_BFaNS will take the supplement of the blade metal angles. This process is needed because the suction-side tangential of the blade is in the –θ direction for counter-clockwise rotation. In addition, Setup_BFaNS will multiply the θ-coordinates of the blade by –1. No changes are made to the vane geometry because its suction-side tangential is already in the +θ direction.
- If the fan is rotating clockwise, Setup_BFaNS will take the supplement of the vane metal angles. This process is needed because the suction-side tangential of the vane is in the –θ direction for clockwise rotation. In addition, Setup_BFaNS will multiply the θ-coordinates of the vane by –1. No changes are made to the blade geometry because its suction-side tangential is already in the +θ direction.
- Translates the blade and vane to ensure that their tip trailing edges are located at ZBLADE and ZVANE, respectively,

$$z = z^* + \text{ZBLADE} \quad \text{or} \quad z = z^* + \text{ZVANE}$$

4.3 Flow-Field Input

Flow-field information is passed into the main program through subroutine “get_flow_data”. This routine reads the velocity, total-pressure and total-temperature fields at four stations (see Figure 5):

- Station 1: upstream from the blade leading edge
- Station 2: downstream from the blade trailing edge (but upstream from the splitter leading edge)
- Station 3: upstream from the vane leading edge (but downstream from the splitter leading edge)
- Station 4: downstream from the vane trailing edge.

It also circumferentially averages the flow field, computes the density and speed-of-sound profiles, and computes the streamfunction. Finally, “get_flow_data” ensures that the fan speed is in the direction specified in the Setup_BFaNS input file.

4.3.1 Velocity Field

The coordinates of the velocity grid are read by the subroutine “get_mesh_data”. This routine reads the *(x*,y*,z*)* coordinates at which velocity data is available (the velocity grid must define an axisymmetric surface). When using experimental data or viscous CFD, the *z**-coordinates must be measured from the blade tip trailing edge. When using results from an inviscid streamline deck, the *z**-coordinates must be measured from the same reference that was used to define the flow-path geometry.

After reading the data points, “get_mesh_data” applies the following transformations:

- Rotates the grid to ensure that the first point is located on the +*x* axis. This transformation is permissible because all flow-field quantities will ultimately be averaged in the circumferential direction (i.e., the location of the wake relative to the blade or vane is not important for broadband noise).

$$\theta^*_{ID} = \tan^{-1}\left(\frac{y^*_{ID}}{x^*_{ID}}\right)$$
$$x = x^* \cos\theta^*_{ID} + y^* \sin\theta^*_{ID}$$
$$y = -x^* \sin\theta^*_{ID} + y^* \cos\theta^*_{ID}$$

- Converts from Cartesian coordinates to cylindrical coordinates, i.e.,

$$r = \sqrt{x^2 + y^2}$$
$$\tan\theta = \frac{y}{x}$$

- If necessary, translates the grid to ensure that the tip is located at the correct axial location, i.e.,

$$z = z^* + ZBLADE$$

This translation is not done when the flow-field is provided by an inviscid streamline prediction.

Once the grid coordinates are known, the mean and turbulent velocity fields are read by the subroutine “get_velocity_data”.

When the flow-field is based on experimental data or viscous CFD, and some components of the Reynolds-stress tensor are not available, “get_velocity_data” estimates them based on the turbulence information that is available (see Table 12). The available turbulence information must satisfy one of the following conditions:

Condition	Known turbulence quantities
1	Turbulent kinetic energy
2	One normal-stress component
3	Two normal-stress components
4	Two normal-stress components, one shear-stress component
5	Three normal-stress components
6	Three normal-stress components, one shear-stress component
7	Three normal-stress components, two shear-stress components
8	Three normal-stress components, three shear-stress components

These conditions are consistent with the turbulence data that might be available from hot-wire or laser Doppler velocimetry (LDV) experiments.

If necessary, “get_velocity_data” extends the grid to the walls, where it sets the velocity to zero (relative to the wall), along with all turbulence quantities. For viscous-CFD results, “get_velocity_data” also checks for negative values of *K* (which are physically impossible) and sets them and the corresponding Reynolds-stress components equal to zero. This step eliminates numerical problems later in the computer program.

For viscous-CFD results, the user has the option to multiply *K* and all components of the Reynolds-stress tensor by the constant specified in Card 5058 of the author file. The user also has the option to multiply the viscous dissipation (ε) by the constant specified in Card 5060 of the author file. These options can be used to “adjust” the turbulence predictions to “improve” agreement with measured data.

4.3.2 Circumferential Averaging

Subroutine “circum_average” circumferentially averages the velocity field in the following manner,

$$Q_{avg} = \frac{1}{\theta_{\max} - \theta_{\min}} \int_{\theta_{\min}}^{\theta_{\max}} Q d\theta$$

In the equation above, *Q* can represent any velocity component (C_i), any Reynolds-stress component ($\overline{c_i c_j}$), turbulent kinetic energy (*K*), or viscous dissipation (ε). The integration uses the trapezoidal rule.

TABLE 12.—ESTIMATING THE REYNOLDS-STRESS TENSOR

Condition	K	$\overline{c_z c_z}$	$\overline{c_\theta c_\theta}$	$\overline{c_r c_r}$	$\overline{c_z c_\theta}$	$\overline{c_z c_r}$	$\overline{c_\theta c_r}$
1	known	$=\frac{2}{3}K$	$=\frac{2}{3}K$	$=\frac{2}{3}K$	0	0	0
2	$=\frac{3}{2}\overline{c_z c_z}$	known	$=\overline{c_z c_z}$	$=\overline{c_z c_z}$	0	0	0
	$=\frac{3}{2}\overline{c_\theta c_\theta}$	$=\overline{c_\theta c_\theta}$	known	$=\overline{c_\theta c_\theta}$	0	0	0
	$=\frac{3}{2}\overline{c_r c_r}$	$=\overline{c_r c_r}$	$=\overline{c_r c_r}$	known	0	0	0
3	$=\frac{3}{4}\left(\overline{c_z c_z}+\overline{c_\theta c_\theta}\right)$	known	known	$=\frac{1}{2}\left(\overline{c_z c_z}+\overline{c_\theta c_\theta}\right)$	0	0	0
	$=\frac{3}{4}\left(\overline{c_z c_z}+\overline{c_r c_r}\right)$	known	$=\frac{1}{2}\left(\overline{c_z c_z}+\overline{c_r c_r}\right)$	known	0	0	0
	$=\frac{3}{4}\left(\overline{c_\theta c_\theta}+\overline{c_r c_r}\right)$	$=\frac{1}{2}\left(\overline{c_\theta c_\theta}+\overline{c_r c_r}\right)$	known	known	0	0	0
4	$=\frac{3}{4}\left(\overline{c_z c_z}+\overline{c_\theta c_\theta}\right)$	known	known	$=\frac{1}{2}\left(\overline{c_z c_z}+\overline{c_\theta c_\theta}\right)$	known	$=\overline{c_z c_\theta}$	$=\overline{c_z c_\theta}$
	$=\frac{3}{4}\left(\overline{c_z c_z}+\overline{c_r c_r}\right)$	known	$=\frac{1}{2}\left(\overline{c_z c_z}+\overline{c_r c_r}\right)$	known	$=\overline{c_z c_r}$	known	$=\overline{c_z c_r}$
	$=\frac{3}{4}\left(\overline{c_\theta c_\theta}+\overline{c_r c_r}\right)$	$=\frac{1}{2}\left(\overline{c_\theta c_\theta}+\overline{c_r c_r}\right)$	known	known	$=\overline{c_\theta c_r}$	$=\overline{c_\theta c_r}$	known
5	$=\frac{1}{2}\left(\overline{c_z c_z}+\overline{c_\theta c_\theta}+\overline{c_r c_r}\right)$	known	known	known	0	0	0
6	$=\frac{1}{2}\left(\overline{c_z c_z}+\overline{c_\theta c_\theta}+\overline{c_r c_r}\right)$	known	known	known	known	$=\overline{c_z c_\theta}$	$=\overline{c_z c_\theta}$
	$=\frac{1}{2}\left(\overline{c_z c_z}+\overline{c_\theta c_\theta}+\overline{c_r c_r}\right)$	known	known	known	$=\overline{c_z c_r}$	known	$=\overline{c_z c_r}$
	$=\frac{1}{2}\left(\overline{c_z c_z}+\overline{c_\theta c_\theta}+\overline{c_r c_r}\right)$	known	known	known	$=\overline{c_\theta c_r}$	$=\overline{c_\theta c_r}$	known
7	$=\frac{1}{2}\left(\overline{c_z c_z}+\overline{c_\theta c_\theta}+\overline{c_r c_r}\right)$	known	known	known	known	known	$=\frac{1}{2}\left(\overline{c_z c_\theta}+\overline{c_z c_r}\right)$
	$=\frac{1}{2}\left(\overline{c_z c_z}+\overline{c_\theta c_\theta}+\overline{c_r c_r}\right)$	known	known	known	known	$=\frac{1}{2}\left(\overline{c_z c_\theta}+\overline{c_\theta c_r}\right)$	known
	$=\frac{1}{2}\left(\overline{c_z c_z}+\overline{c_\theta c_\theta}+\overline{c_r c_r}\right)$	known	known	known	$=\frac{1}{2}\left(\overline{c_z c_r}+\overline{c_\theta c_r}\right)$	known	known
8	$=\frac{1}{2}\left(\overline{c_z c_z}+\overline{c_\theta c_\theta}+\overline{c_r c_r}\right)$	known	known	known	known	known	known

4.3.3 Computing Density and Speed of Sound

Density and speed of sound are computed in subroutine "get_rho_sound". "Get_rho_sound" reads the total-pressure and total-temperature profiles at Stations 1 to 4, and computes density and speed of sound via:

$$T = T_T - \frac{1}{2}\frac{C_{avg}^2}{C_p}$$

$$P = P_T\left(\frac{T_T}{T}\right)^{\frac{-\gamma}{\gamma-1}}$$

$$\rho = \frac{P}{RT}$$

$$a = \sqrt{\gamma RT}$$

The specific heat (C_p), gas constant (R) and specific-heat ratio (γ)are hardcoded in "get_rho_sound" to be 1005 m^2/s^2-K, 287 m^2/s^2-K and 1.4, respectively (i.e., the values for air—see Ref. 8).

In some instances, the total-pressure, total-temperature and velocity data are acquired at different axial and radial locations. To account for these instances, "get_rho_sound" interpolates the total-pressure and total-temperature data onto radii that correspond to the same area fractions that the velocity data were acquired, i.e.,

$$\left(\frac{r_{interp}^2 - r_{ID}^2}{r_{OD}^2 - r_{ID}^2}\right)_{P_T, T_T\ Data} = \left(\frac{r_{measured}^2 - r_{ID}^2}{r_{OD}^2 - r_{ID}^2}\right)_{\text{Velocity Data}}$$

The values of r_{interp} are computed, and the (P_T, T_T) data are interpolated onto those radii. Figure 8 demonstrates this process, as it would apply to Station no. 3. The total pressure and total temperature are assumed to be constant along lines of constant area fraction (i.e., the lines connecting the x's and o's). The assumption is that lines of constant area fraction approximate the streamtubes.

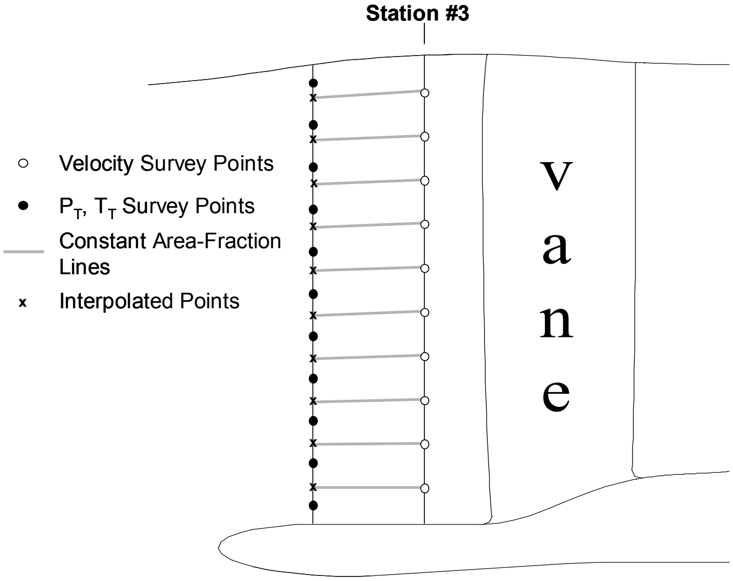


Figure 8.—P_T, T_T interpolation procedure.

4.3.4 Computing Streamfunction

Streamfunction is computed in subroutine “stream_function” by integrating the mass flux profile from the outer wall to the inner wall via:

$$S(r) = \int_{r_{od}}^{r} \rho \vec{C}_{avg} \cdot d\vec{A}$$

Provided there is no reverse flow, there will be a one-to-one correspondence between radius and streamfunction.

Note that the value of the streamfunction at the ID corresponds to the mass flow rate through the measurement surface.

4.3.5 Correcting Direction of Rotation

“Get_flow_data” determines whether the fan speed specified in the flow-field input file is consistent with the direction specified in the Setup_BFaNS input file. If the fan speed is in the wrong direction, then “get_flow_data” makes the following corrections:

$$\Omega = -\Omega$$
$$C_\theta = -C_\theta$$
$$\overline{c_z c_\theta} = -\overline{c_z c_\theta}$$
$$\overline{c_\theta c_r} = -\overline{c_\theta c_r}$$

These corrections are also made to the corresponding circumferentially averaged quantities.

4.4 Checking Consistency of Fan Speeds

After reading the flow-field data at each station, Setup_BFaNS checks the consistency between the fan speeds that were read. If the fan speed at any station is different by 1 to 2 percent of the average fan speed, the program issues a warning and continues. If the fan speed at any station is different by more than 2 percent of the average fan speed, the program is terminated.

4.5 Modifications to Flow Path

To facilitate interpolation of the flow field, the subroutine “add_points” makes the following changes to the flow path:

- ***Shroud Surface***.—Adds the outside-diameter coordinates of the blade leading edge and trailing edge; adds the outside-diameter coordinates of the vane leading edge and trailing edge; adds the outside-diameter coordinates of the velocity grid at Stations 1 to 4
- ***Hub Surface***.—Adds the inside-diameter coordinates of the blade leading edge and trailing edge; adds the inside-diameter coordinates of the velocity grid at Stations 1 and 2
- ***Upper Splitter Surface***.—Adds the inside-diameter coordinates of the vane leading edge and trailing edge; adds the inside-diameter coordinates of the velocity grid at Stations 3 and 4

4.6 Streamline Estimation

The blade geometry, vane geometry and flow field are mapped onto lines of constant streamfunction (i.e., streamtubes). This mapping is done in four steps:

- Subroutine “normalize” normalizes the streamfunction by the mass flow rate passing through the vane. In this manner, the normalized streamfunction varies between 0 and 1 at Stations 3 and 4, and between 0 and 1+BPR at Stations 1 and 2 (where BPR = bypass ratio).
  If the flow rate through Stations 1 and 2 are different by more than 1 percent, this routine warns the user. Likewise, this routine warns the user if the flow rate through Stations 3 and 4 are different by more than 1 percent. It also warns the user if there is more flow passing through the stator than the rotor (i.e., when BPR < 0). These warnings indicate an error in the input files, though the program will continue to run.
- Subroutine “stream_line” maps Station 2 flow-field information onto the streamtubes corresponding to Station 1. The mapping is done by finding the radii at Station 2 that correspond to the normalized streamfunction values at Station 1. A similar mapping is done between Stations 4 and 3. This routine will fail if a one-to-one correspondence between radius and streamfunction does not exist (e.g., when there is reverse flow).
- Subroutine “intersect2” estimates where the streamtubes intersect the blade and vane. It does this by determining the radii at the edge that correspond to the area fractions at the measurement station:

$$\left(\frac{r_{interp}^2 - r_{ID}^2}{r_{OD}^2 - r_{ID}^2}\right)_{Edge} = \left(\frac{r_{measured}^2 - r_{ID}^2}{r_{OD}^2 - r_{ID}^2}\right)_{\substack{Measurement\\Station}}$$

  The assumption is that lines of constant area fraction approximate the streamtubes. The axial coordinate, tangential coordinate and metal angle of the airfoil are interpolated onto these interpolated radii.
- Subroutine “estimate2” estimates the flow-field information at the leading edge and trailing edge of the fan and FEGV. The only quantities affected are the axial, radial and tangential velocities. The axial and radial velocities are adjusted based on the change in area between the measurement station and the airfoil edge. The tangential velocity is adjusted based on the change in radius,

$$C_{z,edge} = C_{z,station}\left(\frac{A_{station}}{A_{edge}}\right)$$

$$C_{r,edge} = C_{r,station}\left(\frac{A_{station}}{A_{edge}}\right)$$

$$C_{\theta,edge} = C_{\theta,station}\left(\frac{r_{station}}{r_{edge}}\right)$$

4.7 Confirm Blade and Vane Orientation

Subroutine “triangle” displays the airfoil geometries and the corresponding inlet and exit velocity triangles on the screen. The user must examine the figures to confirm that the airfoils are oriented properly relative to the velocity triangles. If acceptable, the user must place the cursor in the figure window and hit “x” to continue.

4.8 Airfoil Geometry Calculations

Chord, gap, solidity, stagger angle, geometric sweep and geometric lean are computed in subroutine “geom”. The chord, solidity and stagger angle are computed on the “unwrapped” cascade along the $(m, r\theta)$ surfaces.

4.8.1 Airfoil Chord

Chord is defined by:

$$b = \sqrt{(x_{te} - x_{le})^2 + (y_{te} - y_{le})^2 + (z_{te} - z_{le})^2}$$

Meridional chord is defined by:

$$b_m = \sqrt{(r_{te} - r_{le})^2 + (z_{te} - z_{le})^2}$$

4.8.2 Airfoil Gap

Gap is defined by:

$$\tau = \frac{2\pi r_{le}}{N} \quad \text{where} \quad N = \text{number of airfoils}$$

4.8.3 Solidity

Cascade solidity is defined by:

$$\sigma = \frac{b}{\tau}$$

Gap-to-chord ratio is equal to $\frac{1}{\sigma}$.

4.8.4 Stagger Angle

Stagger angle is defined by:

$$\xi = \tan^{-1}\left(\frac{b_m}{r_{te}\theta_{te} - r_{le}\theta_{le}}\right)$$

In Setup_BFaNS, the stagger angle is measured from the θ-direction. This definition is not consistent with Glegg or Hanson, who measure stagger angle from the axial direction. Therefore, when applying Hanson or Glegg's theories, we must replace their stagger angle with the complement of ξ.

4.8.5 Geometric Sweep

To compute geometric sweep at the *i*'th point on the leading edge, Setup_BFaNS determines the parabola (*z* as a function of *r*) that satisfies the following conditions:

$$\begin{aligned} z &= \zeta_2 r^2 + \zeta_1 r + \zeta_0 \\ z &= z_{le,i-1} \quad \text{at} \quad r = r_{le,i-1} \\ z &= z_{le,i} \quad \text{at} \quad r = r_{le,i} \\ z &= z_{le,i+1} \quad \text{at} \quad r = r_{le,i+1} \end{aligned}$$

The coefficients are determined in subroutine "parabola". From this parabola, the geometric sweep at the *i*'th point is determined via,

$$\tan\psi_S = \frac{dz}{dr}$$

The sweep at the first point is assumed to be equal to that at the second point. Likewise, the sweep at the last point is assumed to be equal to that at the second-to-last point.

4.8.6 Geometric Lean

To compute geometric lean at the *i*'th point on the leading edge, Setup_BFaNS determines the parabola (θ as a function of ln *r*) that satisfies the following conditions:

$$\begin{aligned} \theta &= \zeta_2(\ln r)^2 + \zeta_1(\ln r) + \zeta_0 \\ \theta &= \theta_{le,i-1} \quad \text{at} \quad \ln r = \ln r_{le,i-1} \\ \theta &= \theta_{le,i} \quad \text{at} \quad \ln r = \ln r_{le,i} \\ \theta &= \theta_{le,i+1} \quad \text{at} \quad \ln r = \ln r_{le,i+1} \end{aligned}$$

The coefficients are determined in subroutine “parabola”. From this parabola, the geometric lean at the *i*'th point is determined via,

$$\tan\psi_L = \frac{1}{r}\frac{dr}{d\theta} = \frac{d(\ln r)}{d\theta}$$

The lean at the first point is assumed to be equal to that at the second point. Likewise, the lean at the last point is assumed to be equal to that at the second-to-last point.

4.9 Flow-Field Calculations

Leading-edge Mach number, meridional flow angle and incidence are computed in subroutine “flow”. Cascade parameters such as the lift, drag and loss coefficients are computed in subroutine “cascade”. Equivalent “isolated airfoil” angle-of-attack is computed in subroutine “naca0012”.

4.9.1 Leading-Edge Mach Number

Leading-edge Mach number is defined by:

$$M_{le} = \left(\frac{W_{avg}}{a}\right)_{le} \qquad \text{for the rotor}$$

$$M_{le} = \left(\frac{C_{avg}}{a}\right)_{le} \qquad \text{for the stator}$$

4.9.2 Meridional Flow Angle

The meridional flow angle is defined as the angle between a streamtube and the radial direction. Since the points that define the airfoil have been mapped onto streamtubes, then the meridional flow angle can be determined from:

$$\phi = \tan^{-1}\left(\frac{z_{te} - z_{le}}{r_{te} - r_{le}}\right)$$

4.9.3 Incidence

Incidence is defined as the difference between the leading-edge flow angle and the leading-edge metal angle, both measured from the meridional direction. The leading edge flow angle is,

$$\beta_{le} = \tan^{-1}\left(\frac{W_{\theta,avg}}{W_{m,avg}}\right)_{le}$$

For a blade, the incidence is,

$$i = \beta_{le} - \left(\frac{\pi}{2} - \beta^*_{le}\right)$$

For a vane, the incidence is,

$$i = \left(\frac{\pi}{2} - \beta^*_{le}\right) - \beta_{le}$$

In both cases, the π/2 term is needed because the metal-angle (β*) is referenced to the θ-direction rather than the meridional direction.

4.9.4 Cascade Parameters

The cascade loss coefficient is determined from a correlation based on diffusion factor (Ref. 9 and 10). The diffusion factor for a two-dimensional cascade is defined as,

$$D_f = 1 - \frac{W_{te}}{W_{le}} + \frac{W_{\theta,le} - W_{\theta,te}}{2\sigma W_{le}} \qquad \text{(see Ref. 10, Eq. (54))}$$

When there is a change in radius from the leading edge the trailing edge, the more general form for the diffusion factor is,

$$D_f = 1 - \frac{W_{te}}{W_{le}} + \frac{r_{le}W_{\theta,le} - r_{te}W_{\theta,te}}{\sigma\left(r_{le} + r_{te}\right)W_{le}}$$

Reference 10 presents a plot of wake momentum thickness (normalized by chord) versus diffusion factor for several sets of cascade data. A reasonably good fit to the data is provided by,

$$\frac{\delta_2}{b} = 0.006 + 0.0002\left(e^{7.5D_f} - 1\right) \qquad \text{(see Ref. 10, Fig. 148)}$$

For a two-dimensional cascade with incompressible flow, the cascade loss, drag and lift coefficients can be determined from,

$$\overline{\omega} = \frac{2}{\cos\beta_{te}}\left(\frac{\delta_2}{b}\right)\left(\frac{b}{\tau}\right)\left(\frac{\cos\beta_{le}}{\cos\beta_{te}}\right)^2 \qquad \text{(see Ref. 11, Eq. (3.32))}$$

$$C_D = \overline{\omega}\frac{\tau}{b}\frac{\cos^3\beta_m}{\cos^2\beta_{le}} \qquad \text{(see Ref. 11, Eq. (3.33))}$$

$$C_L = 2\frac{\tau}{b}\left(\tan\beta_{le} - \tan\beta_{te}\right)\cos\beta_m - C_D\tan\beta_m \qquad \text{(see Ref. 11, Eq. (3.18))}$$

where the mean cascade angle is defined by,

$$\beta_m = \tan^{-1}\left(\frac{\tan\beta_{le} + \tan\beta_{te}}{2}\right) \qquad \text{(see Ref. 11, Eq. (3.6))}$$

To avoid a divide-by-zero error in D_f, these calculations are not done at the walls. Rather, the loss, lift and drag coefficients are assumed to be constant over the first two and last two points on the airfoil.

These incompressible cascade relationships assume that the axial velocity at the trailing edge of the cascade is the same as that at the leading edge. However, in a real engine, the axial velocity can change significantly across the cascade due to contraction in the flow path. To assess the effect of streamtube contraction, the author derived alternate expressions for the cascade parameters (see Appendix M). Setup_BFaNS will use these alternate expressions unless Card 5080 in the author file is turned on.

4.9.5 Equivalent Angle of Attack

The equivalent angle-of-attack is defined as the angle-of-attack at which an *isolated airfoil* will have the same lift coefficient as the cascade. The equivalent angle-of-attack for an isolated airfoil is determined from,

$$\alpha = \sin^{-1}\left(\frac{C_L}{2\pi}\right)$$

4.9.6 Upwash

Turbulent upwash is computed in subroutine “upwash”. To compute upwash, we must transform the Reynolds-stress tensor from engine to duct coordinates, and then apply Hanson’s transformation matrix (Ref. 4) between duct coordinates and cascade coordinates. Alternately, we can assume isotropic turbulence and compute the upwash from the turbulent kinetic energy (this simplified approach is the default method in Setup_BFaNS, unless Card 5082 is turned on).

Appendix N shows how Setup_BFaNS transforms the Reynolds-stress tensor from the engine coordinate system (z,r,θ) into the duct coordinate system (i.e., the system that is aligned with the streamtubes). The final equations are:

$$\overline{c_1c_1}' = \overline{c_1c_1}\sin^2\phi + \overline{c_3c_3}\cos^2\phi + 2\overline{c_1c_3}\sin\phi\cos\phi$$

$$\overline{c_2c_2}' = \overline{c_2c_2}$$

$$\overline{c_3c_3}' = \overline{c_1c_1}\cos^2\phi + \overline{c_3c_3}\sin^2\phi - 2\overline{c_1c_3}\sin\phi\cos\phi$$

$$\overline{c_1c_2}' = \overline{c_1c_2}\sin\phi + \overline{c_2c_3}\cos\phi$$

$$\overline{c_1c_3}' = -\overline{c_1c_1}\sin\phi\cos\phi + \overline{c_3c_3}\sin\phi\cos\phi + \overline{c_1c_3}\left(-\cos^2\phi + \sin^2\phi\right)$$

$$\overline{c_2c_3}' = -\overline{c_1c_2}\cos\phi + \overline{c_2c_3}\sin\phi$$

Appendix O shows how Setup_BFaNS computes turbulent upwash when Card 5082 is turned on. The final equation is:

$$\overline{c_2 c_2}'' = \overline{c_m c_n}' \beta_{2m}\beta_{2n}$$

where,

$$\beta_{ij} = \begin{pmatrix} \cos\left(\frac{\pi}{2}-\xi\right)\cos\psi'_S + \sin\left(\frac{\pi}{2}-\xi\right)\sin\psi'_L \sin\psi'_S & \sin\left(\frac{\pi}{2}-\xi\right)\cos\psi'_S - \cos\left(\frac{\pi}{2}-\xi\right)\sin\psi'_L \sin\psi'_S & -\cos\psi'_L \sin\psi'_S \\ -\sin\left(\frac{\pi}{2}-\xi\right)\cos\psi'_L & \cos\left(\frac{\pi}{2}-\xi\right)\cos\psi'_L & -\sin\psi'_L \\ \cos\left(\frac{\pi}{2}-\xi\right)\sin\psi'_S - \sin\left(\frac{\pi}{2}-\xi\right)\sin\psi'_L \cos\psi'_S & \sin\left(\frac{\pi}{2}-\xi\right)\sin\psi'_S + \cos\left(\frac{\pi}{2}-\xi\right)\sin\psi'_L \cos\psi'_S & \cos\psi'_L \cos\psi'_S \end{pmatrix}$$

The aerodynamic sweep and aerodynamic lean used in the transformation matrix are derived in Appendix P. The final equations are:

$$\psi'_S = 90 + \psi_S - \phi$$

$$\tan\psi'_L = \frac{\tan\psi_L}{-\tan\psi_S \cos\phi + \sin\phi}$$

4.10 Viscous Quantities

Setup_BFaNS computes the endwall boundary layers at the blade and vane leading edge, in addition to the blade-wake thickness at the vane leading edge. Endwall boundary layers are computed with subroutines “get_bl” and “calc_bl”. Blade wakes are computed in “get_rotor_wake” and “calc_rotor_wake”.

4.10.1 Endwall Boundary Layers

The user has the option to manually specify the boundary layer thicknesses in the Setup_BFaNS input file, or to estimate them according to the procedure described in Appendix Q.

4.10.2 Rotor Wakes

Appendix R shows how Setup_BFaNS estimates the rotor wakes.

4.11 Turbulent Length Scale

Setup_BFaNS will default to using the integral length scales that are specified in the flow-field input files. However, Cards 5052 or 5054 provide the option to use empirical correlations or the (K,ε) values to determine the turbulent length scales (see Appendix S).

5.0 Concluding Remarks

Pratt & Whitney has developed a Broadband Fan Noise Prediction System (BFaNS) for turbofan engines. The noise prediction system consists of two computer programs: Setup_BFaNS and BFaNS. Setup_BFaNS converts user-specified geometry and flow-field information into a BFaNS input file. From this input file, BFaNS computes the inlet and aft broadband sound power spectra generated by the fan and FEGV. The output file from BFaNS contains the inlet, aft and total sound power spectra from each noise source.

The Broadband Fan Noise Prediction System has been documented in a three-volume report. The present volume provides detailed instructions and technical documentation for Setup_BFaNS. Volume 2 provides detailed instructions and technical documentation for BFaNS. Volume 3 provides validation and test cases for the prediction system.

Appendix A.—Symbols

A.1 Variables

A	area
a	speed of sound
B	blade count
b	airfoil chord
$\vec{C}$ or C_i	absolute velocity vector or tensor
C_D	cascade drag coefficient
C_L	cascade lift coefficient
C_P	constant-pressure specific heat
D_f	diffusion factor
F	longitudinal spectrum
h	cascade gap measured perpendicular to airfoils
i	incidence
K	turbulent kinetic energy
L	length
M	Mach number
P_T	total pressure
R	gas constant
R_L	Reynolds number based on L
R_δ	Reynolds number based on δ
r	radius
S	streamfunction
s	fan-to-FEGV spacing
T_T	total temperature
T_{ij}	generalized 2nd-order tensor
$\vec{U}$ or U_i	blade velocity vector or tensor
U_∞	free-stream velocity
$\overline{c_i c_j}$	*ij*-component of Reynolds-stress tensor
V	vane count
$\vec{W}$ or W_i	relative velocity vector or tensor
x	horizontal coordinate viewed from rear of engine
y	vertical coordinate viewed from rear of engine
z	axial coordinate

A.2 Greek Symbols

α	absolute flow angle; effective angle-of-attack
β^*	airfoil metal angle

β	relative flow angle
β_m	average flow angle in cascade
β_{ij}	transformation matrix for Cartesian coordinates
γ	specific-heat ratio
δ	boundary-layer thickness
δ_2	momentum thickness
δ_W	wake thickness
ε	viscous dissipation
θ	tangential location
Λ	turbulent length scale
ν	kinematic viscosity
ξ	stagger angle from meridional direction (duct coordinate system)
χ	stagger angle from tangential direction (cascade coordinate system)
ρ	density
σ	cascade solidity (b/τ)
τ	cascade pitch
ϕ	meridional flow angle
ϕ_{ij}	energy spectrum tensor
$\widetilde{\Psi}_L$	overall geometric lean angle (engine coordinate system)
$\widetilde{\Psi}_S$	overall geometric sweep angle (engine coordinate system)
$\widetilde{\psi}_L$	local geometric lean angle (duct coordinate system)
$\widetilde{\psi}_S$	local geometric sweep angle (duct coordinate system)
$\hat{\psi}_L$	local aerodynamic lean angle (duct coordinate system)
$\hat{\psi}_S$	local aerodynamic sweep angle (duct coordinate system)
ψ_L	local lean angle (cascade coordinate system)
ψ_S	local sweep angle (cascade coordinate system)
Ω	fan rotational speed
ζ_0, ζ_1, ζ_2	coefficients of parabolic fit
$\overline{\omega}$	cascade loss coefficient

A.3 Subscripts

c	cascade coordinates
d	duct coordinates
e	engine coordinates
x	horizontal component
y	vertical component
z	axial component

r	radial component
θ	tangential component
m	meridional component; mean cascade
le	leading edge
te	trailing edge
ID	inside diameter
OD	outside diameter
avg	circumferential average
nose	nose-cone leading edge
high	hilite leading edge
core	splitter leading edge

A.4 Superscripts

*	rotated or translated coordinates
'	duct coordinate system
"	cascade coordinate system

Appendix B.—Installing the BFaNS System

The BFaNS system can be downloaded from the following web site: http://www.grc.nasa.gov/WWW/5900/5940/code/BFANS/

B.1 Step Number I: Setting Up the Directory Structure

1. In your home directory, create a directory called **Codes**.
2. Copy the file **fans_nasa.tar** to your **~/Codes** directory.
3. In your **~/Codes** directory, extract the files using the following command:
   ```
   tar -xvf fans_nasa.tar
   ```
4. After executing this command, you will have a **~/Codes/fans_nasa** directory. This directory will contain the following sub-directories**:**

   setup_bfans: contains source code and make files for Setup_BFaNS
   bfans: contains source code and make files for BFaNS
   lib: contains source code and make files for GRAFIC (a freely distributed graphics library created by MIT)
   makeflags: contains compiler options for Sun, SGI, and IBM workstations
5. Appendix K lists the files contained in each directory

B.2 Step Number II: Setting Up the Graphics Library (GRAFIC)

1. Go to the **~/Codes/fans_nasa/lib** directory. There you will find libraries compiled for an SGI workstation (**libgrafic.a.Iris**), a Sun workstation (**libgrafic.a.Sun4**), and an IBM workstation (**libgrafic.a.Ibm6000**). You can use one of these existing libraries, or compile the library yourself. If you choose to use an existing library, proceed to step no. 2. If you choose to compile the library yourself, proceed to step no. 3.
2. If you are working on an SGI workstation (with iris architecture), type the command:
   ```
   cp libgrafic.a.Iris libgrafic.a
   ```
   If you are working on a Sun workstation (with svr4 architecture), type the command:
   ```
   cp libgrafic.a.Sun4 libgrafic.a
   ```
   If you are working on an IBM workstation (with rs6 architecture), type the command:
   ```
   cp libgrafic.a.Ibm6000  libgrafic.a
   ```
   For all the above workstations, proceed to Step no. III to compile Setup_BfaNS.
3. Go to the **~/Codes/fans_nasa/lib/grafic** directory.
4. If you are working on an SGI workstation (with iris architecture), type the command:
   ```
   make -f Makefile.Iris
   ```
   If you are working on a Sun workstation (with svr4 architecture), type the command:
   ```
   make -f Makefile.Sun4
   ```
   If you are working on an IBM workstation (with rs6 architecture), type the command:
   ```
   make -f Makefile.Ibm6000
   ```
   The above commands will create a file called **libgrafic.a** in your **~/Codes/fans_nasa/lib** directory. If you have problems compiling, go to the **~/Codes/fans_nasa/makeflags** directory and check the compiler options that are set for your workstation (i.e., **makeflags.Iris**, **makeflags.Sun4**, or **makeflags.Ibm6000**).

B.3 Step Number III: Compiling Setup_BFaNS

1. Go to the **~/Codes/fans_nasa/setup_bfans** directory.
2. If you are working on an SGI workstation (with iris architecture), type the command:
   ```
   make -f Makefile.Iris
   ```

If you are working on a Sun workstation (with svr4 architecture), type the command:

```
make -f Makefile.Sun4
```

If you are working on an IBM workstation (with rs6 architecture), type the command:

```
make -f Makefile.Ibm6000
```

The above commands will create an executable file called **setup_bfans.x** in your **~/Codes/fans_nasa/setup_bfans** directory. If you have problems compiling, go to the **~/Codes/fans_nasa/makeflags** directory and check the compiler options that are set for your workstation (i.e., **makeflags.Iris**, **makeflags.Sun4**, or **makeflags.Ibm6000**).

3. If you want to change the compiler optimization (or other options), edit the appropriate makeflags file and repeat step no. 2 in this section.

B.4 Step Number IV: Compiling BFaNS

1. Go to the **~/Codes/fans_nasa/bfans** directory.
2. If you are working on an SGI workstation (with iris architecture), type the command:

   ```
   make -f Makefile.Iris
   ```

   If you are working on a Sun workstation (with svr4 architecture), type the command:

   ```
   make -f Makefile.Sun4
   ```

   If you are working on an IBM workstation (with rs6 architecture), type the command:

   ```
   make -f Makefile.Ibm6000
   ```

   The above commands will create an executable file called **bfans.x** in your **~/Codes/fans_nasa/bfans** directory. If you have problems compiling, go to the **~/Codes/fans_nasa/makeflags** directory and check the compiler options that are set for your workstation (i.e., **makeflags.Iris**, **makeflags.Sun4**, or **makeflags.Ibm6000**).
3. If you want to change the compiler optimization (or other options), edit the appropriate makeflags file and repeat step no. 2 in this section.

B.5 Step Number V: Setting Up Aliases

1. Inside your **.cshrc** file, create the following aliases:

   ```
   alias sbf ~/Codes/fans_nasa/setup_bfans/setup_bfans.x
   alias bf ~/Codes/fans_nasa/bfans/bfans.x
   ```
2. In your home directory, type `source .cshrc`
3. These aliases are not required, but they make running much easier.

Appendix C.—Running the Test Case

The BFaNS test case can be downloaded from the following web site: http://www.grc.nasa.gov/WWW/5900/5940/code/BFANS/

C.1 Step Number I: Setting Up the Test Case

1. In your home directory, create a directory called **bbchallenge**.
2. Copy the file **bbchallenge.tar** to your **~/bbchallenge directory**.
3. In your **~/bbchallenge** directory, extract the files using the following command:

   tar -xvf bbchallenge.tar

   After executing this command, you will have the following 13 sub-directories inside ~/bbchallenge:

   bfans_geometry: contains files that describe the flow path, blade, and vane geometry
   bfans_cfd: contains files that describe viscous CFD-predicted flow field near blade l.e., blade t.e., vane l.e., and vane t.e. for each configuration and operating point
   bfans_data: not used (normally contains files that describe measured flow field near blade l.e., blade t.e., vane l.e., and vane t.e. for each configuration and operating point)
   bfans_sline: not used (normally contains files that describe inviscid streamline-predicted flow field near blade l.e., blade t.e., vane l.e., and vane t.e. for each configuration and operating point)
   54vr_spd1: contains test case for 54 radial vanes, sideline power
   54vr_spd2: contains test case for 54 radial vanes, cutback power
   54vr_spd3: contains test case for 54 radial vanes, approach power
   26vr_spd1: contains test case for 26 radial vanes, sideline power
   26vr_spd2: contains test case for 26 radial vanes, cutback power
   26vr_spd3: contains test case for 26 radial vanes, approach power
   26vs_spd1: contains test case for 26 radial vanes, sideline power
   26vs_spd2: contains test case for 26 radial vanes, cutback power
   26vs_spd3: contains test case for 26 radial vanes, approach power

C.2 Step Number II: Running a Test Case for Setup_BFaNS

1. Inside your ~/**bbchallenge/54vr_spd3** directory, type the command **sbf** or **~/Codes/fans_nasa/setup_bfans/setup_bfans.x**. This command will run Setup_BFaNS in the current directory.
2. When prompted for the name of the input file, enter **setupbfans.input**. This file contains the file names that describe the geometry and flow field (these files are located in the **bfans_geometry** and **bfans_cfd** sub-directories under **~/bbchallenge**), along with some additional information.
3. A figure will be displayed on the screen showing the blade velocity triangles. Look at the figure and confirm that the velocity triangles are correct relative to the blade orientation. Once satisfied, place your cursor inside this window, and type the letter **x** on your keyboard. Typing **x** will continue program execution.
4. A figure will be displayed on the screen showing the vane velocity triangles. Look at the figure and confirm that the velocity triangles are correct relative to the vane orientation. Once satisfied, place your cursor inside this window, and type the letter **x** on your keyboard. Typing **x** will continue program execution.
5. A figure will be displayed on the screen showing the fan-duct configuration. Look at the figure and confirm that the configuration is correct. Once satisfied, place your cursor inside this window, and type the letter **x** on your keyboard. Typing **x** will continue program execution.
6. A menu will be displayed to allow you to change the geometry. If you wish to change something, type in the number corresponding to your selection and press return. Then provide the requested

input. The old and new geometry will now be shown on the screen. Place your cursor inside the figure window, and type the letter **x** on your keyboard. Typing **x** will continue program execution, allowing you to make more changes to the geometry.

7. Type **99** to finish running Setup_BFANS.
8. You will now have a file named **bfans.input** in the **~/bbchallenge/54vr_spd3** directory. If you didn't make any changes to the geometry, then **bfans.input** should be nearly identical to **bfans.input.Sun4** which was created on a Sun workstation at P&W (there might be some very small differences depending on the compiler). To confirm that the two files are nearly identical, type the following command:

   ```
   diff bfans.input bfans.input.Sun4
   ```

   This command will show all differences between the two files.

C.3 Step Number III: Running a Test Case for BFaNS

1. Inside your **~/bbchallenge/54vr_spd3** directory, type the command **bf** or **~/Codes/fans_nasa/bfans/bfans.x**. This command will run BFaNS inside the current directory (note: to save time during code checkout, you might want to edit the **bfans.author** file to reduce the number of frequencies calculated).
2. When prompted for the name of the input file, enter **bfans.input**. This file contains the output from Setup_BFaNS.
3. When done, BFaNS will display "calculation completed" on the screen. You will now have a file named **bfans.output** in the **~/bbchallenge/54vr_spd3** directory. If you didn't make any changes to the geometry, then **bfans.output** should be nearly identical to **bfans.output.Sun4** which was created on a Sun workstation at P&W (there might be some very small differences depending on the compiler). To confirm that the two files are nearly identical, type the following command:

   ```
   diff bfans.output bfans.output.Sun4
   ```

   This command will show all differences between the two files.

Appendix D.—Sample Flow-Path Geometry File

filedescr	Flowpath: Inside Diameter Wall - HUB				
nduct_h	90				
parmname	Axial				
zhub	-6.502327	-6.232568	-5.962808	-5.693049	-5.423289
	-5.153530	-4.883770	-4.614011	-4.344251	-4.074492
	-3.804733	-3.534973	-3.265214	-2.995454	-2.725695
	-2.455935	-2.186176	-1.916416	-1.646657	-1.376897
	-1.107138	-0.837379	-0.567619	-0.297860	-0.028100
	0.241659	0.511419	0.781178	1.050938	1.320697
	1.590456	1.860216	2.129975	2.399735	2.669494
	2.939254	3.209013	3.478773	3.748532	4.018291
	4.288051	4.557810	4.827570	5.097329	5.367089
	5.636848	5.906608	6.176367	6.446126	6.715886
	6.985645	7.255405	7.525164	7.794924	8.064683
	8.334443	8.604202	8.873961	9.143721	9.413480
	9.683240	9.952999	10.222759	10.492518	10.762278
	11.032037	11.301796	11.571556	11.841315	12.111075
	12.380834	12.650594	12.920353	13.190113	13.459872
	13.729631	13.999391	14.269150	14.538910	14.808669
	15.078429	15.348188	15.617948	15.887707	16.157467
	16.427226	16.696985	16.966745	17.236504	17.506264
parmname	Radius				
rhub	2.250805	2.250757	2.250756	2.250848	2.250945
	2.250903	2.250703	2.251318	2.250003	2.268878
	2.297416	2.373692	2.525183	2.674208	2.791127
	2.900357	3.002891	3.099274	3.189671	3.276292
	3.358367	3.435867	3.509917	3.581078	3.650682
	3.720369	3.794309	3.878063	3.969007	4.067281
	4.170762	4.275789	4.379734	4.479830	4.572974
	4.656779	4.731721	4.799961	4.862569	4.921029
	4.976265	5.029055	5.079441	5.127756	5.173022
	5.214311	5.251548	5.287234	5.322437	5.356940
	5.391048	5.429903	5.478404	5.534245	5.592687
	5.649140	5.699345	5.740850	5.772924	5.795676
	5.810992	5.821017	5.827340	5.831110	5.833467
	5.835088	5.836256	5.836969	5.836978	5.836439
	5.836241	5.837039	5.839078	5.843604	5.850944
	5.859892	5.875535	5.892634	5.916299	5.941113
	5.971357	6.001601	6.037731	6.073982	6.113309
	6.155001	6.196693	6.242894	6.289515	6.336137
blank					
filedesc	Primary Splitter: Inside Diameter Wall - UPPER				
nsplt_u	50				
parmname	Axial				
zsup	6.756294	6.975681	7.195069	7.414456	7.633843
	7.853230	8.072617	8.292004	8.511391	8.730778
	8.950166	9.169553	9.388940	9.608327	9.827714
	10.047101	10.266488	10.485875	10.705263	10.924650
	11.144037	11.363424	11.582811	11.802198	12.021585
	12.240973	12.460360	12.679747	12.899134	13.118521
	13.337908	13.557295	13.776682	13.996070	14.215457
	14.434844	14.654231	14.873618	15.093005	15.312392
	15.531780	15.751167	15.970554	16.189941	16.409328
	16.628715	16.848102	17.067489	17.286877	17.506264
parmname	Radius				
rsup	5.362028	5.389747	5.420434	5.457386	5.500269
	5.546825	5.594395	5.640565	5.683004	5.719951
	5.750893	5.775476	5.793916	5.807358	5.816899
	5.823565	5.828091	5.831034	5.833065	5.834527
	5.835635	5.836469	5.836996	5.836986	5.836640
	5.836273	5.836223	5.837216	5.838722	5.842403
	5.846898	5.854175	5.862536	5.875341	5.888147
	5.907170	5.926415	5.948395	5.972991	5.997588
	6.026151	6.055633	6.085115	6.118328	6.152235
	6.186142	6.222389	6.260305	6.298221	6.336137
blank					
filedesc	Primary Splitter: Outside Diameter Wall - LOWER				

nsplt_l	50				
parmname	Axial				
zslo	6.756294	6.975681	7.195069	7.414456	7.633843
	7.853230	8.072617	8.292004	8.511391	8.730778
	8.950166	9.169553	9.388940	9.608327	9.827714
	10.047101	10.266488	10.485875	10.705263	10.924650
	11.144037	11.363424	11.582811	11.802198	12.021585
	12.240973	12.460360	12.679747	12.899134	13.118521
	13.337908	13.557295	13.776682	13.996070	14.215457
	14.434844	14.654231	14.873618	15.093005	15.312392
	15.531780	15.751167	15.970554	16.189941	16.409328
	16.628715	16.848102	17.067489	17.286877	17.506264
parmname	Radius				
rslo	5.362028	5.389747	5.420434	5.457386	5.500269
	5.546825	5.594395	5.640565	5.683004	5.719951
	5.750893	5.775476	5.793916	5.807358	5.816899
	5.823565	5.828091	5.831034	5.833065	5.834527
	5.835635	5.836469	5.836996	5.836986	5.836640
	5.836273	5.836223	5.837216	5.838722	5.842403
	5.846898	5.854175	5.862536	5.875341	5.888147
	5.907170	5.926415	5.948395	5.972991	5.997588
	6.026151	6.055633	6.085115	6.118328	6.152235
	6.186142	6.222389	6.260305	6.298221	6.336137
blank					
filedesc	Flowpath: Outside Diameter Wall - TIP				
nduct_t	35				
parmname	Axial				
ztip	-6.502327	-6.232568	-5.962808	-5.693049	-5.423289
	-5.153530	-4.883770	-4.614011	-4.344251	-4.074492
	-3.804733	-3.534973	-3.265214	-2.995454	-2.725695
	-2.455935	-2.186176	-1.916416	-1.646657	-1.376897
	-1.107138	-0.837379	-0.567619	-0.297860	-0.028100
	0.241659	0.511419	0.781178	1.050938	1.320697
	1.590456	1.860216	2.129975	2.399735	2.669494
	2.939254	3.209013	3.478773	3.748532	4.018291
	4.288051	4.557810	4.827570	5.097329	5.367089
	5.636848	5.906608	6.176367	6.446126	6.715886
	6.985645	7.255405	7.525164	7.794924	8.064683
	8.334443	8.604202	8.873961	9.143721	9.413480
	9.683240	9.952999	10.222759	10.492518	10.762278
	11.032037	11.301796	11.571556	11.841315	12.111075
	12.380834	12.650594	12.920353	13.190113	13.459872
	13.729631	13.999391	14.269150	14.538910	14.808669
	15.078429	15.348188	15.617948	15.887707	16.157467
	16.427226	16.696985	16.966745	17.236504	17.506264
parmname	Radius				
rtip	11.010941	11.012493	11.013998	11.015268	11.016401
	11.017396	11.018124	11.018691	11.019068	11.019232
	11.019168	11.018869	11.018327	11.017558	11.016528
	11.015241	11.013677	11.011839	11.011221	11.006656
	10.995469	10.981015	10.966768	10.953494	10.940141
	10.926359	10.912204	10.898097	10.884698	10.873096
	10.865967	10.865466	10.870844	10.879611	10.891271
	10.905508	10.922375	10.941645	10.962651	10.984550
	11.006749	11.028477	11.048728	11.067650	11.085357
	11.101904	11.117003	11.130196	11.141621	11.150951
	11.157653	11.160125	11.156557	11.145514	11.127017
	11.104327	11.080913	11.059781	11.042190	11.028078
	11.017560	11.010750	11.006802	11.004715	11.003883
	11.003863	11.003966	11.003947	11.003903	11.003971
	11.003982	11.003810	11.003820	11.006583	11.013209
	11.022273	11.036808	11.051996	11.069672	11.087250
	11.104318	11.121385	11.134995	11.148494	11.158052
	11.164576	11.171100	11.168430	11.164928	11.161425

Note: the shaded column is not part of the input file

D.1 FORTRAN Code Used to Read File

```
    read(unit,'(a50)')  filedescr             ! hub surface description
    read(unit,*)        nduct_h               ! number of points along hub
    read(unit,'(a50)')  parmname              ! hub axial coordinate name
    read(unit,*)        (zhub(i),i=1,nduct_h) ! hub axial coordinates
    read(unit,'(a50)')  parmname              ! hub radial coordinate name
    read(unit,*)        (rhub(i),i=1,nduct_h) ! hub radial coordinates
    read(unit,'(a1)')   blank                 ! blank line
    read(unit,'(a50)')  filedescr             ! upper splitter surface description
    read(unit,*)        nsplt_u               ! number of points along upper splitter
    read(unit,'(a50)')  parmname              ! upper splitter axial coordinate name
    read(unit,*)        (zsup(i),i=1,nsplt_u) ! upper splitter axial coordinates
    read(unit,'(a50)')  parmname              ! upper splitter radial coordinate name
    read(unit,*)        (rsup(i),i=1,nsplt_u) ! upper splitter radial coordinates
    read(unit,'(a1)')   blank                 ! blank line
    read(unit,'(a50)')  filedescr             ! lower splitter surface description
    read(unit,*)        nsplt_l               ! number of points along lower splitter
    read(unit,'(a50)')  parmname              ! lower splitter axial coordinate name
    read(unit,*)        (zslo(i),i=1,nsplt_l) ! lower splitter axial coordinates
    read(unit,'(a50)')  parmname              ! lower splitter radial coordinate name
    read(unit,*)        (rslo(i),i=1,nsplt_l) ! lower splitter radial coordinates
    read(unit,'(a1)')   blank                 ! blank line
    read(unit,'(a50)')  filedescr             ! shroud surface description
    read(unit,*)        nduct_t               ! number of points along shroud
    read(unit,'(a50)')  parmname              ! shroud axial coordinate name
    read(unit,*)        (ztip(i),i=1,nduct_t) ! shroud axial coordinates
    read(unit,'(a50)')  parmname              ! shroud radial coordinate name
    read(unit,*)        (rtip(i),i=1,nduct_t) ! hub radial coordinates
```

Appendix E.—Sample Blade Geometry File

filedescr	NASA SOURCE DIAGNOSTICS FAN ROTOR AIRFOIL DATA				
numrle	51				
parmname	XLE				
xle	-3.300717	-3.303915	-3.308069	-3.313466	-3.320478
	-3.329237	-3.340175	-3.353832	-3.370878	-3.392150
	-3.418686	-3.451780	-3.493053	-3.544538	-3.608829
	-3.689264	-3.790090	-3.916614	-4.075167	-4.273249
	-4.520267	-4.828773	-5.215665	-5.700811	-6.305202
	-7.061386	-7.823350	-8.427943	-8.918634	-9.317897
	-9.638566	-9.894466	-10.098216	-10.260604	-10.390247
	-10.493849	-10.576728	-10.643064	-10.696157	-10.738645
	-10.772639	-10.799839	-10.821598	-10.839005	-10.852930
	-10.864068	-10.872978	-10.880105	-10.885588	-10.889806
	-10.893050				
parmname	YLE				
yle	0.056513	0.058546	0.061189	0.064622	0.069084
	0.074656	0.081611	0.090283	0.101076	0.114469
	0.131012	0.151285	0.175824	0.204982	0.238555
	0.275329	0.313901	0.353320	0.395976	0.448298
	0.516396	0.600231	0.691755	0.792850	0.932052
	1.099041	1.227651	1.360980	1.423135	1.429222
	1.423645	1.421848	1.426293	1.433713	1.441308
	1.448014	1.453255	1.457092	1.459912	1.462011
	1.463594	1.464802	1.465734	1.466459	1.467026
	1.467472	1.467826	1.468104	1.468316	1.468480
	1.468602				
parmname	ZLE				
zle	-1.297397	-1.297990	-1.298759	-1.299756	-1.301048
	-1.302654	-1.304649	-1.307122	-1.310177	-1.313935
	-1.318526	-1.324082	-1.330689	-1.338260	-1.346318
	-1.353738	-1.359192	-1.361348	-1.360299	-1.357423
	-1.351905	-1.338306	-1.312679	-1.274927	-1.226397
	-1.165348	-1.119212	-1.100427	-1.086893	-1.073364
	-1.064533	-1.060213	-1.057601	-1.055210	-1.052501
	-1.049508	-1.046523	-1.043761	-1.041353	-1.039327
	-1.037654	-1.036288	-1.035179	-1.034283	-1.033560
	-1.032978	-1.032511	-1.032136	-1.031847	-1.031624
	-1.031453				
parmname	BETAT1				
betale	66.841801	66.815885	66.783103	66.742593	66.683989
	66.614494	66.518115	66.399710	66.240933	66.033458
	65.750624	65.367511	64.835773	64.115349	63.168008
	61.994470	60.685418	59.429298	58.195041	56.664610
	54.802715	53.044276	51.019784	48.935041	46.509141
	43.978135	41.697607	39.574998	38.087785	36.851772
	35.808748	35.100205	34.436097	33.761993	33.149026
	32.648950	32.246985	31.915624	31.643311	31.425093
	31.245763	31.101723	30.986595	30.895777	30.819186
	30.759661	30.714022	30.676131	30.646688	30.624129
	30.608167				
parmname	XTE				
xte	-4.088949	-4.092535	-4.097195	-4.103248	-4.111110
	-4.120854	-4.132927	-4.147878	-4.166385	-4.189279
	-4.217576	-4.252514	-4.295610	-4.348718	-4.414104
	-4.494511	-4.593260	-4.714419	-4.862988	-5.045069
	-5.268126	-5.541716	-5.878458	-6.292905	-6.801577
	-7.431083	-8.067842	-8.579360	-8.995760	-9.339889
	-9.620321	-9.846050	-10.027071	-10.172226	-10.288950
	-10.382961	-10.458827	-10.520140	-10.569674	-10.609668
	-10.641939	-10.667974	-10.688969	-10.705896	-10.719543
	-10.730546	-10.739413	-10.746562	-10.752061	-10.756291
	-10.759545				
parmname	YTE				

yte	0.595555	0.593313	0.590397	0.586604	0.581671
	0.575546	0.567943	0.558513	0.546824	0.532360
	0.514516	0.492602	0.465780	0.433002	0.392969
	0.344221	0.285327	0.214571	0.129390	0.026424
	-0.097081	-0.237919	-0.392177	-0.576410	-0.812905
	-1.075679	-1.269325	-1.430582	-1.533526	-1.565878
	-1.572818	-1.579427	-1.590009	-1.603047	-1.615155
	-1.625258	-1.632740	-1.637674	-1.640837	-1.642914
	-1.644310	-1.645274	-1.645954	-1.646446	-1.646806
	-1.647074	-1.647280	-1.647433	-1.647550	-1.647635
	-1.647700				
parmname	ZTE				
zte	1.490532	1.491784	1.493412	1.495524	1.498265
	1.501657	1.505851	1.511032	1.517420	1.525280
	1.534918	1.546681	1.560933	1.578008	1.598115
	1.621252	1.647092	1.675143	1.705382	1.738630
	1.773224	1.795254	1.785352	1.761071	1.718796
	1.631256	1.531697	1.440159	1.362147	1.302295
	1.252064	1.208955	1.174382	1.147466	1.126418
	1.109854	1.096681	1.086109	1.077594	1.070730
	1.065192	1.060726	1.057124	1.054220	1.051879
	1.049991	1.048470	1.047244	1.046300	1.045575
	1.045016				
parmname	BETAT2				
betate	123.481562	123.359068	123.200406	122.994211	122.723927
	122.390499	121.977237	121.463885	120.829013	120.048534
	119.084911	117.907573	116.482608	114.764199	112.735775
	110.376564	107.716961	104.740661	101.130447	96.493614
	90.782852	84.861986	79.986424	74.797613	68.264310
	59.838997	53.438499	49.173076	45.258556	43.025379
	42.047168	41.267618	40.301256	39.256656	38.427494
	37.915711	37.615648	37.431940	37.300851	37.201613
	37.121749	37.054726	37.000575	36.955608	36.922670
	36.894472	36.866245	36.849647	36.828629	36.820108
	36.812149				

Note: the shaded column is not part of the input file

E.1 FORTRAN Code Used to Read File

```
read(unit,'(a50)')  filedescr              ! file description
read(unit,*)        numrle                 ! number of points defining blade
read(unit,'(a50)')  parmname               ! l.e. x-coordinate name
read(unit,*)        (xle(i),i=1,numrle)    ! l.e. x-coordinates
read(unit,'(a50)')  parmname               ! l.e. y-coordinate name
read(unit,*)        (yle(i),i=1,numrle)    ! l.e. y-coordinates
read(unit,'(a50)')  parmname               ! l.e. z-coordinate name
read(unit,*)        (zle(i),i=1,numrle)    ! l.e. z-coordinates
read(unit,'(a50)')  parmname               ! l.e. metal angle name
read(unit,*)        (betale(i),i=1,numrle) ! l.e. metal angle
read(unit,'(a50)')  parmname               ! t.e. x-coordinate name
read(unit,*)        (xte(i),i=1,numrle)    ! t.e. x-coordinates
read(unit,'(a50)')  parmname               ! t.e. y-coordinate name
read(unit,*)        (yte(i),i=1,numrle)    ! t.e. y-coordinates
read(unit,'(a50)')  parmname               ! t.e. z-coordinate name
read(unit,*)        (zte(i),i=1,numrle)    ! t.e. z-coordinates
read(unit,'(a50)')  parmname               ! t.e. metal angle name
read(unit,*)        (betate(i),i=1,numrle) ! t.e. metal angle
```

Appendix F.—Sample Vane Geometry File

filedescr	NASA SOURCE DIAGNOSTICS VANE STATOR AIRFOIL DATA
numrle	51
parmname	XLE

xle	-5.416727 -5.419984 -5.424215 -5.429717 -5.436869
	-5.445656 -5.456452 -5.469716 -5.486012 -5.506033
	-5.530631 -5.560848 -5.597970 -5.643573 -5.699591
	-5.768402 -5.852918 -5.956717 -6.084190 -6.240739
	-6.433002 -6.669145 -6.959180 -7.315404 -7.752908
	-8.290226 -8.827341 -9.264373 -9.619956 -9.909304
	-10.144755 -10.336369 -10.492330 -10.619282 -10.722619
	-10.806724 -10.875172 -10.930878 -10.976212 -11.013106
	-11.043136 -11.067576 -11.087468 -11.103658 -11.116835
	-11.127559 -11.136287 -11.143391 -11.148856 -11.153060
	-11.156293

parmname	YLE

yle	-0.406363 -0.406194 -0.405973 -0.405686 -0.405312
	-0.404852 -0.404287 -0.403591 -0.402734 -0.401682
	-0.400392 -0.398819 -0.396905 -0.394579 -0.391767
	-0.388400 -0.384440 -0.379891 -0.374796 -0.369099
	-0.362613 -0.354976 -0.345888 -0.335125 -0.322678
	-0.308797 -0.296720 -0.288643 -0.284890 -0.283530
	-0.283985 -0.285139 -0.285764 -0.285726 -0.285371
	-0.285092 -0.285040 -0.285216 -0.285526 -0.285869
	-0.286190 -0.286472 -0.286714 -0.286918 -0.287089
	-0.287229 -0.287347 -0.287443 -0.287516 -0.287574
	-0.287618

parmname	ZLE

zle	7.267998 7.268016 7.268040 7.268072 7.268112
	7.268163 7.268225 7.268301 7.268394 7.268508
	7.268649 7.268822 7.269035 7.269298 7.269619
	7.270013 7.270496 7.271083 7.271793 7.272658
	7.273715 7.275023 7.276612 7.278489 7.280753
	7.283655 7.286515 7.288820 7.290805 7.292380
	7.293630 7.294653 7.295467 7.296127 7.296694
	7.297163 7.297535 7.297823 7.298047 7.298222
	7.298362 7.298473 7.298564 7.298636 7.298696
	7.298744 7.298783 7.298814 7.298839 7.298857
	7.298871

parmname	BETAT1

betale	131.281769 131.272356 131.246212 131.223061 131.190830
	131.152542 131.100057 131.044341 130.971196 130.874959
	130.761550 130.623424 130.456048 130.250116 129.982424
	129.669266 129.283355 128.854058 128.385733 127.861015
	127.242161 126.511595 125.711721 124.809808 123.857544
	123.024853 122.618754 122.587236 122.795412 123.133131
	123.582709 124.028165 124.490860 124.887123 125.190549
	125.514202 125.867713 126.185061 126.509030 126.772852
	127.005851 127.182839 127.330880 127.471321 127.568032
	127.651582 127.713520 127.773353 127.810731 127.848052
	127.867114

parmname	XTE

xte	-5.731794 -5.735029 -5.739233 -5.744697 -5.751801
	-5.760493 -5.771126 -5.784135 -5.800049 -5.819518
	-5.843336 -5.872474 -5.908123 -5.951734 -6.005087
	-6.070358 -6.150211 -6.247900 -6.367413 -6.513622
	-6.692492 -6.911319 -7.179029 -7.506539 -7.907212
	-8.397385 -8.887561 -9.288233 -9.615743 -9.883452
	-10.102280 -10.281151 -10.427359 -10.546871 -10.644561
	-10.724414 -10.789686 -10.843039 -10.886650 -10.922299
	-10.951438 -10.975257 -10.994726 -11.010640 -11.023649
	-11.034283 -11.042974 -11.050076 -11.055543 -11.059748
	-11.062981

parmname	YTE

yte	0.000471	0.000471	0.000471	0.000472	0.000471
	0.000470	0.000470	0.000469	0.000469	0.000468
	0.000469	0.000470	0.000471	0.000473	0.000473
	0.000470	0.000468	0.000469	0.000473	0.000473
	0.000472	0.000471	0.000468	0.000475	0.000471
	0.000464	0.000464	0.000468	0.000471	0.000460
	0.000467	0.000467	0.000462	0.000457	0.000460
	0.000466	0.000469	0.000470	0.000467	0.000464
	0.000463	0.000463	0.000463	0.000463	0.000466
	0.000465	0.000467	0.000467	0.000467	0.000468
	0.000468				
parmname	ZTE				
zte	8.809691	8.809727	8.809773	8.809833	8.809911
	8.810006	8.810122	8.810264	8.810437	8.810648
	8.810906	8.811219	8.811600	8.812058	8.812607
	8.813257	8.814007	8.814852	8.815793	8.816864
	8.818124	8.819621	8.821311	8.823187	8.825278
	8.827568	8.829410	8.830547	8.831099	8.831244
	8.831086	8.830804	8.830573	8.830452	8.830420
	8.830431	8.830447	8.830440	8.830409	8.830365
	8.830317	8.830269	8.830224	8.830184	8.830150
	8.830121	8.830096	8.830075	8.830059	8.830048
	8.830036				
parmname	BETAT2				
betate	86.977677	86.973932	86.976088	86.973618	86.968883
	86.975419	86.969165	86.970769	86.963695	86.964786
	86.966653	86.968617	86.977814	86.980755	86.998967
	86.987708	86.995244	86.996859	87.022108	87.024978
	87.028719	87.050052	87.087533	87.184788	87.230916
	87.323123	87.347619	87.449030	87.501262	87.537642
	87.590556	87.606019	87.640125	87.669546	87.656125
	87.649347	87.658172	87.670551	87.678676	87.676479
	87.700663	87.709592	87.709429	87.701384	87.694857
	87.696417	87.686811	87.693660	87.671975	87.677902
	87.682160				

Note: the shaded column is not part of the input file

F.1 FORTRAN Code Used to Read File

```
read(unit,'(a50)')  filedescr              ! file description
read(unit,*)        numrle                 ! number of points defining vane
read(unit,'(a50)')  parmname               ! l.e. x-coordinate name
read(unit,*)        (xle(i),i=1,numrle)    ! l.e. x-coordinates
read(unit,'(a50)')  parmname               ! l.e. y-coordinate name
read(unit,*)        (yle(i),i=1,numrle)    ! l.e. y-coordinates
read(unit,'(a50)')  parmname               ! l.e. z-coordinate name
read(unit,*)        (zle(i),i=1,numrle)    ! l.e. z-coordinates
read(unit,'(a50)')  parmname               ! l.e. metal angle name
read(unit,*)        (betale(i),i=1,numrle) ! l.e. metal angle
read(unit,'(a50)')  parmname               ! t.e. x-coordinate name
read(unit,*)        (xte(i),i=1,numrle)    ! t.e. x-coordinates
read(unit,'(a50)')  string                 ! t.e. y-coordinate name
read(unit,*)        (yte(i),i=1,numrle)    ! t.e. y-coordinates
read(unit,'(a50)')  parmname               ! t.e. z-coordinate name
read(unit,*)        (zte(i),i=1,numrle)    ! t.e. z-coordinates
read(unit,'(a50)')  parmname               ! t.e. metal angle name
read(unit,*)        (betate(i),i=1,numrle) ! t.e. metal angle
```

Appendix G.—FORTRAN Code Used to Read Flow-Field File (Viscous CFD and Experimental Data)

```
      read(unitnum,*)     iwaketyp
      read(unitnum,*)     omega,xprtip,rhub,rtip,npassage,phipr
      read(unitnum,*)     numr,numt
      read(unitnum,*)     (z(i),i=1,numr)
      read(unitnum,*)     ((x(i,j),i=1,numr),j=1,numt)
      read(unitnum,*)     ((y(i,j),i=1,numr),j=1,numt)
      read(unitnum,*)     icz,icu,icr,itke,ieps,icpzcpz,icpucpu,
     &                    icprcpr,icpzcpu,icpzcpr,icpucpr,iscale
      if (icz .eq. 1) then
           read(unitnum,*)  ((cz(i,j),i=1,numr),j=1,numt)
      endif
      if (icu .eq. 1) then
           read(unitnum,*)  ((cu(i,j),i=1,numr),j=1,numt)
      endif
      if (icr .eq. 1) then
           read(unitnum,*)  ((cr(i,j),i=1,numr),j=1,numt)
      endif
      if (itke .eq. 1) then
           read(unitnum,*)  ((tke(i,j),i=1,numr),j=1,numt)
      endif
      if (ieps .eq. 1) then
           read(unitnum,*)  ((eps(i,j),i=1,numr),j=1,numt)
      endif
      if (icpzcpz .eq. 1) then
           read(unitnum,*)  ((cpzcpz(i,j),i=1,numr),j=1,numt)
      endif
      if (icpucpu .eq. 1) then
           read(unitnum,*)  ((cpucpu(i,j),i=1,numr),j=1,numt)
      endif
      if (icprcpr .eq. 1) then
           read(unitnum,*)  ((cprcpr(i,j),i=1,numr),j=1,numt)
      endif
      if (icpzcpu .eq. 1) then
           read(unitnum,*)  ((cpzcpu(i,j),i=1,numr),j=1,numt)
      endif
      if (icpzcpr .eq. 1) then
           read(unitnum,*)  ((cpzcpr(i,j),i=1,numr),j=1,numt)
      endif
      if (icpucpr .eq. 1) then
           read(unitnum,*)  ((cpucpr(i,j),i=1,numr),j=1,numt)
      endif
      if (iscale .eq. 1) then
           read(unitnum,*)  (scale(i),i=1,numr)
      endif
      read (unitnum,*)      ztip2,rhub2,rtip2
      read (unitnum,*)      numr2
      read (unitnum,*)      (r2(i),i=1,numr2)
      read (unitnum,*)      (z2(i),i=1,numr2)
      read (unitnum,*)      (pt2(i),i=1,numr2)
      read (unitnum,*)      (tt2(i),i=1,numr2)
```

Appendix H.—FORTRAN Code Used to Read Flow-Field File (Inviscid Streamline Code)

```
read (unitnum,'(a50)') filedescr
read (unitnum,*)       numr
read (unitnum,'(a50)') parmname
read (unitnum,*)       (diam(i),i=1,numr)
read (unitnum,'(a50)') parmname
read (unitnum,*)       (z(i),i=1,numr)
read (unitnum,'(a50)') parmname
read (unitnum,*)       (u(i),i=1,numr)
do i = 1,numr
  r(i) = diam(i)/2.
  z(i) = z(i) - zblade/in_
enddo
numt = 1
rhub = r(1)
rtip = r(numr)
omega = (u(1)*12.)/r(1)
read (unitnum,'(a50)') parmname
read (unitnum,*)       ((cz(i,j),i=1,numr),j=1,numt)
read (unitnum,'(a50)') parmname
read (unitnum,*)       ((ct(i,j),i=1,numr),j=1,numt)
read (unitnum,'(a50)') parmname
read (unitnum,*)       ((cr(i,j),i=1,numr),j=1,numt)
read (unitnum,'(a50)') parmname
read (unitnum,*)       (loss(i),i=1,numr)
read (unitnum,'(a50)') parmname
read (unitnum,*)       (gapchord(i),i=1,numr)
numr2 = numr1
rhub2 = rhub1/in_
rtip2 = rtip1/in_
read (unitnum,'(a50)') parmname
read (unitnum,*)       (pt2(i),i=1,numr2)
read (unitnum,'(a50)') parmname
read (unitnum,*)       (tt2(i),i=1,numr2)
do i = 1,numr2
  r2(i) = r1(i)/in_
enddo
```

Appendix I.—Sample Setup_BFaNS Input File

```
case_54vr_ap.data             : 10.  DUCT GEOMETRY FILE
1.0450                        : 20.  AXIAL COORDINATE OF BLADE TRAILING EDGE TIP (IN)
8.8300                        : 30.  AXIAL COORDINATE OF VANE TRAILING EDGE TIP (IN)
-6.5023                       : 40.  AXIAL COORDINATE OF HILITE (IN)
-4.0074                       : 50.  AXIAL COORDINATE OF NOSE CONE (IN)
6.7563                        : 60.  AXIAL COORDINATE OF CORE INLET (IN)
17.5063                       : 70.  AXIAL COORDINATE OF DUCT EXIT (IN)
rotor_54vr_ap.data            : 80.  ROTOR GEOMETRY FILE
22                            : 90.  BLADE NUMBER
-1                            : 100. DIRECTION OF ROTATION FROM REAR (1: CCW, -1: CW)
stator_54vr_ap.data           : 110. STATOR GEOMETRY FILE
54                            : 120. VANE NUMBER
wake_54vr_ap_rotorle.data     : 130. ROTOR UPSTREAM FLOWFIELD FILE
2                             : 140. PRECEDING FILE TYPE (1=SLINE, 2=CFD, 3=EXPER)
wake_54vr_ap_rotorte.data     : 150. ROTOR DOWNSTREAM FLOWFIELD FILE
2                             : 160. PRECEDING FILE TYPE (1=SLINE, 2=CFD, 3=EXPER)
wake_54vr_ap_statorle.data    : 170. STATOR UPSTREAM FLOWFIELD FILE
2                             : 180. PRECEDING FILE TYPE (1=SLINE, 2=CFD, 3=EXPER)
wake_54vr_ap_statorte.data    : 190. STATOR DOWNSTREAM FLOWFIELD FILE
2                             : 200. PRECEDING FILE TYPE (1=SLINE, 2=CFD, 3=EXPER)
0.06                          : 210. B.L. THKNESS AT BLADE HUB (IN) (0 FOR AUTOCALC)
0.09                          : 220. B.L. THKNESS AT BLADE TIP (IN) (0 FOR AUTOCALC)
0.13                          : 230. B.L. THKNESS AT VANE HUB (IN) (0 FOR AUTOCALC)
0.64                          : 240. B.L. THKNESS AT VANE TIP (IN) (0 FOR AUTOCALC)
54-Vane Radial Stator         : 250. RUN TITLE #1
Approach Condition (7,808 rpm): 260. RUN TITLE #2
```

Notes:

1. Numeric input is free format.
2. Character input is formatted as A30, and must not be in quotes.
3. The first 30 columns are reserved for user input.
4. Comments may be added in columns 31 and greater.
5. The blade and vane number must correspond to the blade and vanes numbers used to generate the flow field files

Appendix J.—Sample Setup_BFaNS Author File

```
c-------------------------------------------------------------------------------
c author file for setup_bfans
c-------------------------------------------------------------------------------
c section #1: read format is (a1,i5,a20,a54)
c-------------------------------------------------------------------------------
c card description      : a54
c-------------------------------------------------------------------------------
 -5000 input file name  : setupbfans.input
 -5002 geometry dir     : /bfans_geometry/
 -5004 streamline dir   : /bfans_sline/
 -5006 cfd dir          : /bfans_cfd/
 -5008 data dir         : /bfans_data/
 -5010 source dir       : ~/Codes/fans_nasa/setup_bfans/
 -5012 through 5048     : reserved for future development
c-------------------------------------------------------------------------------
c section #2: read format is (a1,i5,a45,2i4,2f8,a5)
c-------------------------------------------------------------------------------
c card description                                :  i4  i4      f8      f8   a5
c-------------------------------------------------------------------------------
 -5050 create debug files                         :
 -5052 do not use measured scale at station #1    :
 -5054 do not use measured scale at station #3    :
 -5056 over-ride scale * epsilon / tke**1.5       :            1.00
    -1 defaults to 1.00                           :
 -5058 tke fudge factor                           :            1.00
    -1 defaults to 1.00                           :
 -5060 epsilon fudge factor:                      :            1.00
    -1 defaults to (tke fudge factor)**1.5        :
 -5062 over-ride scale cal. factor at blade hub   :            0.62
    -1 rmoda = cal. factor (default = 0.62)       :
 -5064 over-ride scale cal. factor at blade tip   :            0.62
    -1 rmoda = cal. factor (default = 0.62)       :
 -5066 over-ride scale cal. factor at vane hub    :            0.62
    -1 rmoda = cal. factor (default = 0.62)       :
 -5068 over-ride scale cal. factor at vane tip    :            0.62
    -1 rmoda = cal. factor (default = 0.62)       :
 -5070 over-ride scale cal. factor for blade wake :            0.68
    -1 rmoda = cal. factor (default = 0.68)       :
 -5072 over-ride order of wake curve fit          :   5
    -1 default = 5                                :
 -5074 over-ride tol. for unwrapping blade wake   :            0.90
    -1 rmoda = tolerance (def = 0.9)              :
 -5076 write wake lean to output file             :
 -5078 account for wake lean in wake width        :
 -5080 do not account for streamtube contraction  :
    -1 in cascade parameters                      :
 -5082 use reynolds stress tensor to compute      :
    -1 upwash (rather than TKE)                   :
 -5084 do not check to see if cfd or experimental :
    -1 data covers one full pitch                 :
  9999 end of author file
c-------------------------------------------------------------------------------
```

Appendix K.—Directory Structure of ~/Codes/fans_nasa

~/Codes/fans_nasa/				
bfans/	lib/	lib/grafic/		setup_bfans/
Makefile.Ibm6000	libgrafic.a.Ibm6000	Apollo.m4	grflip.m4	Makefile.Ibm6000
Makefile.Iris	libgrafic.a.Iris	Athena.m4	grgrey.m4	Makefile.Iris
Makefile.Sun4	libgrafic.a.Sun4	Aviion.m4	grgri1.m4	Makefile.Sun4
Makefile.wnt		Dec.m4	grgrid.m4	add_points.F
angles.F	makeflags/	Hp9000.m4	grinit.m4	angles.F
author.def	makeflags.Ibm6000	Ibm6000.m4	grinpa.m4	arrowhead.F
bbcascade.F	makeflags.Iris	Iris.m4	grinpf.m4	author.def
bf_author.m	makeflags.Sun4	Makefile.Ibm6000	grinpi.m4	cal_coeff.F
bfans_ibm6000.F	makeflags.wnt	Makefile.Iris	grkey1.m4	calc_bl.F
bfans_iris.F		Makefile.Sun4	grkey2.m4	calc_rotor_wake.F
bfans_sun4.F		Newsys.m4	grkey3.m4	cascade.F
bfans_wnt.F		Stellar.m4	grkli1.m4	chan_duct_geom.F
capz.F		Sun4.m4	grklin.m4	circum_average.F
cascade2.F		Titan.m4	grklst.m4	constants.def
constants.def		Vapor.m4	grkpla.m4	converts.def
converts.def		Vax.m4	grlin1.m4	cross_ref.F
cross_ref.F		XApollo.c	grline.m4	debug_add_points.F
dateandtime.F		XAthena.c	grloc2.m4	debug_calc_bl.F
error.F		XAviion.c	grloct.m4	debug_calc_rotor_wake.F
exact_sol_hub.F		XDec.c	grmesg.m4	debug_estimate.F
f_ex.F		XHp9000.c	grmode.m4	debug_geom.F
f_ex2.F		XIbm6000.c	grmon2.m4	debug_get_bl.F
f_plus_minus2.F		XIris.c	grmon3.m4	debug_get_duct_geom.F
fbessj.F		XNewsys.c	grmov1.m4	debug_get_flow_data.F
fbessj0.F		XStellar.c	grmov2.m4	debug_get_geom.F
fbessj1.F		XSun4.c	grmov3.m4	debug_get_rotor_wake.F
fbessy.F		XTitan.c	grmovy.m4	debug_get_row_geom.F
fbessy0.F		XVapor.c	grnear.m4	debug_intersect.F
fbessy1.F		contng.m4	grper1.m4	debug_normalize.F
fgen.F		conxch.m4	grpers.m4	debug_scales.F
get_alpha_range.F		gr2alst.m4	grplst.m4	debug_stream_line.F
get_freqs.F		gr2axes.m4	grply2.m4	disp.F
get_input.F		gr2ctrl.m4	grply3.m4	error.F
get_nu_range.F		gr2curs.m4	grprnt.m4	estimate2.F
get_pratios_reba.F		gr2get.m4	grscal.m4	flow.F
getalphas.F		gr2lin1.m4	grscpt.m4	geom.F
getdeltas.F		gr2line.m4	grsfac.m4	get_bl.F
input_bb.F		gr2plst.m4	grsset.m4	get_duct_geom.F
input_new.F		gr2set.m4	grsymb.m4	get_file_name.F
interp.F		gr2sfac.m4	grthr1.m4	get_flow_data.F
jsp_te.F		gr2sset.m4	grthre.m4	get_geom.F
jsplit.F		grafic.doc	grtime.m4	get_mesh_data.F
lusolve.F		gralst.m4	grtran.m4	get_rho_sound.F
modes.F		granot.m4	grvalu.m4	get_rotor_wake.F
mysqrt.F		graxes.m4	grvec1.m4	get_row_geom.F
nasa_s.F		grcfil.m4	grvect.m4	get_velocity_data.F
parameters.def		grcler.m4	mrcon1.m4	interp.F
parameters.fgen.def		grclos.m4	mrcont.m4	intersect2.F
power_ratio.F		grcnew.inc		leasqr.F
read_author.F		grcntr.m4		ludcmq.F
read_input.F		grcol2.m4		mkdir_matlab.F
response.F		grcol3.m4		naca0012.F
run_bbcasc.F		grcolr.m4		normalize.F
s_pred.F		grcomn.inc		parabola.F
s_spec.F		grcon1.m4		parameters.def
trapezoid.F		grcon2.m4		plot_bl.m
turbspect.F		grcont.m4		plot_bv.F
zeros_ex_sub.F		grctrl.m4		plot_geom.F
		grcube.m4		plot_wake.m
		grcurs.m4		read_author.F
		grcutt.m4		reduce.m
		grdash.m4		save_get_bl.F
		grdrw2.m4		sbf_author.m

~/Codes/fans_nasa/				
bfans/	lib/	lib/grafic/		setup_bfans/
		grdrw3.m4		scales.F
		grdump.m4		setupbfans.F
		gredi1.m4		solnq.F
		gredit.m4		sort.F
		greror.m4		stream_function.F
		grfil2.m4		stream_line.F
		grfil3.m4		stress.F
				stress.m
				triangle.F
				upwash.F
				vect_comp.F
				wake_lean.F
				write_file.F

Appendix L.—Setup_BFaNS Subroutine Hierarchy

Bold items are called by multiple routines

- I. setup_bfans
  - A. read_author
    - 1. cross_ref
  - B. mkdir_matlab
  - C. get_geom
    - **1. disp**
    - 2. get_duct_geom
      - **a. get_file_name**
      - **b. error**
      - c. sort
      - d. debug_get_duct_geom
    - 3. get_row_geom
      - **a. get_file_name**
      - **b. error**
      - c. debug_get_row_geom
    - 4. debug_get_geom
  - **D. disp**
  - E. get_flow_data
    - **1. get_file_name**
    - 2. get_mesh_data
      - **a. error**
    - 3. get_velocity_data
      - **a. error**
      - b. stress
    - **4. circum_average**
      - **a. error**
    - 5. get_rho_sound
      - **a. error**
      - **b. interp**
    - 6. stream_function
      - **a. error**
    - 7. debug_get_flow_data
  - F. add_points
    - **1. error**
    - 2. debug_add_points
  - G. normalize
    - **1. error**
    - 2. debug_normalize
  - H. stream_line
    - **1. error**
    - **2. interp**
    - 3. debug_stream_line
  - I. intersect2
    - **1. error**
    - **2. interp**
    - 3. debug_intersect2
  - J. estimate2
    - **1. error**
    - **2. interp**
    - 3. debug_estimate2
  - **K. circum_average**
    - **1. error**
  - L. triangle
    - 1. cal_coeff
    - 2. vect_comp
    - 3. plot_bv
      - **a. grinit**
      - b. arrowhead
      - **c. grscpt**
      - **d. grline**
  - M. geom
    - **1. error**
    - 2. parabola
    - 3. debug_geom
  - N. flow
    - **1. error**
  - O. cascade
    - **1. error**
  - P. naca0012
  - Q. angles
  - R. upwash
  - S. get_bl
    - **1. error**
    - **2. interp**
    - 3. save_get_bl
    - 4. debug_get_bl
  - T. calc_bl
    - 1. debug_calc_bl
  - U. get_rotor_wake
    - **1. error**
    - **2. interp**
    - 3. debug_get_rotor_wake
  - V. wake_lean
    - 1. leasqr
      - a. ludcmq
      - b. solnq
  - W. calc_rotor_wake
    - **1. error**
    - **2. interp**
    - 3. debug_calc_rotor_wake
  - X. chan_duct_geom
    - **1. error**
    - 2. plot_geom
      - **a. grinit**
      - **b. interp**
      - **c. error**
      - **d. gr2sset**
      - **e. gr2line**
  - Y. scales
    - 1. debug_scales
  - Z. write_file
    - **1. error**

Appendix M.—Cascade Parameters

For a two-dimensional cascade with incompressible flow, the cascade loss, drag and lift coefficients coefficient can be determined from,

$$\overline{\omega} = \frac{2}{\cos\beta_{te}}\left(\frac{\delta_2}{b}\right)\left(\frac{b}{\tau}\right)\left(\frac{\cos\beta_{le}}{\cos\beta_{te}}\right)^2 \qquad \text{(see Ref. 11, Eq. (3.3)2)}$$

$$C_D = \overline{\omega}\frac{\tau}{b}\frac{\cos^3\beta_m}{\cos^2\beta_{le}} \qquad \text{(see Ref. 11, Eq. (3.33))}$$

$$C_L = 2\frac{\tau}{b}\left(\tan\beta_{le} - \tan\beta_{te}\right)\cos\beta_m - C_D\tan\beta_m \qquad \text{(see Ref. 11, Eq. (3.18))}$$

$$\beta_m = \tan^{-1}\left(\frac{\tan\beta_{le} + \tan\beta_{te}}{2}\right) \qquad \text{(see Ref. 11, Eq. (3.6))}$$

These incompressible cascade relationships assume that the axial velocity at the trailing edge of the cascade is the same as that at the leading edge. However, in a real engine, the axial velocity may change significantly across the cascade due to contraction in the flow path. The following derivation attempts to include the effect of streamtube contraction on the cascade parameters. The derivation assumes steady, incompressible flow through a stationary, rectilinear cascade. The flow is assumed to be uniform upstream and downstream from the cascade.

M.1 Total Pressure Loss

The cascade loss coefficient will be defined as,

$$\overline{\omega} = \frac{\left(P_1 + \frac{1}{2}\rho W_1^2\right) - \left(P_2 + \frac{1}{2}\rho W_2^2\right)}{\frac{1}{2}\rho W_1^2}$$

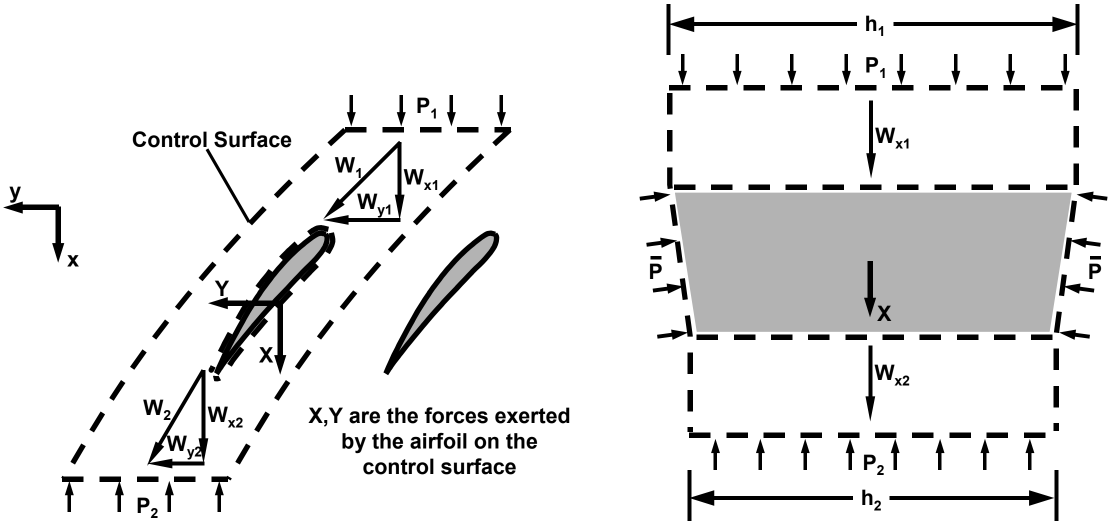

Control Surface
y
x
P1
W1
Wx1
Wy1
Y
X
W2
Wx2
Wy2
P2
X,Y are the forces exerted by the airfoil on the control surface
h1
P1
Wx1
P̄
P̄
X
Wx2
P2
h2

Figure M.1.—Cascade control volume.

From the loss coefficient, the pressure difference across the cascade is,

$$P_1 - P_2 = \frac{1}{2}\rho\left[\left(\overline{\omega}-1\right)W_1^2 + W_2^2\right]$$

$$P_1 - P_2 = \frac{1}{2}\rho\left[\left(\overline{\omega}-1\right)\left(\frac{W_{x,1}}{\cos\beta_1}\right)^2 + \left(\frac{W_{x,2}}{\cos\beta_2}\right)^2\right]$$

$$P_1 - P_2 = \frac{1}{2}\rho W_{x,1}^2\left[\left(\overline{\omega}-1\right)\left(\frac{1}{\cos\beta_1}\right)^2 + \left(\frac{W_{x,2}}{W_{x,1}}\right)^2\left(\frac{1}{\cos\beta_2}\right)^2\right]$$

If we define $\chi = W_{x,2}/W_{x,1}$, then

$$P_1 - P_2 = \frac{1}{2}\rho W_{x,1}^2\left[\left(\overline{\omega}-1\right)\left(\frac{1}{\cos\beta_1}\right)^2 + \left(\frac{\chi}{\cos\beta_2}\right)^2\right]$$

M.2 Lift and Drag Definition

We will define the mean velocity and mean angle such that,

$$W_m = \frac{W_{x,1}}{\cos\beta_m}$$

The equation used to determine β_m will be derived later. We will define lift as the force component that is perpendicular to the mean direction, and drag as the force component that is parallel to the mean direction.

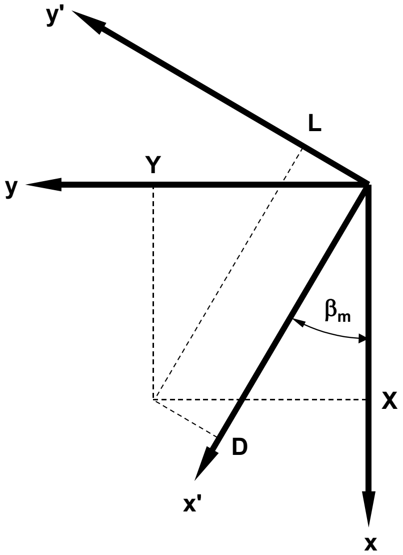


Figure M.2.—Lift and drag forces.

The lift and drag are related to axial and tangential forces via,

$$\begin{pmatrix} D \\ L \end{pmatrix} = \begin{pmatrix} \cos\beta_m & \sin\beta_m \\ -\sin\beta_m & \cos\beta_m \end{pmatrix} \begin{pmatrix} X \\ Y \end{pmatrix}$$

or inversely

$$\begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} \cos\beta_m & -\sin\beta_m \\ \sin\beta_m & \cos\beta_m \end{pmatrix} \begin{pmatrix} D \\ L \end{pmatrix}$$

We will define the lift and drag coefficients as,

$$\hat{C}_L = \frac{L}{\frac{1}{2}\rho W_m^2 b h_1} \quad \text{and} \quad \hat{C}_D = \frac{D}{\frac{1}{2}\rho W_m^2 b h_1}$$

The "hats" on $\hat{C}_L$ and $\hat{C}_D$ indicate that the lift and drag are defined as being positive in the *x*' and *y*' directions shown in Figure M.2 In reality, the lift and drag forces on the control surface are in the opposite direction to those shown in Figure M.2. Therefore, the actual lift and drag coefficients (without the "hats") are:

$$C_L = \frac{-L}{\frac{1}{2}\rho W_m^2 b h_1} \quad \textit{and} \quad C_D = \frac{-D}{\frac{1}{2}\rho W_m^2 b h_1}$$

We will define the axial and tangential force coefficients as,

$$C_X = \frac{X}{\frac{1}{2}\rho W_{x,1}^2 \tau h_1} \quad \textit{and} \quad C_Y = \frac{Y}{\frac{1}{2}\rho W_{x,1}^2 \tau h_1}$$

The lift and drag coefficients are related to the axial and tangential force coefficients via,

$$\begin{pmatrix} -C_D \\ -C_L \end{pmatrix} = \left(\frac{W_{x,1}}{W_m}\right)^2 \frac{\tau}{b} \begin{pmatrix} \cos\beta_m & \sin\beta_m \\ -\sin\beta_m & \cos\beta_m \end{pmatrix} \begin{pmatrix} C_X \\ C_Y \end{pmatrix}$$

or inversely

$$\begin{pmatrix} C_X \\ C_Y \end{pmatrix} = \left(\frac{W_m}{W_{x,1}}\right)^2 \frac{b}{\tau} \begin{pmatrix} \cos\beta_m & -\sin\beta_m \\ \sin\beta_m & \cos\beta_m \end{pmatrix} \begin{pmatrix} -C_D \\ -C_L \end{pmatrix}$$

M.3 Axial Momentum Equation

$$\sum F_x = \oint_A W_x \rho \vec{W} \cdot d\vec{A}$$

$$X + P_1 A_1 - P_2 A_2 - \overline{P}(A_1 - A_2) = W_{x,2}(\rho W_{x,2} A_2) - W_{x,1}(\rho W_{x,1} A_1)$$

From mass conservation, we know that $\rho W_{x,2} A_2 = \rho W_{x,1} A_1$. Therefore,

$$X + P_1 A_1 - P_2 A_2 - \overline{P}\left(A_1 - A_2\right) = \rho W_{x,1} A_1 \left(W_{x,2} - W_{x,1}\right)$$

$$X = P_2 A_2 - P_1 A_1 + \overline{P}\left(A_1 - A_2\right) + \rho W_{x,1}^2 A_1 \left(\chi - 1\right)$$

The average pressure on the endwalls will depend on the endwall geometry. For simplicity, we will assume that,

$$\overline{P} = \frac{1}{2}\left(P_1 + P_2\right)$$

Thus, the *x*-momentum equation becomes,

$$X = \frac{1}{2}\left(P_2 - P_1\right) A_1 \left(1 + \frac{A_2}{A_1}\right) + \rho W_{x,1}^2 A_1 \left(\chi - 1\right)$$

After substituting the pressure difference from the loss equation, we get,

$$X = -\frac{1}{4} \rho W_{x,1}^2 A_1 \left[\left(\overline{\omega} - 1\right)\left(\frac{1}{\cos\beta_1}\right)^2 + \left(\frac{\chi}{\cos\beta_2}\right)^2\right]\left(1 + \frac{A_2}{A_1}\right) + \rho W_{x,1}^2 A_1 \left(\chi - 1\right)$$

Therefore, the axial force coefficient is,

$$C_X = \frac{X}{\frac{1}{2}\rho W_{x,1}^2 \tau h_1} = -\frac{1}{2}\frac{A_1}{\tau h_1}\left[\left(\overline{\omega} - 1\right)\left(\frac{1}{\cos\beta_1}\right)^2 + \left(\frac{\chi}{\cos\beta_2}\right)^2\right]\left(1 + \frac{A_2}{A_1}\right) + 2\frac{A_1}{\tau h_1}\left(1 - \chi\right)$$

Since $A_1 = \tau h_1$, then,

$$C_X = \frac{X}{\frac{1}{2}\rho W_{x,1}^2 \tau h_1} = -\frac{1}{2}\left[\left(\overline{\omega} - 1\right)\left(\frac{1}{\cos\beta_1}\right)^2 + \left(\frac{\chi}{\cos\beta_2}\right)^2\right]\left(1 + \frac{A_2}{A_1}\right) + 2\left(1 - \chi\right)$$

From mass conservation for incompressible flow, the area ratio is related to the velocity ratio via,

$$\frac{A_2}{A_1} = \frac{W_{x,1}}{W_{x,2}} = \frac{1}{\chi}$$

Therefore, the axial force coefficient becomes,

$$C_X = -\frac{1}{2}\left[\left(\overline{\omega} - 1\right)\left(\frac{1}{\cos\beta_1}\right)^2 + \left(\frac{\chi}{\cos\beta_2}\right)^2\right]\left(\frac{1 + \chi}{\chi}\right) + 2\left(1 - \chi\right)$$

$$C_X = \frac{1}{2}\left[\left(\frac{1}{\cos\beta_1}\right)^2 - \left(\frac{\chi}{\cos\beta_2}\right)^2\right]\left(\frac{1 + \chi}{\chi}\right) + 2\left(1 - \chi\right) - \frac{1}{2}\overline{\omega}\left(\frac{1}{\cos\beta_1}\right)^2\left(\frac{1 + \chi}{\chi}\right)$$

M.4 Tangential Momentum Equation

$$\sum F_y = \oint_A W_y \rho \vec{W} \cdot d\vec{A}$$

$$Y = W_{y,2}\left(\rho W_{x,2} A_2\right) - W_{y,1}\left(\rho W_{x,1} A_1\right)$$

From mass conservation, we know that $\rho W_{x,2} A_2 = \rho W_{x,1} A_1$. Therefore,

$$Y = \rho W_{x,1} A_1 \left(W_{y,2} - W_{y,1}\right)$$

$$Y = \rho W_{x,1} A_1 \left(W_{x,2} \tan\beta_2 - W_{x,1} \tan\beta_1\right)$$

$$Y = \rho W_{x,1}^2 A_1 \left(\chi \tan\beta_2 - \tan\beta_1\right)$$

Therefore, the tangential force coefficient is,

$$C_Y = \frac{Y}{\frac{1}{2}\rho W_{x,1}^2 \tau h_1} = 2\left(\chi \tan\beta_2 - \tan\beta_1\right)$$

M.5 Mean Cascade Angle

Substitute the expressions for C_X and C_Y into the equation for C_D,

$$-C_D = \left(\frac{W_{x,1}}{W_m}\right)^2 \frac{\tau}{b} \left(\begin{array}{l} \left[\frac{1}{2}\left[\left(\frac{1}{\cos\beta_1}\right)^2 - \left(\frac{\chi}{\cos\beta_2}\right)^2 \right] \left(\frac{1+\chi}{\chi}\right) + 2(1-\chi) - \frac{1}{2}\overline{\omega}\left(\frac{1}{\cos\beta_1}\right)^2 \left(\frac{1+\chi}{\chi}\right) \right] \cos\beta_m \\ + 2\left(\chi \tan\beta_2 - \tan\beta_1\right) \sin\beta_m \end{array} \right)$$

We will now choose the mean angle such that the drag coefficient is directly proportional to the loss coefficient. This requires that the following condition be satisfied,

$$\left[\frac{1}{2}\left[\left(\frac{1}{\cos\beta_1}\right)^2 - \left(\frac{\chi}{\cos\beta_2}\right)^2 \right] \left(\frac{1+\chi}{\chi}\right) + 2(1-\chi) \right] \cos\beta_m + 2\left(\chi \tan\beta_2 - \tan\beta_1\right) \sin\beta_m = 0$$

This can be solved for β_m to give the following expression,

$$\tan\beta_m = \frac{\frac{1}{4}\left[\left(\frac{1}{\cos\beta_1}\right)^2 - \left(\frac{\chi}{\cos\beta_2}\right)^2 \right] \left(\frac{1+\chi}{\chi}\right) + (1-\chi)}{\tan\beta_1 - \chi \tan\beta_2}$$

After a little algebra, we can show that for $\chi = 1$, the above expression reduces to the equation shown earlier for a 2–D cascade,

$$\tan\beta_m = \frac{1}{2}\left(\tan\beta_2 + \tan\beta_1\right)$$

M.6 Drag Coefficient

The condition above also results in the following expression,

$$-C_D = -\frac{1}{2}\varpi\frac{\tau}{b}\left(\frac{1}{\cos\beta_1}\right)^2 \cos\beta_m\left(\frac{W_{x,1}}{W_m}\right)^2\left(\frac{1+\chi}{\chi}\right)$$

Since $W_m = \dfrac{W_{x,1}}{\cos\beta_m}$, then,

$$C_D = \frac{1}{2}\varpi\frac{\tau}{b}\frac{\cos^3\beta_m}{\cos^2\beta_1}\left(\frac{1+\chi}{\chi}\right)$$

For $\chi = 1$, the above expression reduces to the equation shown earlier for a 2–D cascade,

$$C_D = \varpi\frac{\tau}{b}\frac{\cos^3\beta_m}{\cos^2\beta_1}$$

M.7 Lift Coefficient

$$C_Y = \left(\frac{W_m}{W_{x,1}}\right)^2\frac{b}{\tau}\left(-C_D\sin\beta_m - C_L\cos\beta_m\right)$$

$$-C_L = \frac{1}{\cos\beta_m}\left[\left(\frac{W_{x,1}}{W_m}\right)^2\frac{\tau}{b}C_Y + C_D\sin\beta_m\right]$$

Since $W_m = \dfrac{W_{x,1}}{\cos\beta_m}$, then,

$$-C_L = \cos\beta_m\frac{\tau}{b}C_Y + C_D\tan\beta_m$$

$$-C_L = 2\frac{\tau}{b}\left(\chi\tan\beta_2 - \tan\beta_1\right)\cos\beta_m + C_D\tan\beta_m$$

$$C_L = 2\frac{\tau}{b}\left(\tan\beta_1 - \chi\tan\beta_2\right)\cos\beta_m - C_D\tan\beta_m$$

which is the same as that for a 2–D cascade when $\chi = 1$. As noted in Reference 11 (p. 59), the drag contribution to the lift coefficient is very small; therefore, the $C_D\tan\beta_m$ term is neglected in Setup_BFaNS.

Appendix N.—Reynolds-Stress Transformation

N.1 Generalized Tensor Transformation

Any coordinate transformation can be written in the following form (Ref. 12, p. 31),

$$dy_i = \frac{\partial y_i}{\partial x_j} dx_j$$

where x_j is the original coordinate system, and y_i is the new coordinate system. If we limit ourselves to rectangular Cartesian coordinate system (which can only undergo rotations, reflections, and translations), then each term in the transformation matrix is constant, and we can integrate to get,

$$y_i = \beta_{ij} x_j + \alpha_i$$

The coefficients, β_{ij}, are the direction cosines between the new coordinate axes, y_i, and the old coordinate axes, x_j.

For any second-order tensor (such as Reynolds stress), the transformation from one coordinate system to another can be written as (Ref. 12, p. 33),

$$T'_{ij} = T_{mn} \frac{\partial y_i}{\partial x_m} \frac{\partial y_j}{\partial x_n}$$

For rectangular Cartesian coordinates, this transformation becomes,

$$T'_{ij} = T_{mn} \beta_{im} \beta_{jn}$$

where T_{ij} is the tensor in original coordinate system, and T'_{ij} is the tensor in the new coordinate system.

N.2 Transformation from Engine Coordinates to Duct Coordinates

Hanson derived his turbulent inflow noise theory for a rectilinear cascade (Ref. 4). He defined the duct coordinates such that x_d is in the axial direction, y_d is in the tangential direction ($y_d = r\theta$), and z_d is binormal to x_d and y_d. He assumed that the velocity was uniform and that the binormal component was zero.

To satisfy the assumption that the binormal component is zero, Setup_BFaNS defines the duct coordinates such that x_d is aligned with the meridional flow direction, y_d is in the tangential direction ($y_d = r\theta$), and z_d is binormal to x_d and y_d. If we denote the duct coordinates by (x_d, y_d, z_d) and the engine coordinates by (x_e, y_e, z_e), then the transformation from engine coordinates to duct coordinates is,

$$\begin{pmatrix} x_d \\ y_d \\ z_d \end{pmatrix} = \begin{pmatrix} \cos(90-\phi) & 0 & \sin(90-\phi) \\ 0 & 1 & 0 \\ -\sin(90-\phi) & 0 & \cos(90-\phi) \end{pmatrix} \begin{pmatrix} x_e \\ y_e \\ z_e \end{pmatrix}$$

where ϕ is the meridional flow angle. Therefore, the transformation matrix simply becomes,

$$\beta_{ij} = \begin{pmatrix} \sin\phi & 0 & \cos\phi \\ 0 & 1 & 0 \\ -\cos\phi & 0 & \sin\phi \end{pmatrix}$$

N.3 Reynolds Stress-Transformation

As noted earlier, the transformation for a second-order tensor is,

$$T'_{ij} = T_{mn}\beta_{im}\beta_{jn}$$

We will let T_{ij} represent the Reynolds stress ($\overline{c_i c_j}$). Since the Reynolds-stress tensor is symmetric, so we only need to consider the following six components,

$$\begin{aligned} T'_{11} = {} & T_{11}\beta_{11}\beta_{11} + T_{22}\beta_{12}\beta_{12} + T_{33}\beta_{13}\beta_{13} + T_{12} \\ & + T_{13}\left(\beta_{13}\beta_{11} + \beta_{11}\beta_{13}\right) + T_{23}\left(\beta_{13}\beta_{12} + \beta_{12}\beta_{13}\right) \end{aligned}$$

$$\begin{aligned} T'_{22} = {} & T_{11}\beta_{21}\beta_{21} + T_{22}\beta_{22}\beta_{22} + T_{33}\beta_{23}\beta_{23} + T_{12}\left(\beta_{22}\beta_{21} + \beta_{21}\beta_{22}\right) \\ & + T_{13}\left(\beta_{23}\beta_{21} + \beta_{21}\beta_{23}\right) + T_{23}\left(\beta_{23}\beta_{22} + \beta_{22}\beta_{23}\right) \end{aligned}$$

$$\begin{aligned} T'_{33} = {} & T_{11}\beta_{31}\beta_{31} + T_{22}\beta_{32}\beta_{32} + T_{33}\beta_{33}\beta_{33} + T_{12}\left(\beta_{32}\beta_{31} + \beta_{31}\beta_{32}\right) \\ & + T_{13}\left(\beta_{33}\beta_{31} + \beta_{31}\beta_{33}\right) + T_{23}\left(\beta_{33}\beta_{32} + \beta_{32}\beta_{33}\right) \end{aligned}$$

$$\begin{aligned} T'_{12} = {} & T_{11}\beta_{11}\beta_{21} + T_{22}\beta_{12}\beta_{22} + T_{33}\beta_{13}\beta_{23} + T_{12}\left(\beta_{12}\beta_{21} + \beta_{11}\beta_{22}\right) \\ & + T_{13}\left(\beta_{13}\beta_{21} + \beta_{11}\beta_{23}\right) + T_{23}\left(\beta_{13}\beta_{22} + \beta_{12}\beta_{23}\right) \end{aligned}$$

$$\begin{aligned} T'_{13} = {} & T_{11}\beta_{11}\beta_{31} + T_{22}\beta_{12}\beta_{32} + T_{33}\beta_{13}\beta_{33} + T_{12}\left(\beta_{12}\beta_{31} + \beta_{11}\beta_{32}\right) \\ & + T_{13}\left(\beta_{13}\beta_{31} + \beta_{11}\beta_{33}\right) + T_{23}\left(\beta_{13}\beta_{32} + \beta_{12}\beta_{33}\right) \end{aligned}$$

$$\begin{aligned} T'_{23} = {} & T_{11}\beta_{21}\beta_{31} + T_{22}\beta_{22}\beta_{32} + T_{33}\beta_{23}\beta_{33} + T_{12}\left(\beta_{22}\beta_{31} + \beta_{21}\beta_{32}\right) \\ & + T_{13}\left(\beta_{23}\beta_{j1} + \beta_{21}\beta_{33}\right) + T_{23}\left(\beta_{23}\beta_{32} + \beta_{22}\beta_{33}\right) \end{aligned}$$

For the conversion from (x_e, y_e, z_e) to (x_d, y_d, z_d), we get,

$$\begin{gathered} T'_{11} = T_{11}\sin^2\phi + T_{33}\cos^2\phi + 2T_{13}\sin\phi\cos\phi \\ T'_{22} = T_{22} \\ T'_{33} = T_{11}\cos^2\phi + T_{33}\sin^2\phi - 2T_{13}\sin\phi\cos\phi \end{gathered}$$

$$\begin{gathered} T'_{12} = T_{12}\sin\phi + T_{23}\cos\phi \\ T'_{13} = -T_{11}\sin\phi\cos\phi + T_{33}\sin\phi\cos\phi + T_{13}\left(-\cos^2\phi + \sin^2\phi\right) \\ T'_{23} = -T_{12}\cos\phi + T_{23}\sin\phi \end{gathered}$$

Appendix O.—Upwash Computation

In Reference 4, Hanson derived the transformation that converts from duct coordinates to cascade coordinates,

$$\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = \begin{pmatrix} \cos\left(\frac{\pi}{2}-\xi\right)\cos\psi_S + \sin\left(\frac{\pi}{2}-\xi\right)\sin\psi_L \sin\psi_S & \sin\left(\frac{\pi}{2}-\xi\right)\cos\psi_S - \cos\left(\frac{\pi}{2}-\xi\right)\sin\psi_L \sin\psi_S & -\cos\psi_L \sin\psi_S \\ -\sin\left(\frac{\pi}{2}-\xi\right)\cos\psi_L & \cos\left(\frac{\pi}{2}-\xi\right)\cos\psi_L & -\sin\psi_L \\ \cos\left(\frac{\pi}{2}-\xi\right)\sin\psi_S - \sin\left(\frac{\pi}{2}-\xi\right)\sin\psi_L \cos\psi_S & \sin\left(\frac{\pi}{2}-\xi\right)\sin\psi_S + \cos\left(\frac{\pi}{2}-\xi\right)\sin\psi_L \cos\psi_S & \cos\psi'_L \cos\psi_S \end{pmatrix} \begin{pmatrix} x_d \\ y_d \\ z_d \end{pmatrix}$$

where x_d is in the axial direction, y_d is in the tangential direction ($y_d = r\theta$), z_d is binormal to x_d and y_d. Due to the difference in stagger conventions between Hanson's derivation and Setup_BFaNS, Hanson's stagger angle has been replaced with the complement of ξ.

Hanson's derivation assumed that the binormal velocity component is zero. Therefore, in Hanson's derivation, the geometric lean and sweep are the same as the aerodynamic lean and sweep. However, Setup_BFaNS defines the duct coordinates such that x_d is aligned with the meridional flow direction rather than the axial direction. Thus, in Setup_BFaNS, the geometric lean and sweep are *not* the same as the aerodynamic lean and sweep. To account for this difference, we must use aerodynamic sweep and lean in the transformation matrix, i.e.,

$$\beta_{ij} = \begin{pmatrix} \cos\left(\frac{\pi}{2}-\xi\right)\cos\psi'_S + \sin\left(\frac{\pi}{2}-\xi\right)\sin\psi'_L \sin\psi'_S & \sin\left(\frac{\pi}{2}-\xi\right)\cos\psi'_S - \cos\left(\frac{\pi}{2}-\xi\right)\sin\psi'_L \sin\psi'_S & -\cos\psi'_L \sin\psi'_S \\ -\sin\left(\frac{\pi}{2}-\xi\right)\cos\psi'_L & \cos\left(\frac{\pi}{2}-\xi\right)\cos\psi'_L & -\sin\psi'_L \\ \cos\left(\frac{\pi}{2}-\xi\right)\sin\psi'_S - \sin\left(\frac{\pi}{2}-\xi\right)\sin\psi'_L \cos\psi'_S & \sin\left(\frac{\pi}{2}-\xi\right)\sin\psi'_S + \cos\left(\frac{\pi}{2}-\xi\right)\sin\psi'_L \cos\psi'_S & \cos\psi'_L \cos\psi'_S \end{pmatrix}$$

The derivation for aerodynamic sweep and lean is shown in Appendix P. The upwash is defined as the Reynolds-stress component that is normal to the airfoil surface (i.e., in the y_c direction). Therefore, the upwash can be determined from,

$$T''_{22} = T'_{mn}\beta_{2m}\beta_{2n}$$

where T'_{mn} is the Reynolds-stress tensor in the duct coordinate system (i.e., x_d, y_d, z_d).

Appendix P.—Aerodynamic Sweep and Lean

P.1 Coordinate Transformation

Hanson derived his turbulent inflow noise theory for a rectilinear cascade (Ref. 4). He defined the duct coordinates such that x_d is in the axial direction, y_d is in the tangential direction ($y_d = r\theta$), and z_d is binormal to x_d and y_d. He assumed that the velocity was uniform and that the binormal component was zero.

To satisfy the assumption that the binormal component is zero, Setup_BFaNS defines the duct coordinates such that x_d is aligned with the meridional flow direction, y_d is in the tangential direction ($y_d = r\theta$), and z_d is binormal to x_d and y_d. If we denote the duct coordinates by (x_d, y_d, z_d) and the engine coordinates by (x_e, y_e, z_e), then the transformation from engine coordinates to duct coordinates is,

$$\begin{pmatrix} x_d \\ y_d \\ z_d \end{pmatrix} = \begin{pmatrix} \cos(90-\phi) & 0 & \sin(90-\phi) \\ 0 & 1 & 0 \\ -\sin(90-\phi) & 0 & \cos(90-\phi) \end{pmatrix} \begin{pmatrix} x_e \\ y_e \\ z_e \end{pmatrix}$$

where ϕ is the meridional flow angle measured from the z_d axis. Therefore, the transformation matrix simply becomes,

$$\beta_{ij} = \begin{pmatrix} \sin\phi & 0 & \cos\phi \\ 0 & 1 & 0 \\ -\cos\phi & 0 & \sin\phi \end{pmatrix}$$

Thus, the relationship between duct coordinates and engine coordinates is,

$$\begin{aligned} x_d &= x_e \sin\phi + z_e \cos\phi \\ y_d &= y_e \\ z_d &= -x_e \cos\phi + z_e \sin\phi \end{aligned}$$

P.2 Geometric Sweep and Lean

The engine coordinates of the airfoil leading edge are defined parametrically as a function of z_e,

$$\begin{aligned} x_{e,le} &= x_{e,le}(z_e) \\ y_{e,le} &= y_{e,le}(z_e) \end{aligned}$$

From this parametric representation, we can determine the geometric sweep and lean via,

$$\tan\psi_S = \frac{dx_{e,le}}{dz_{e.le}}$$

$$\tan\psi_L = \frac{dy_{e,le}}{dz_{e,le}}$$

P.3 Aerodynamic Sweep and Lean

Aerodynamic sweep and aerodynamic lean are defined in the (x_d, y_d, z_d) coordinate system as,

$$\tan \psi'_S = \frac{dx_{d,le}}{dz_{d,le}}$$

$$\tan \psi'_L = \frac{dy_{d,le}}{dz_{d,le}}$$

From the coordinate transformation, we see that,

$$\tan \psi'_S = \frac{dx_{e,le}}{dz_{d,le}} \sin \phi + \frac{dz_{e,le}}{dz_{d,le}} \cos \phi$$

$$\tan \psi'_L = \frac{dy_{e,le}}{dz_{d,le}}$$

After applying the chain rule, we get,

$$\tan \psi'_S = \frac{\left(\frac{dx_{e,le}}{dz_{e,le}} \sin \phi + \cos \phi \right)}{\left(-\frac{dx_{e,le}}{dz_{e,le}} \cos \phi + \sin \phi \right)}$$

$$\tan \psi'_L = \frac{\left(\frac{dy_{e,le}}{dz_{e,le}} \right)}{\left(-\frac{dx_{e,le}}{dz_{e,le}} \cos \phi + \sin \phi \right)}$$

After substituting the definition of geometric sweep and lean, we get,

$$\tan \psi'_S = \frac{\tan \psi_S \sin \phi + \cos \phi}{-\tan \psi_S \cos \phi + \sin \phi}$$

$$\tan \psi'_L = \frac{\tan \psi_L}{-\tan \psi_S \cos \phi + \sin \phi}$$

After a little algebra, we finally get,

$$\psi'_S = 90 + \psi_S - \phi$$

$$\tan \psi'_L = \frac{\tan \psi_L}{-\tan \psi_S \cos \phi + \sin \phi}$$

Appendix Q.—Boundary-Layer Estimation

Q.1 Manually Specifying Boundary-Layer Thicknesses

The boundary-layer thicknesses at Stations 1 and 3 can be manually specified in the Setup_BFaNS input file. If the boundary layers are not manually specified, one of the following procedures will be used to estimate them.

Q.2 When the Flow Field is Determined from an Inviscid Flow-Field Prediction

For an inviscid flow-field prediction, Setup_BFaNS assumes a flat-plate turbulent boundary layer. The boundary-layer thickness is computed from,

$$R_L = \frac{U_\infty L}{\nu}$$

$$\delta = 0.37 L R_L^{-1/5} \qquad \text{(Ref. 13, p. 638)}$$

where U_∞ is the free-stream velocity *relative to the surface*, and L is the distance over which the boundary layer grows. The kinematic viscosity is hardcoded to 1.5×10^{-5} m^2/s (air at 20 °C—see Ref. 8). The free-stream velocity is estimated from the flow speed at the surface,

$$U_\infty = \sqrt{C_{r,surf}^2 + C_{z,surf}^2 + \left(C_{\theta,surf} - U_{surf}\right)^2}$$

where

$$U_{surf} = \text{rotational speed of the surface}$$

The distance over which the boundary layer grows is estimated from,

$$L = \frac{U_\infty}{C_{z,surf}}\left(z - z_0\right)$$

where

$$z_0 = \text{origin of the boundary layer}$$

The following table shows the parameters used for z_0 at each station.

Location	Value used for z_0
Station 1, Hub	z_{nose}
Station 1, Tip	z_{high}
Station 3, Hub	z_{core}
Station 3, Tip	z_{high}

Note that the boundary-layer origin for “Station 3, Tip” is assumed to be the hilite. This can be a gross approximation since the flow has passed through the fan.

Based on turbulence profiles shown in Reference 14, the average turbulence in a flat-plate turbulent boundary layer is approximately 5 percent of the free-stream velocity. For simplicity, Setup_BFaNS assumes a uniform 5 percent turbulence intensity across the boundary layer.

Q.3 When the Flow Field is Determined from a Viscous CFD Prediction

In classical boundary-layer theory, the boundary-layer thickness is defined as the point where the velocity is 99 percent of the free-stream velocity (Ref. 8, p. 388). Unfortunately, the velocity profiles in the fan stage are very non-uniform, making it difficult to determine a "free-stream" velocity.

A method was developed to estimate the boundary-layer thickness based on the circumferentially averaged turbulent kinetic energy profiles. Away from the endwalls, these profiles are much flatter than the velocity profiles, making them a better discriminator for determining the edge of the boundary layer. An example of this procedure is shown in Figure Q.1.

At each axial station, Setup_BFaNS divides the flow field into a "tip region" and a "hub region". The radial location that corresponds to the minimum turbulent kinetic energy divides these two regions. Setup_BFaNS computes the slope of the TKE profile in each region, and normalizes the slope such that its maximum value is one. Setup_BFaNS then searches for the point where the normalized slope is equal to a tolerance value, which is hardcoded to 0.1. This point is defined as the edge of the boundary layer. Via interpolation, Setup_BFaNS then computes the free-stream velocity (relative to the surface), the free-stream meridional velocity, and the turbulence level at the edge of the boundary layer.

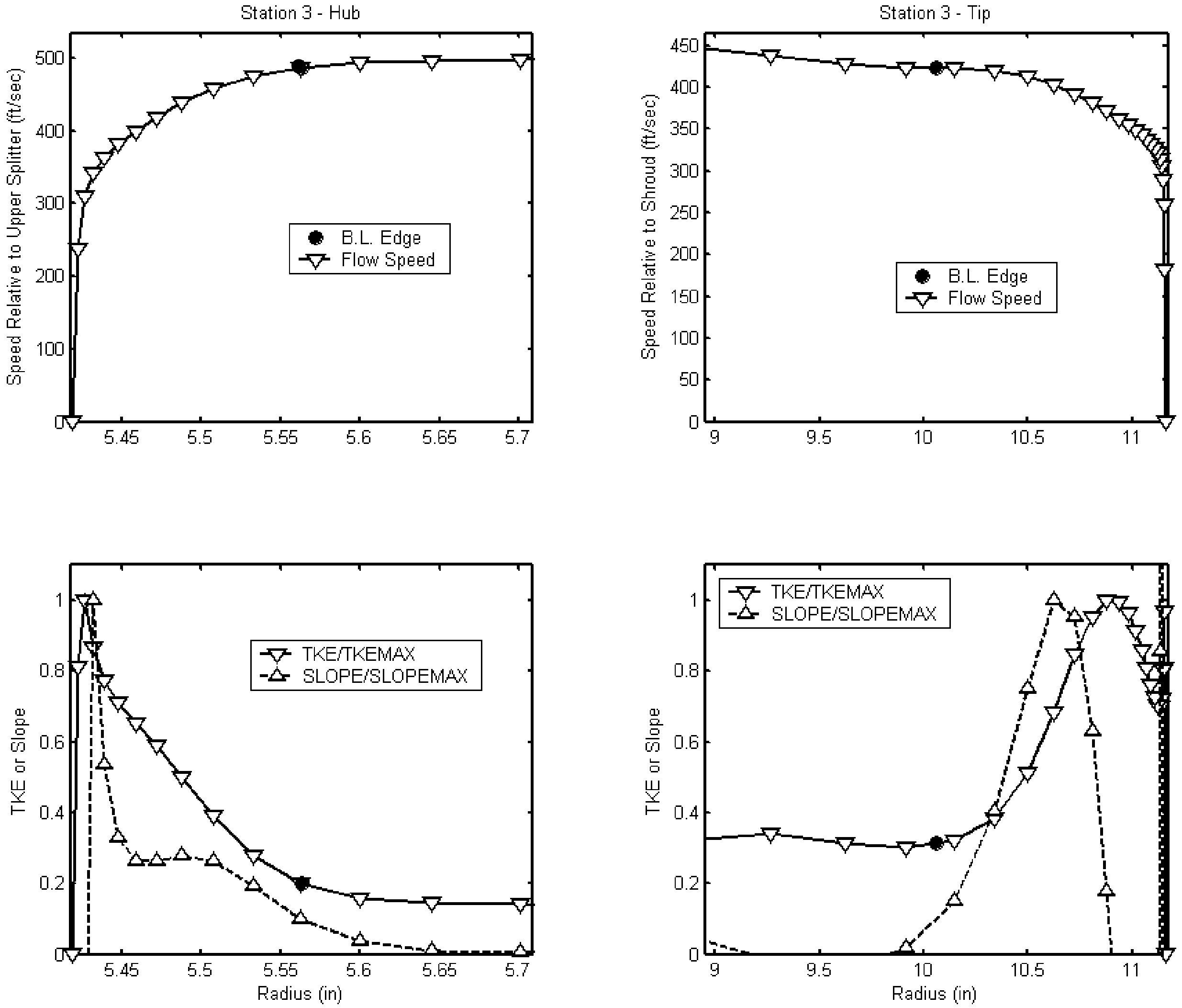


Figure Q.1.—Using K to determine edge of boundary layer.

Q.4 When the Flow Field is Determined from Experimental Data

For experimental data, Setup_BFaNS uses the same procedure that it uses for a viscous CFD prediction.

Q.5 Extrapolation to Blade and Vane Leading Edges

Once the boundary layers are determined (or manually specified) at Stations 1 and 3, they are extrapolated to the blade and vane leading edges. The extrapolation assumes a flat-plate turbulent boundary layer.

Appendix R.—Rotor-Wake Estimation

R.1 When the Flow Field is Determined From an Inviscid Flow-Field Prediction

For an inviscid flow-field prediction, Setup_BFaNS uses empirical correlations to estimate the rotor wake properties. In Reference 15, Glegg used the following correlations to estimate the wake velocity defect (ΔW), wake thickness (δ_w) and wake turbulence ($w'w'$),

$$\Delta W = \sqrt{\frac{K_0^2 \left|\vec{W}\right|^2 \delta_2}{s - K_1 \delta_2}}$$

$$\delta_w = 2\sqrt{K_2^2 \delta_2 \left(s - K_1 \delta_2\right)}$$

$$w'w' = K_3 \left(\Delta W\right)^2$$

These correlations are based on work by Wygnanski et al (Ref. 16). The constants depend on the shape of the body that generates the wake. Following Glegg’s example, Setup_BFaNS uses the constants that correspond to a symmetric airfoil, which are,

$$\begin{aligned} K_0 &= 1.56 \\ K_1 &= -380 \\ K_2 &= 0.32 \\ K_3 &= 0.15 \end{aligned}$$

Based on these correlations, the only information that is needed to compute the wake properties is the momentum thickness, the free-stream velocity, and the streamwise distance downstream from the airfoil trailing edge.

As shown in Section 4.9.4, the momentum thickness can be determined from,

$$\frac{\delta_2}{b} = 0.006 + 0.0002 e^{7.5 D_f - 1}$$

If the axial velocity changes between the blade trailing edge and Station 3, then the momentum thickness is adjusted according to,

$$\delta_2\big|_{Station\ 3} = \delta_2 \left(\frac{W_{z,te}}{W_{z,3}}\right)^2$$

The free-stream velocity is assumed to be the circumferentially averaged relative flow speed at Station 3. The streamwise spacing between the fan trailing edge and Station 3 is,

$$s = \frac{\left|\vec{W}_3\right|}{W_{z,3}} \left(z_3 - z_{te}\right)$$

R.2 When the Flow Field is Determined From a Viscous CFD Prediction

For viscous-CFD predictions, Setup_BFaNS determines the θ-locations where,

$$\frac{W - W_{\min}}{W_{\max} - W_{\min}} = 0.5$$

These locations define the edge of the wake. The wake thickness is defined as,

$$\delta_w = r\left(\theta^+ - \theta^-\right)$$

Figure R.1 demonstrates this process.

The wake turbulence is defined as the maximum value of $\sqrt{\frac{2}{3}K}$ that occurs in the blade passage. It is assumed to be uniform across the width of the wake.

R.3 When the Flow Field is Determined from Experimental Data

For experimental data, Setup_BFaNS uses the same procedure that it uses for a viscous CFD prediction.

R.4 When the Flow Field is Determined from Experimental Data

For experimental data, Setup_BFaNS uses the same procedure that it uses for a viscous CFD prediction.

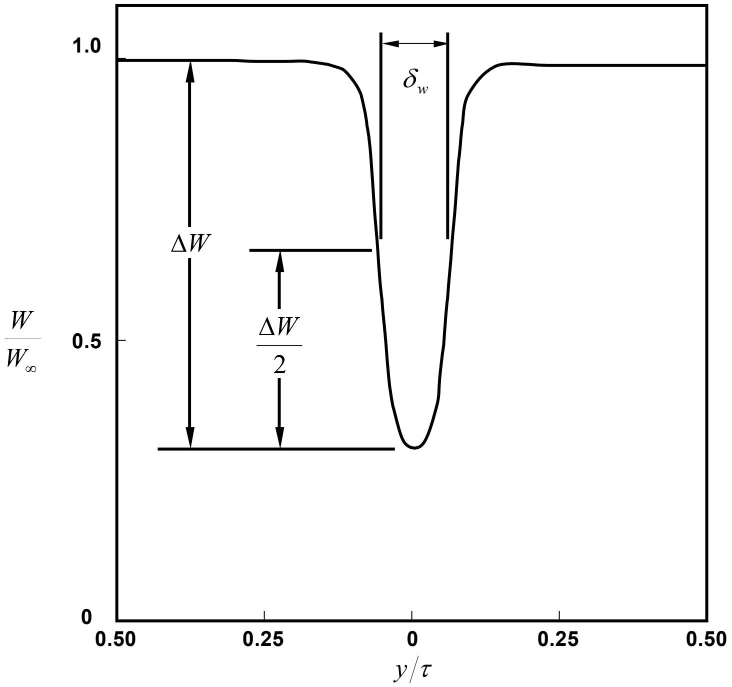


Figure R.1.—Wake terminology.

Appendix S.—Estimating Integral Length Scale

S.1 Integral Length Scale From Empirical Correlations

Endwall Boundary Layer.—In Reference 15, Glegg used a Von Karman type spectrum to model the boundary-layer turbulence interacting with the fan tip. In his model, Glegg used the following relationship,

$$k_e \delta = 1.2 \tag{S1}$$

where the turbulence wavenumber (k_e) is related to the integral length scale via,

$$k_e = \frac{\Gamma(5/6)}{\Gamma(1/3)} \frac{\sqrt{\pi}}{\Lambda} = \frac{0.747}{\Lambda} \tag{S2}$$

After substituting Equation (S2) into Equation (S1), we get the following relationship,

$$\Lambda = 0.62\delta$$

Following Glegg's example, we will assume that the integral length scale is constant across the layer, and is proportional to the thickness of the layer,

$$\Lambda = C\delta$$

The default value for the coefficient (C) is 0.62, consistent with the relationship that Glegg used. The user can over-ride this default with Card 5062 (Station no. 1, ID), Card 5064 (Station no. 1, OD), Card 5066 (Station no. 3, ID) and Card 5068 (Station no. 3, OD) in the author file.

Rotor Wake.—In Reference 15, Glegg also used a Von Karman type spectrum to model the wake turbulence interacting with the FEGV. In his model, Glegg used the following relationship,

$$k_e L_o = 1.81 \tag{S3}$$

where

$$L_o = \frac{\delta_w}{2}$$

After substituting Equation (S2) into Equation (S3), we get the following relationship,

$$\Lambda = 0.68\delta_w$$

Following Glegg's example, we will assume that the integral length scale is constant across the wake, and is proportional to the thickness of the wake,

$$\Lambda = C\delta_w$$

The default value for the coefficient (C) is 0.68, consistent with the relationship that Glegg used. The user can over-ride this default with Card 5070 in the author file.

S.2 Integral Length Scale From Turbulence Model

The integral length scale is assumed to be proportional to the turbulent mixing length,

$$\Lambda = C\frac{K^{\frac{3}{2}}}{\varepsilon}$$

The default value for the coefficient (*C*) is one. However, the user can over-ride this default with Card 5056 in the author file. At the walls, the turbulent length scale is hardcoded to be zero.

S.3 Circumferential Averaging

Results from Reference 17 showed that the Liepmann spectrum provided a good fit to the 1–D turbulence spectra measured downstream from a 22 in. fan rig. In addition, the results showed that the circumferentially averaged axial, tangential and radial turbulence components were nearly equal. Those observations suggest that the turbulence may be modeled with an isotropic spectrum model, even though the turbulence is non-homogenous.

We will assume that the Liepmann spectrum adequately represents the circumferentially averaged turbulence. Therefore, the 3–D upwash spectrum is (Ref. 4),

$$\phi_{22} = \frac{2}{\pi^2}\left(\frac{2}{3}K_{avg}\right)\Lambda^5_{avg}\frac{k_1^2 + k_3^2}{\left[1 + \Lambda^2_{avg}\left(k_1^2 + k_2^2 + k_3^2\right)\right]}$$

In this equation, K_{avg} and Λ_{avg} represent the turbulent kinetic energy and integral length scale associated with the circumferentially averaged turbulence.

Reference 17 describes a procedure for determining K_{avg} and Λ_{avg} from experimental data. When experimental measurements of the integral length scale are not available, Setup_BFaNS uses the procedure described below.

We will assume that the turbulence spectrum at each point can be represented by the Liepmann spectrum. With that assumption, the longitudinal turbulence spectrum at each point is (Ref. 18),

$$F(r,\theta,z,k_1) = \frac{2}{\pi}\left(\frac{2}{3}K\right)\frac{\Lambda}{1 + k_1^2\Lambda^2}$$

where *K* and Λ depend on (*r*, θ, *z*), and $k_1 = \omega/\left|\vec{C}\right|$. If we circumferentially average the longitudinal spectrum, we get,

$$F_{avg}(r,z,k_1) = \frac{2}{\pi}\frac{2}{3}\left[\frac{1}{2\pi}\int_0^{2\pi}\frac{K\Lambda}{1 + k_1^2\Lambda^2}d\theta\right]$$

At $k_1 = 0$, the average longitudinal spectrum evaluates to,

$$F_{avg}(r,z,0) = \frac{2}{\pi}\frac{2}{3}\left[\frac{1}{2\pi}\int_0^{2\pi}K\Lambda d\theta\right] \quad \text{(S4)}$$

Since we have assumed that the average spectrum can be adequately modeled with the Liepmann spectrum, then

$$F_{avg}(r,z,k_1) = \frac{2}{\pi}\left(\frac{2}{3}K_{avg}\right)\frac{\Lambda_{avg}}{1 + k_1^2\Lambda^2_{avg}}$$

At $k_1 = 0$, this becomes,

$$F_{avg}\left(r, z, 0\right) = \frac{2}{\pi}\frac{2}{3}K_{avg}\Lambda_{avg} \tag{S5}$$

After equating (S4) to (S5), we find that,

$$K_{avg}\Lambda_{avg} = \left[\frac{1}{2\pi}\int_0^{2\pi} K\Lambda d\theta\right]$$

Thus, the average length scale is,

$$\Lambda_{avg} = \left[\frac{1}{2\pi}\int_0^{2\pi}\left(\frac{K}{K_{avg}}\right)\Lambda d\theta\right]$$

where

$$K_{avg} = \left[\frac{1}{2\pi}\int_0^{2\pi} K d\theta\right]$$

References

1. Hanson, D.B., "Quantification of Inflow Turbulence for Prediction of Cascade Broadband Noise," Paper No. 990544, presented at the 5th International Congress on Sound and Vibration, Adelaide, Australia, Dec. 15–18, 1997.
2. Hanson, D.B., and Horan, K.P., "Turbulence/Cascade Interaction: Spectra of Inflow, Cascade Response, and Noise," Paper No. 98-2319, presented at the 4th AIAA/CEAS Aeroacoustics Conference, Toulouse, France, June 2–4, 1998.
3. Hanson, D.B., "Influence of Lean and Sweep On Noise of Cascades With Turbulent Inflow," Paper No. 99–1863, presented at the 5th AIAA/CEAS Aeroacoustics Conference, Seattle, WA, May 10–12, 1999.
4. Hanson, D.B., "Theory for Broadband Noise of Rotor and Stator Cascades With Inhomogeneous Inflow Turbulence Including Effects of Lean and Sweep," NASA/CR—2001-210762, May 2001.
5. Glegg, S.A.L., "Airfoil Self Noise Generated in a Cascade," AIAA Paper 96–1739, presented at the 2nd AIAA/CEAS Aeroacoustics Conference, State College, PA, May 6–8, 1996.
6. Glegg, S.A.L., and Jochault, C., "Broadband Self Noise from a Ducted Fan," presented at the 3rd AIAA/CEAS Aeroacoustics Conference, Atlanta, GA, May 12–14, 1997.
7. Glegg, S.A.L, and Jochault, C., "Fan Self Noise Prediction," Florida Atlantic University Report, April 1997.
8. White, F.M., *Fluid Mechanics*, 3rd Edition, McGraw-Hill, Inc, NY, 1994, pp. 512, 701.
9. Leiblein, S., Schwenk, F.C., and Broderick, R.L., "Diffusion Factor for Estimating Losses and Limiting Blade Loading in Axial-Flow-Compressor Blade Elements," NACA RM E53D01.
10. Leiblein, S., "Experimental Flow in Two-Dimensional Cascades," Chapter VI in *Aerodynamic Design of Axial-Flow Compressors*, NASA SP–36, 1965.
11. Dixon, S.L. *Fluid Mechanics, Thermodynamics of Turbomachinery*, 3rd Edition, Pergamon Press, NY, 1978 (reprinted in 1984).
12. McConnell, A.J., *Applications of Tensor Analysis*, Dover Publications, NY, 1957, p. 31, Eq. 22.
13. Schlichting, H., *Boundary-Layer Theory*, 7th Edition, McGraw-Hill Publishing Co., NY, 1987.
14. Hinze, J.O., *Turbulence*, 2nd Edition, McGraw-Hill, Inc, NY, 1987, pp. 245, 639.
15. Glegg, S.A.L., "Broadband Noise From Ducted Prop Fans," AIAA 93-4402, presented at the 15th AIAA Aeroacoustics Conference, Long Beach, CA, October 25–27, 1993.
16. Wygnanski, I., Champagne, F., and Marasli, B., "On the large-scale structures in two-dimensional, small-deficit, turbulent wakes," J. Fluid Mechanics, Vol. 168, 1986, pp. 31–71.
17. Morin, B.L. and Gilson, J., "Analysis of Wake Data for Predicting Broadband Noise from Fan Exit Guide Vanes," United Technologies Research Center, UTRC98–09, August 1999.
18. Liepmann H. W., "Extension of the Statistical Approach to Buffeting and Gust Response of Wings of Finite Span," Journal of the Aeronautical Sciences, March 1955.

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

1. REPORT DATE *(DD-MM-YYYY)* 01-11-2010	**2. REPORT TYPE** Final Contractor Report	**3. DATES COVERED** *(From - To)*

4. TITLE AND SUBTITLE
Broadband Fan Noise Prediction System for Turbofan Engines
Volume 1: Setup_BFaNS User’s Manual and Developer’s Guide

5a. CONTRACT NUMBER
NAS3-27727

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Morin, Bruce, L.

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
WBS 561581.02.08.03.18.03

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Pratt & Whitney
400 Main Street
East Hartford, CT 06108

8. PERFORMING ORGANIZATION REPORT NUMBER
E-17477-1

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING/MONITOR'S ACRONYM(S)
NASA

11. SPONSORING/MONITORING REPORT NUMBER
NASA/CR-2010-216898-VOL1

12. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified-Unlimited
Subject Categories: 71 and 02
Available electronically at http://gltrs.grc.nasa.gov
This publication is available from the NASA Center for AeroSpace Information, 443-757-5802

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Pratt & Whitney has developed a Broadband Fan Noise Prediction System (BFaNS) for turbofan engines. This system computes the noise generated by turbulence impinging on the leading edges of the fan and fan exit guide vane, and noise generated by boundary-layer turbulence passing over the fan trailing edge. BFaNS has been validated on three fan rigs that were tested during the NASA Advanced Subsonic Technology Program (AST). The predicted noise spectra agreed well with measured data. The predicted effects of fan speed, vane count, and vane sweep also agreed well with measurements. The noise prediction system consists of two computer programs: Setup_BFaNS and BFaNS. Setup_BFaNS converts user-specified geometry and flow-field information into a BFaNS input file. From this input file, BFaNS computes the inlet and aft broadband sound power spectra generated by the fan and FEGV. The output file from BFaNS contains the inlet, aft and total sound power spectra from each noise source. This report is the first volume of a three-volume set documenting the Broadband Fan Noise Prediction System: Volume 1: Setup_BFaNS User’s Manual and Developer’s Guide; Volume 2: BFaNS User’s Manual and Developer’s Guide; and Volume 3: Validation and Test Cases. The present volume begins with an overview of the Broadband Fan Noise Prediction System, followed by step-by-step instructions for installing and running Setup_BFaNS. It concludes with technical documentation of the Setup_BFaNS computer program.

15. SUBJECT TERMS
Fan blades; Vanes; Broadband; Turbulence; Rotor stator interactions

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** STI Help Desk (email:help@sti.nasa.gov)
a. REPORT U	**b. ABSTRACT** U	**c. THIS PAGE** U	UU	90	**19b. TELEPHONE NUMBER** ***(include area code)*** 443-757-5802