

General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.
- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.
- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.
- This document is paginated as submitted by the original source.
- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

(NASA-TM-X-73187) NASA AMES
THREE-DIMENSIONAL POTENTIAL FLOW ANALYSES
SYSTEM (POTFAN) BOUNDARY CONDITION CODE
(BCDN), VERSION 1 (NASA) 53 P HC A04/MF A01
CSCL 01A G3/02

N77-14995

Unclas
11506

**NASA TECHNICAL
MEMORANDUM**

NASA TM X-73,187

NASA TM X-73,187

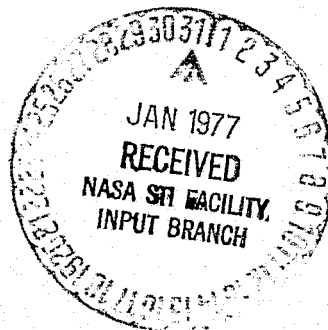
**NASA AMES THREE-DIMENSIONAL POTENTIAL FLOW ANALYSIS
SYSTEM (POTFAN) BOUNDARY CONDITION CODE
(BCDN) VERSION 1**

**J. E. Davis
Computer Sciences Corporation
Mountain View, California 94042**

and

**R. T. Medan
Ames Research Center
Moffett Field, California 94035**

December 1976



1. Report No. NASA TM X-73,187	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle NASA AMES THREE-DIMENSIONAL POTENTIAL FLOW ANALYSIS SYSTEM (POTFAN) BOUNDARY CONDITION CODE (BCDN) VERSION 1		5. Report Date	
		6. Performing Organization Code A-6844	
7. Author(s) J. E. Davis* and R. T. Medan**		8. Performing Organization Report No.	
		10. Work Unit No. 505-06-11	
9. Performing Organization Name and Address *Computer Sciences Corporation, Mountain View, Calif. 94042 **Ames Research Center, Moffett Field, Calif. 94035		11. Contract or Grant No.	
		13. Type of Report and Period Covered Technical Memorandum	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D. C. 20546		14. Sponsoring Agency Code	
		15. Supplementary Notes	
16. Abstract <p>This document describes a computer program known as BCDN which has been developed as an independent segment of the NASA-Ames Three-Dimensional Potential Flow Analysis System (POTFAN). This segment of the POTFAN system is used to generate right hand sides (boundary conditions) of the system of equations associated with the flow field under consideration. These specified flow boundary conditions are encountered in the oblique derivative boundary value problem (boundary value problem of the third kind) and contain the Neumann boundary condition as a special case. Arbitrary angle of attack and/or sideslip and/or rotation rates may be specified, as well as an arbitrary, nonuniform external flow field and the influence of prescribed singularity distributions.</p>			
17. Key Words (Suggested by Author(s)) Potential flow; Incompressible flow; Laplace's equation; Computer program; Computer analysis		18. Distribution Statement Unlimited STAR Category - 02	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 54	22. Price* \$4.25

NASA AMES THREE-DIMENSIONAL POTENTIAL FLOW
ANALYSIS SYSTEM (POTFAN)
BOUNDARY CONDITION CODE (BCDN) VERSION 1

J. E. Davis
Computer Sciences Corporation

and

R. T. Medan
Ames Research Center

SUMMARY

This document describes a computer program known as BCDN which has been developed as an independent segment of the NASA-Ames Three-Dimensional Potential Flow Analysis System (POTFAN). This segment of the POTFAN system is used to generate right hand sides (boundary conditions) of the system of equations associated with the flow field under consideration. These specified flow boundary conditions are encountered in the oblique derivative boundary value problem (boundary value problem of the third kind) and contain the Neumann boundary condition as a special case. Arbitrary angle of attack and/or sideslip and/or rotation rates may be specified, as well as an arbitrary, nonuniform external flow field and the influence of prescribed singularity distributions.

1 INTRODUCTION

This document describes Version 1 of a computer code (BCDN) which is a segment of the NASA-Ames Three-Dimensional Potential Flow Analysis System (POTFAN). This segment of the system computes the right hand side of the system of equations that represent the discretization of the boundary condition of a specified flow on all aircraft component surfaces in the flow field under consideration. This specified flow boundary condition is more general in nature than the Neumann boundary condition that is usually satisfied in the analysis of potential flow fields. In particular, the boundary condition in the BCDN code allows for the specification of the flow at any angle to the aircraft component surfaces and not necessarily normal to them. The boundary value problem that encompasses this boundary condition is referred to as the regular oblique derivative boundary value problem (boundary value problem of the third kind). The Neumann boundary value problem (boundary value problem of the second kind) is a special case of the oblique derivative boundary value problem.

The right hand side of the system of equations associated with this boundary condition represents the influence of the uniform and nonuniform freestream velocity, arbitrary rotation rates, and prescribed singularity distributions.

The prescribed singularity influence matrices, prescribed solution vectors, and geometric description of the surfaces of the aircraft components in the flow field are computed by other programs in the POTFAN system and are transmitted to the BCDN program as files through the auxiliary storage devices. The BCDN code reads in these files, performs its computations, and writes the results as files to be read in by the equation solving portion of the POTFAN system (SOLN).

2 PROBLEM TASK DESCRIPTION

This computer program was developed under Task 2 of NASA-Ames Contract NAS2-7571, and Task 26 of NASA-Ames Contract NAS2-6912. The purpose of these tasks was to develop a computer code which would independently compute the influence of the uniform and non-uniform freestream, rotation rates, and prescribed singularity distributions on the surfaces of an aircraft component. These influences are referred to as the right hand sides of the system of equations that are developed in discretely satisfying the boundary condition.

The boundary condition to be satisfied was to be more general than the Neumann boundary condition in that it was to allow for the flow at the aircraft component surfaces to be specified in any direction, not necessarily the direction normal to the surfaces.

Furthermore, the code was to allow for a top and/or bottom surface boundary condition as well as an arbitrary inward or outward specified flow at any of the boundary condition control points. In addition, the code was to have the capability for arbitrarily deleting boundary condition control points to reduce the number of equations that must be solved. This reduction is performed in those cases for which constraint functions are being employed. The constraint function concept is a method for relating the singularity strengths of panels so that they are not all independent and, therefore, the number of unknowns is reduced.

The program was also required to be modular in nature so that it could be used independently and so that any modifications or improvements to the code would not affect the other segments of the POTFAN system. The code was further required to be versatile, yet easy to use and modify.

3 METHOD OF SOLUTION

This section describes how the problems posed in the previous section were solved.

3.1 GENERAL SPECIFIED FLOW BOUNDARY CONDITION

In performing a potential flow analysis of a flow field, the surfaces of the bodies within the flow field are subdivided into quadrilateral finite elements called panels. Potential flow singularities are then distributed on these panels. The strengths of these singularities may either be known or unknown. For those singularities whose strengths are unknown, the boundary condition of a specified flow is satisfied at various locations on each body. The number and locations where these boundary conditions are satisfied are dependent on the type of singularity distribution and the type of constraint function applied to this distribution. The equations describing these boundary conditions form a set of linear, algebraic equations that are solved simultaneously to determine the strengths of the unknown singularities. For those singularities whose strengths are known directly or are directly calculated from known information or are calculated from approximate analytical theories, no boundary condition need be satisfied. Instead, the velocity induced by these singularities is evaluated as a part of the known flow field. Once all singularity strengths have been determined, the velocities induced by each of these singularities may be evaluated at various points on the body surfaces and in the flow field to determine either the loads on the body surfaces or the velocity at arbitrary points in space.

In mathematical form, the boundary condition of a specified flow at each boundary condition control point may be expressed as follows:

$$\underline{v} \cdot \underline{n} = v_{in} \quad (3.1-1)$$

where,

\underline{v} = The vector velocity induced by all influences in the flow field at the boundary condition control point. This includes known as well as unknown influences.

\underline{n} = Vector defining any arbitrary direction from the surface at the boundary control point. This vector need not be a unit vector. In fact, for any single boundary condition point, there may be two such vectors. This is to allow, for example, thin surfaces to be modelled by singularities on an infinitely thin mean plane. In the computer program, \underline{n} may be identified with the unit normals, or the NTOP vectors, or the NBOT vectors from the geometry file.

V_{in} = Specified flow through the surface in the \underline{n} direction at the boundary condition control point. For a solid surface, V_{in} would be zero. In the computer program V_{in} may be identified with the OTOP or OBOT values from the geometry file.

The velocity vector, \underline{v} , can be subdivided into two types of influences; those that are known or preset, and those that are unknown and must be determined. That is,

$$\underline{v} = \underline{v}_u + \underline{v}_k \quad (3.1-2)$$

\underline{v}_u = Vector velocity induced by all singularities.

\underline{v}_k = Vector velocity induced by the freestream linear and angular velocities and by all singularities whose strengths are known a priori.

Hence, the equation describing the boundary condition of a specified flow can be restated as,

$$\underline{v}_u \cdot \underline{n} = -\underline{v}_k \cdot \underline{n} + V_{in} \quad (3.1-3)$$

Satisfying equation (3.1-3) at each boundary condition control point of every aircraft component results in a set of linear, algebraic equations. This set of equations is then simultaneously solved to determine the unknown singularity strength distributions. In matrix notation, this set of equations can be expressed as:

$$Ax = b \quad (3.1-4)$$

where, A = influence matrix coefficients.

x = unknown singularity strength or constraint coefficient.

b = constant vector or right hand side vector.

The boundary condition program computes the right hand side of equation (3.1-4).

3.2 SYSTEM MODULARITY

The required degree of modularity, which implies that the program should operate independently from and without interfering with other portions of the POTFAN system, has been guaranteed by the use of auxiliary storage devices (tapes, disks, or drums) as the only method by which the various segments of POTFAN can communicate with one another. This has the disadvantage that communicating through auxiliary devices is relatively slow and on some computer systems will require many job control cards to manage the devices. However, the disadvantages of this approach are more than offset by the advantages. The principal advantage is that it strictly guarantees segment independence. Also this approach is necessary to maximize the size of problem that can be solved with a fixed amount of memory.

3.3 COMMAND FORMAT PROCESSING

Command format programming is a phrase coined to describe a programming method which uses words or acronyms (called commands) to control the actions taken by the program. Although this technique is not new, the particular style employed in POTFAN programs originated with the computer program reported in Medan (1973).

The manner in which command format programs work is the following. First of all there is a program known as the control program, which may be either the main program or the principal subroutine called by the main program. For the BCDN code the control program is the subroutine BCIO. The first two actions performed by the control program are the initialization of default values and the establishment of whether or not the program is being run in a batch or conversational mode. In the conversational mode the program prompts the user for commands and data and in the event of a recoverable error pauses to allow the user to perform a fixup. In the batch mode the program echoes each command

read in and stops upon detecting an error. Other than that, the two modes of operation are identical. Following these actions the program enters the command phase. In this phase the program reads in various four character commands, takes the action associated with the command, and then (provided the command is not STOP) reads in the next command and so on. The specific commands available are given in Section 5.2.

3.4 MAXIMIZING PROGRAM SIZE

The technique used for handling problems of varying size was dynamic memory allocation. This consists of establishing a single array (referred to as the working space array) that contains all the data that would vary with problem size. To facilitate usage of such an array, it is subdivided into smaller arrays when passed through the argument list to various subroutines and these smaller arrays are dimensioned only in the amount required to solve the problem. In this way, all of the available storage space is utilized. The working space array need only be dimensioned once in a small main program.

4 PROGRAM DESCRIPTION

To a great extent the description of the inner workings of the program has been relegated to comment cards in the FORTRAN source decks. This includes descriptions of the functions of the subroutines and their input and output. The remainder of the section presents relevant descriptive data which could not effectively be placed on comment cards.

Table 4-1 shows the subroutine calling structure.

Table 4-2 shows the common blocks used, their sizes, and the subprograms in which they appear.

Table 4-3 summarizes the logical units (tape, disks, or drums) which the program uses. Note that not all units would be used for each program run. For the worst case, the number of units used would be four. The specific data input on or output to each of these units except for the line printer unit is discussed in detail in Sections 5, 6, and 7.

Without the working storage array, the BCDN code requires approximately 13,000 decimal words of core storage. This requirement includes all system subroutines and internal symbol dictionaries and was determined on the INFONET UNIVAC 1108 operating system. The size of the working storage array must be added to this number to determine the total amount of storage required by the program. No overlay structure is employed in the BCDN code.

5 OPERATING INSTRUCTIONS

The purpose of this section is to provide the user with information necessary to execute the program.

5.1 GENERAL DATA INPUT CONSIDERATIONS

The input data for the boundary condition program consists of punched cards and tape, disk, or drum files when the code is utilized in the batch mode, and on-line data with tape, disk, or drum files when it is executed in the conversational mode. All punched card and conversational terminal input is prescribed in namelist or regular formats, whereas all bulk data input through tape, disk, or drum files is unformatted. The latter data is described in Sections 6 and 7.

The program is designed to use commands as the basic form of input to control the program flow. These commands consist of four letter words placed in the first four characters of an input record (first four columns of an input card) and are recognized as keys that cause the program to perform particular operations. After the operations are performed, the program flow returns to the beginning of the program and reads the next command. This continues until a STOP command is encountered, whereby the program terminates. Any command input record (card) whose first four characters or columns are left blank is considered a "comment" command. Any command that is not recognized by the program is printed and program flow is either returned to the next command without any operations being performed or terminated depending on whether or not the job is conversational or batch. The particular commands recognized by the boundary condition code are discussed in more detail in Section 5.2.

All of the commands available in the boundary condition code are related in some way to the computation of the right hand side of the system of equations associated with the specified flow boundary conditions, although several commands in sequence are required to complete the computation.

All geometric information about the aircraft components

under consideration is taken from geometry files. The preset singularity input information is taken from preset solution files and preset influence matrix files. The only other input information required are the boundary conditions (i.e., angle of attack and/or sideslip, rotation rates, inflows, center of gravity coordinates, and any nonuniform freestream velocities) and some miscellaneous parameters. The geometry information need only be input once; however, the boundary condition and preset singularity information must be input once for each set of boundary conditions.

5.2 INPUT DESCRIPTION

The first line of input to the program consists of a single logical variable. Its value depends on whether the command inputs are to be made utilizing the conversational or batch modes, true for conversational, false for batch mode. The format for this input record (card) is L1.

The remaining input for the boundary condition code consists of commands, and whatever data is required to effect the commands. A command statement must begin in the first column or with the first character of an input card or record. Only the first four characters of any command are recognized by the program. If the first four characters or columns are left blank, the command is assumed to be for comment purposes, and the entire card or line is printed. In the conversational mode, an unrecognizable command returns control to the next command after the unrecognizable characters are printed; otherwise, the program terminates. Data required to effect any command must immediately follow the command and precede the next command. A detailed description of the commands recognized by the code, as well as the input associated with them is given below. All logical unit numbers may range from 1 to 99, except 5 and 6. File identification numbers may range from 1 to 9999.

5.2.1 (4-blanks) Command

Explanation

Any command with blanks in the first four positions is considered a comment card. All information on this card or input record is printed, then control proceeds to the next command.

5.2.2 BCRE_Command

Explanation

This command causes the boundary condition description information to be read in. This command must be executed once for each set of boundary conditions to be satisfied for a given geometry (e.g., if two angles of attack are to be considered, then BCRE must be executed twice). A GREA command must precede the first BCRE command for each geometry to be considered.

Input

ALPHA--Angle of attack in degrees. The default value is zero.

BETA--Angle of sideslip in degrees. The default value is zero.

P--Roll rate about the center of gravity in radians per unit of time. The default value is zero.

Q--Pitch rate about the center of gravity in radians per unit of time. The default is zero.

RR--Yaw rate about the center of gravity in radians per unit of time. The default value is zero.

(RCG(3))--The x, y, and z coordinates of the center of gravity. The default values are zero.

IDEXT--Option flag whose value indicates whether the non-uniform freestream velocity vector at each boundary condition control point will be printed.

IDEXT=0--Not printed.

IDEXT=1--Printed.

The default value is zero.

TOP--Logical variable whose truth indicates that the boundary conditions are to be satisfied on the top surface. See Section 3.1 for a definition of the top surface. The default value is true. Input only once with the first set of boundary conditions for each geometry.

BOT--Logical variable whose truth indicates that the boundary conditions are to be satisfied on the bottom surface. See Section 3.1 for a definition of the bottom surface. The default value is the value of NBOT on the geometry file. Input BOT only with the first BCRE command

for each geometry.

NSETS--Number of sets of boundary conditions to be satisfied for the current geometry. The default value is one. Input only once with the first set of boundary conditions for each geometry.

(EXT(N1BC,N2BC,3))--Non-uniform velocity vector at each boundary condition control point of the current aircraft component. The default value is zero. If IDEXT=1, this array will be printed.

Format

Namelist BCREAD.

5.2.3 CBCV Command

Explanation

This command causes one right hand side vector to be computed. In addition, the velocity induced by the uniform and non-uniform freestream and rotation rate vector is also computed. In order to have the results of this computation printed and/or stored, the PBCV and/or SBCV commands must be executed. The CBCV command must follow the BCRE and PSRE commands for each set of boundary conditions.

Input

None required.

5.2.4 DATA Command

Explanation

The DATA command is used to initialize all input and output logical units. This command need not be used if the default values of the logical units are satisfactory. If used, this command must precede all other commands.

Input

NTCP--Logical unit number of the output device used for conversational mode operations. The default value is 6.

NTP--Logical unit number of the input device used for conversational mode operations. The default value is 6.

NTPC--Logical unit number of the input device used for conversational mode operations. The default value is 5.

NTR--Logical unit number of the input device used for batch mode operations. The default value is 5.

Format

Namelist MDATA.

5.2.5 GREA Command

Explanation

This command causes the geometry information concerning the aircraft component under consideration to be read from a geometry file. Only the information pertinent to the calculation of the right hand side vector is taken from the geometry file. This command must precede all other commands except the DATA and (4-blanks) command and must be input each time a new geometry is to be considered. In the event that multiple sets of boundary conditions are to be satisfied, the GRE command need only be entered before the first BCRE command and need not be entered for subsequent sets of boundary conditions.

Input

NTGR--The identification number of the geometry file.

NTG--The logical unit number of the geometry file.

Format

2I5.

5.2.6 PBCV Command

Explanation

This command causes the right hand side vector created by the uniform and nonuniform freestream, and rotation rates to be printed out. The PBCV command prints the results associated with only the most recent CBCV command, and may only be executed immediately following a CBCV or SBCV command.

Input

None.

5.2.7 PSRE Command

Explanation

This command causes the velocity influence on the boundary conditions induced by a prescribed singularity distribution to be computed. A preset solution and influence matrix are read from their respective files and multiplied together to determine this influence. A special option flag allows for the contributions of succeeding PSRE commands to be added together. The PSRE command follows the BCRE command and precedes the CBCV command. Unless a PSRE command is executed, the influence of any prescribed singularity distributions is assumed to be zero.

Input

NTPSR--Identification number of the preset solution file.

NTPS--Logical unit number of the preset solution file.

NTPVNR--Identification number of the influence matrix file.

NTPVN--Logical unit number of the influence matrix file.

MODEPS--A flag whose value indicates whether the influence of the current prescribed singularity distribution is to be added to the influences of the prescribed singularity distributions of the previous PSRE command.

MODEPS=1--The influence is added to the contribution of the previous prescribed singularity distribution

MODEPS=0--The influence is not added. It is assumed to be the only prescribed singularity distribution.

Format

2I5.

5.2.8 SBCV Command

Explanation

This command causes the right hand side vector and the boundary condition input information associated with the current set of boundary conditions to be stored on a boundary condition file. The SBCV command may only be executed immediately following a CBCV or PBCV command. The results of the successive sets of boundary conditions for the same aircraft component are placed on successive records of the boundary condition file.

Input

NTBCW--Identification number of the boundary condition file.

NTBC--Logical unit number for the boundary condition file.

NOTE--The above quantities should only be input for the first set of boundary conditions for each aircraft component. For subsequent SBCV commands there is no input. Thus NTBCW and NTBC should only be input as many times as there are GREA commands.

Format

2I5.

5.2.9 STOP Command

Explanation

This command terminates the program execution.

5.3 SYSTEM CONTROL CARDS

This section describes the control cards that are necessary to execute the BCDN code on the various computer systems that have or are being used to execute the BCDN code.

5.3.1 INFONET UNIVAC 1108 System.

Since this system allows automatic file definition, commands (see Appendix A), the only control card required is the program name, BCDN/POTF.

6 PROGRAM INPUT BINARY FILES

The BCDN code uses three types of files created by other POTFAN segments. These files contain the geometric information about the aircraft component on which the boundary condition is to be satisfied (geometry file), the prescribed singularity strengths of preset singularity distributions (preset solution file), and the influence matrix associated with the preset singularity distributions (influence matrix file). These files are accessed according to the procedures in Appendix A and Section 5.3. Each of these files conforms to the standardized POTFAN format, which is discussed in Appendix B. The notation used in the remainder of this section will be clear to the reader if he reads Appendix B. All of the data available on these files is not used by BCDN. The following subsections describe the files as well as the variables on the files that affect the operation of the BCDN code.

6.1 GEOMETRY FILES

Each geometry file contains all the information necessary to describe the geometry of the aircraft component on whose surface the boundary condition of a specified flow is to be satisfied. The file itself consists of 8 to 18 binary records. The actual number of records on the file will depend on the type of aircraft component analyzed as well as the approach used in performing the analysis. The first record is referred to as the introductory record, while all subsequent records contain various arrays of geometry related data.

Referring to the notation of Appendix B and the variables names in the BCDN code, the following information from a geometry file is used by the BCDN code.

1st Record

NCTIME--Number of words in the date and time array.

(CTIME(NCTIME))--Array containing the date and time of creation of the geometry file.

(TITL(NTITL))--Array containing the title information of the

geometry file.

NRECS--Number of records in the geometry file. sp;un 5
LOG(1)=BCFLG--Logical variable whose truth indicates that there are boundary condition flags stored on a subsequent record of the geometry file.

LOG(3)=UVW--Logical variable whose truth indicates that there are unit vectors in the direction of shed wake lines on a subsequent record of the file.

LOG(4)=DANDS--Logical variable whose truth indicates that there are both doublet and source singularities representing the body.

LOG(5)=DSF--Logical variable whose truth indicates that there are doublet singularity flags on a subsequent record of the geometry file.

LOG(6)=SSF--Logical variable whose truth indicates that there are source singularity flags on a subsequent record of the geometry file.

LOG(7)=NTOP--Logical variable whose truth indicates that there is an array of (RNTOP) values on a subsequent record of the geometry file.

LOG(8)=NBOT--Logical variable whose truth indicates that there is an array of (RNBOT) values on a subsequent record of the geometry file.

LOG(9)=OTOPL--Logical variable whose truth indicates that there is an array of (OTOP) values on a subsequent record of the geometry file.

LOG(10)=OBOTL--Logical variable whose truth indicates that there is an array of (OBOT) values on a subsequent record of the geometry file.

LOG(17)=DBLT--Logical variable whose truth indicates that the aircraft component is represented by doublet type singularities.

LOG(18)=SOURCE--Logical variable whose truth indicates that the aircraft component is represented by source-type singularities.

INT(2)=N1--Number of panel corner points in the S direction.

INT(3)=N2--Number of panel corner points in the V direction.

INT(4)=N1BC--Number of panels in the S direction.

INT(5)=N2BC--Number of panels in the V direction.

Subsequent Records

(BCFLAG(N1BC,N2BC))--Integer boundary condition flag denoting the type of boundary condition control point. A value of zero indicates a regular boundary condition control point. A value of one indicates a null boundary condition control point. Any other value indicates that the scalar influences of other components and other portions of the current component are not to be calculated or considered, but the velocity influences are to be calculated. This would be the case if constraint functions with deleted boundary conditions were to be used.

(RNTOP(N1BC,N2BC,3))--Array containing the direction vector associated with the top surface boundary condition.

(RNBOT(N1BC,N2BC,3))--Array containing the direction vector associated with the bottom surface boundary condition.

(VNTOP(N1BC,N2BC,3))--Desired value of net velocity along the (RNTOP) vectors on the bottom or inward side of the surface.

(VNBOT(N1BC,N2BC))--Desired value of net velocity along the (RNBOT) vector on the bottom or inward side of the surface.

6.2 PRESET SOLUTION FILES

A preset solution file contains all the information necessary to describe the singularity strength distribution representing an aircraft component whose influence on the flow field is prescribed before the boundary conditions are satisfied. The influence of this aircraft component is a part of the known flow field. The file itself contains three records. The first record is referred to as the introductory record. The second record is not used by BCDN. The third record contains the solution vector, i.e., the preset singularity strength.

Referring to the notation of Appendix B and the variable names in the BCDN code, the following information from a preset solution file is used by the BCDN code.

1st Record

NCTIME--Number of words in the date and time array.

(CTIME(NCTIME))--Array containing the creation date and time of the preset solution file.

NTITL--Number of words in the title information array.

NRECS--Number of records in the preset solution file.

NID--Number of file identification numbers on the introductory record.

NLOG--Number of logical variables on the introductory record.

NINT--Number of integer variables on the introductory record.

INT(2)=NW--Number of elements in the preset solution vector.

3rd Record

J1=NR--Number of elements in the preset solution vector.

(XPS(NR))--Array containing the elements of the preset solution vector.

6.3 INFLUENCE MATRIX FILES

Each influence matrix file contains the influences of singularities or constraint function terms of unit strength. The file itself consists of two or more records of information. The first record is referred to as the introductory record, while all subsequent records contain the influence coefficients. In many cases the influence matrix is very large and therefore resides on several records of information. Each record of influence coefficients is referred to as a matrix fragment.

Referring to the notation of Appendix B and the variable names in the BCDN code, the following information from the influence matrix file is used by the BCDN code.

1st Record

NCTIME--Number of words in the date and time array.

(CTIME(NCTIME))--Array containing the creation date and time of the preset influence matrix file.

INT(13)=NCFR--Number of influence matrix fragments.

INT(14)=NFC--Number of columns in the influence matrix.

INT(15)=NRPF--Maximum number of rows in any fragment of the influence matrix file.

2nd and Subsequent Records

J1=NFT--Number of rows of coefficients in the current fragment of the influence matrix.

NW=NRNC--Number of coefficients in the current fragment of the influence matrix.

(UVNPS(NRNC))--Influence coefficients on the current record.

7 PROGRAM OUTPUT

Output from the program consists of line printer output and boundary condition files. The line printer output is meant to be self-explanatory and will not be discussed further. The boundary condition files are created according to the procedure in Appendix A and conform to the format in Appendix B. Control cards for managing the boundary condition files are given in Section 5.3. The boundary condition files are discussed in more detail in the following subsections.

Each boundary condition file contains all the information necessary to describe the influence of the uniform and non-uniform freestream, the rotation rates, and the prescribed singularity distributions on the boundary conditions of a single aircraft component. The file itself consists of three or more records of information. The first record is referred to as the introductory record. Each group of two records thereafter describes all the flow field parameters associated with a single set of boundary conditions as well as their influence on this aircraft component. There may be several sets of boundary conditions on a single boundary condition file. The second, fourth, sixth, etc., records on the file contain the right hand side subvector associated with the system of equations describing the specified flow boundary condition. The third, fifth, seventh, etc., records contain the parameters defining the freestream conditions; that is, the freestream velocity vector, the angles of attack and sideslip, the rotation rates, and the center of gravity location for the flow field.

Referring to the notation of Appendix B and the variable names in the BCDN code, the following information is written on the boundary condition file.

1st Record

NCTIME--Number of words in the date and time array.

(CTIME(NCTIME))--Array containing the date and time that the solution file was created.

NTITL--Number of words in the title information array.

(TITL(NTITL))--Title of the boundary condition file.

NRECS--One plus two times the number of sets of boundary conditions.

(IFORM(NRECS))--0,1,1,...

NID--Number of identification numbers on the introductory record.

ID(1)-- Identification number of the geometry file associated with the aircraft component on whose surface the boundary conditions are to be satisfied.

ID(2)--Identification number of the boundary condition file.

NLOG--Number of logical variables on the introductory record.

LOG(1)=NTOP--Logical variable whose truth indicates that there is an array of (RNTOP) values on a record of the geometry file.

LOG(2)=NBOT--Logical variable whose truth indicates that there is an array of (RNBOT) values on a record of the geometry file.

LOG(3)=OTOPL--Logical variable whose truth indicates that there is an array of (VNTOP) values on a record of the geometry file.

LOG(4)=OBOTL--Logical variable whose truth indicates that there is an array of (VNBOT) values on a record of the geometry file.

LOG(5)=DBLT--Logical variable whose truth indicates that the aircraft component is represented by doublet-type singularities.

LOG(6)=SOURCE--Logical variable whose truth indicates that the aircraft component is represented by source-type singularities.

LOG(7)=TOP--Logical variable whose truth indicates that a top surface boundary condition was satisfied for this aircraft component.

LOG(8)=BOT--Logical variable whose truth indicates that a bottom surface boundary condition was satisfied for this aircraft component.

LOG(9)--Unused variable.

NINT--Number of integer variables on the introductory record.

INT(1)=IDEXT--Flag that indicates whether the arbitrary non-uniform freestream velocity at each boundary condition control point was specified and printed.

IDEXT=0--Not specified or printed.

IDEXT=1--Specified and printed.

INT(2)=IROW--Number of elements in the right hand side vector.

INT(3)=NSETS--Number of sets of boundary conditions.

NFLT=Number of floating point variables on the introductory record.

FLT(1)--Not used at present.

2nd, 4th, 6th, etc., Records

J1=IROW--Number of elements in the right hand side vector array.

J2=1

J3=1

NW=IROW--See above.

(BCC(NW))--Elements of a right hand side vector.

3rd, 5th, 7th, etc., Record

J1=11

J2=1

J3=1

NW=11

UINF, VINP, WINF--Components of the uniform freestream velocity vectory.

ALPHA--Angle of attack in degrees.

BETA--Angle of sideslip in degrees.

P,Q,RR--Components of the rotation rate vector.

(RCG(3))--Coordinate location of the center of gravity.

8 TEST CASES

This section describes test cases which were developed to test the program. These test cases are useful as an aid in familiarizing the user with the manner in which the BCDN code operates. They may also be used as a debugging aid when transferring the BCDN code to a different computer system.

Two test cases are included in this section. Two different geometries were used for these test cases. The first is the same as that used for sample case 4 in the documentation for the POTGEM Code Medan (1976). This case consists of a thin, flat half-wing with dihedral, sweepback, and taper, and at a large angle of attack, and represented by two chordwise and two spanwise panels. The second geometry is the same as that used for sample case 5 in the documentation of the POTGEM Code Medan (1976). This case consists of a hemisphere of radius one represented by five streamwise panels and ten circumferential panels.

8.1 TEST CASE NO. 1

The first test case evaluates the boundary condition on the top surface of the half-wing at two different angles of attack as well as one roll rate. The results are printed and stored on a boundary condition file. The input for this case is shown in Figure 8.1-1, while the output is shown in Figure 8.1-2.

8.2 TEST CASE NO. 2

The second test case evaluates the boundary conditions on the top surface of the half-wing for two sets of combined boundary conditions, then evaluates the boundary conditions on the hemisphere under the same two sets of boundary conditions. All results are printed and the results associated with the half-wing are stored on one boundary condition file while the results for the hemisphere are stored on a different boundary condition file. The input for this case is shown in Figure 8.2-1, while the output is

shown in Figure 8.2-2.

9 REFERENCES

Medan, R. T., and Ray, K. S. (1973). "Aerodynamic Influence Matrix Program for Aerodynamic Lifting surface Theory." NASA Report No. TMX-62,324.

Medan, R. T., and Bullock, R. B. (1976). "NASA Ames Potential Flow Analysis (POTFAN) Geometry Program (POTGEM) Version 1." NASA Report No. TM-X 73,127.

APPENDIX A

A--STANDARDIZED FILE HANDLING PROCEDURES FOR POTFAN PROGRAMS

Standardized FORTRAN procedures and subroutines for opening and closing files have been developed to facilitate using and coding POTFAN programs and the conversion of these codes to different computer systems.

A.1 FILE CREATION

This section describes actions taken before and after any POTFAN program attempts to write a POTFAN file.

Prior to writing any permanent file onto a unit, all POTFAN programs call a system dependent subroutine as follows:

```
CALL OPENW (NT,IFTYP,ID,IR)
```

If IR is not zero, then NT and ID are considered subroutine inputs. NT is the logical unit number on which the file will be written and ID is the file creation identifier, which should also be the primary file identification number. If IR is zero, then ID is not considered a subroutine input and NT is only the default unit number. In this case the program reads in ID and NT from a card via 2I5 format. If the value of NT on the card is zero, the subroutine replaces NT with the default value.

If the value of ID determined in either case is then still zero and if it is possible on the computer system being used, the program will replace ID with the current number on the identification number file and also update the identification number file.

In addition to NT, ID, and IR, IFTYP is also input to the program. IFTYP defines the type of file being created according to the following table:

<u>IFTYP</u>	<u>TYPE OF FILE</u>
1	Geometry
2	Boundary condition
3	Influence matrix
4	Velocity matrix
5	Solutions
6	Velocity at force sensing location of N1 segments
7	Velocity at force sensing location of N2 segments
8	Constraint function transformation matrix
9	Zeta plot file
10	Constrained influence matrix
12	Preset solution
15	Inverse or decomposition of influence matrix
16	External velocity
17	Surface pressures
18	Surface velocity

Once ID and NT have been determined, the program opens (if possible on the system being used) the file for writing using a file name determined from ID and IFTYP. On IBM systems, opening a file consists of issuing a DDEF to the operating system. On the INFONET UNIVAC 1108 system, an EQUATE command is involved. This feature eliminates the need for job control cards to handle files on those systems for which FORTRAN programs can open files.

The program then rewinds the file and writes a message indicating which unit has been opened and the value of ID and IFTYP.

After the file has been opened and written upon, it is released by calling another system dependent subroutine as follows:

CALL ENDFIL (NT)

This subroutine writes an end-of-file mark on the unit and (if required by the system being used), releases the unit. The subroutine also writes a message indicating that unit NT has been closed.

A.2 FILE ACCESSING

This section describes actions taken before and after any POTFAN program attempts to read any POTFAN file.

Prior to reading any permanent file from a unit all POTFAN programs call a system dependent subroutine as follows:

```
CALL OPENR(NT,IFTYP,ID,IR)
```

If IR is not zero, then NT and ID are considered subroutine inputs. NT is the logical unit number from which the file is read and ID is the file access identifier, which should also be the primary file identification number. If IR is zero, then ID is not considered a subroutine input and NT is only the default unit number. In this case, the program reads in ID and NT from a card via 2I5 format. If the value of NT on the card is zero, the subroutine replaces NT with the default value.

In addition to NT, ID, and IR, IFTYP is also input to the program. IFTYP defines the type of file being read according to the table in the previous section.

Once ID and NT have been determined, the program attempts to open the file using a file name determined from ID and IFTYP. The capability to open a file from a FORTRAN program depends on the system being used. As explained in the previous section, this may involve a DDEF or EQUATE command and can eliminate the need for job control cards to handle files.

The program rewinds the file and writes a message indicating which unit has been opened and the value of ID and IFTYP.

After control is returned to the calling program and the first record of the file has been read, all POTFAN programs check to see if the access identifier is equal to the actual primary file identification number existing on the first record. If not equal, the program writes an informational diagnostic message and proceeds. This feature is meant to be a helpful filekeeping technique for those systems that do not permit automated file control.

After the file has been read and there is no further use for it, it is released by calling another system

dependent subroutine as follows:

CALL FILEND (NT)

This subroutine rewinds unit NT and (if required by the system being used) releases the unit.

APPENDIX B

B STANDARDIZED FORMAT OF POTFAN FILES

A standard format has been developed for POTFAN files. This format is applicable to all files except scratch files and plot files. This standard has been developed for the following reasons:

1. to minimize the effects of changes in one POTFAN segment on other POTFAN segments;
2. to allow a program to be developed (EDITPF) which can list and/or edit the contents of any POTFAN file; and
3. to promote consistency among POTFAN programs.

Briefly, the standardized POTFAN file consist of one or more records. The first record is called the introductory record and contains miscellaneous data including the primary identification number, a title, and real, integer, and logical parameters reflecting how the data on the remaining records was calculated and/or how it is to be used. The second and subsequent records generally contain the bulk of the data and are called data records. The latter records contain one or more arrays which are always either integer or floating point numbers (i.e. integer and floating point numbers are not mixed on a single record). A detailed description is given below.

First Record (Introductory Record)

This record is created by an unformatted write statement such as the following:

```
WRITE(NT) NCTIME, (CTIME(N),N=1,NCTIME), NTITL,  
*(TITL(N),N=1,NTITL), NRECS, (IFORM(N),N=1,NRECS),  
#NID, (ID(N),N=1,NID), NLOG, (LOG(N),N=1,NLOG),  
#NINT, (INT(N),N=1,NINT), NFLT, (FLT(N),N=1,NFLT)
```

The values of NCTIME, NRECS, NID, NLOG, NINT, and NFLT are all at least one and can vary from file to file even for files of the same type (e.g. NINT may be different on two different geometry files). An explanation of these variables is given below:

NCTIME Number of words in (CTIME)

(CTIME) Creation time in A4 alphanumeric format. Whether or not this array can be filled out depends on the availability of a system dependent subroutine to compute it. This array is used only as a filekeeping aid. It is printed out whenever a file is created or read.

NTITL The number of words in (TITL). Generally NTITL is a multiple of 20.

(TITL) Alphanumeric titling information (e.g. "Delta wing with flaps"). This array is to be written under a format such as (1x,20A4/).

NRECS The number of records (including the first) comprising the file. NRECS is also the number of words in (IFORM).

(IFORM) An integer array indicating the kind of numbers on each record. A value of zero implies an integer and a value of one implies a floating point number. IFORM(1) has no significance.

NID The number of words in (ID)

(ID) Identification number array. ID(NID) is the primary file identification number. In order to keep track of files ID(NID) should be unique for each file. This number is printed out whenever the file is created or read.

NLOG The number of words in (LOG)

(LOG) An array of logical parameters

NINT Number of words in (INT)

(INT) An array of integer parameters

NFLT Number of words in (FLT)

(FLT) An array of floating point parameters. If the remaining data on the file is dependent on Mach number, then FLT(1) is the Mach number.

Second and Subsequent Records (Data Records)

The remaining records of POTFAN files contain one or more arrays. If the data record contains more than one array, then all arrays on the record must be of the same type (i.e. either integers or real numbers, but not both) and all arrays must have the same number of words. The records also contain array dimensions (J1, J2, and J3) and the total number of words in all arrays on the record (NW). Following are some examples of code used to create data records:

```
NW = J1*J2*J3
WRITE(NT) J1,J2,J3,NW,(((A(I,J,K),I=1,J1),J=1,J2),K=1,J3)
```

```
J3 = 2
NW = J1*J2*J3
WRITE(NT) J1,J2,J3,NW,((A(I,J),I=1,J1),J=1,J2),
#((B(I,J),I=1,J1),J=1,J2)
```

```
J2 = 1
J3 = 1
NW = J1
WRITE(NT) J1,J2,J3,NW,(A(I),I=1,NW)
```

```
J2 = 3
J3 = 1
NW = 3*J1
WRITE(NT) J1,J2,J3,NW,(A(I),I=1,J1),(B(I),I=1,J1),
#(C(I),I=1,J1)
```

Note that in the above examples all dimensions with multiple arrays were written with the leftmost indices varying most rapidly. This practice is always followed unless it is strictly necessary to do otherwise.

No matter how a data record was created, it can be read in by either of the following:

```
READ(NT) J1,J2,J3,NW,(A(I),I=1,NW)
READ(NT) J1,J2,J3,NW,(((A(I,J,K),I=1,J1),J=1,J2),K=1,J3)
```

In the former case, the data is packed solidly into core. In the latter case, some a priori knowledge of J1, J2, and

J3 or their maximum allowable values must have been available in order to properly dimension (A). Such a priori knowledge is generally contained as elements of (INT).

Different data records may contain data of different types and may have differing values of J1, J2, J3, and NW.

APPENDIX C

C ARRAY NOTATION

A shorthand notation for referring to arrays in the internal and external documentation of POTFAN programs has been developed. This notation is illustrated by the following examples:

- (A) This implies that A is an array.
- (A(N)) This refers to all the words in (A) from 1 through N.
- A(N) This refers only to the Nth word of (A).
- (A(I,J)) This refers to all the words in the doubly dimensioned array A for which the first index varies from 1 to I and the second from 1 to J.
- A(I,J) This refers to the element in (A) for which the first index is I and the second is J.
- (A(I,J),J=3,K) This refers to the words of (A) for which the first index is I and the second index varies from 3 to K.
- (A(I,*)) This refers to those elements of (A) for which the first index varies from 1 to I and the second index varies from 1 to some value which for some reason cannot be defined.

PROGRAM	CALLED BY	CALLS
BCDN	None	BCIO, TIMEST
BCIO	BCDN	CBC, OPENR, PBC, READBC, RELFIL, RGBC, RPS, SBC
CBC	BCIO	None
FILEND	SBC	None
OPENR	BCIO, RPS	None
OPENW	SBC	None
PBC	BCIO	None
READBC	BCIO	None
RELFIL	BCIO, RPS	None
RGBC	BCIO	None
RPS	BCIO	OPENR, RELFIL
SBC	BCIO	FILEND, OPENW, TIMEST
TIMEST	SBC, BCDN	None

TABLE 4-1. Subprogram Calling Structure

COMMON BLOCK NAME	SIZE	USING SUBPROGRAMS
BCCOM	141	BCIO, CBC, PBC, READBC, SBC
CONST	9	BCIO, BCDN, CBC, PBC, READBC, RGBC, RPS, SBC
GCOM	102	BCIO, CBC, RGBC, SBC
PSCOM	4	BCIO, RPS

TABLE 4-2. Common Block Usage

VARIABLE	LOGICAL UNIT DESCRIPTION	CURRENT VALUE
NTBC	Boundary Condition Output File	11
NTCP	Printed output device in conversational mode	6
NTPCR	Input device in conversational mode	5
NTG	Geometry input file	8
NTP	Printed output device in batch mode	6
NTPS	Preset colution input file	9
NTPVN	Preset influence matrix input file	10
NTR	Input device in batch mode	5

TABLE 4-3. Summary of Logical Units Used by BCDN Code

Figure 8.1-1. Input for BCDN test case 1

```
1 true
2 sample case 1 for the bcdn code.
3 three sets of boundary conditions.
4 results are printed and stored.
5 batch mode.
6 read the geometry file.
7 grea
8 4
9 define the first set of boundary conditions
10 as the default values.
11 bcre
12 $bcread nsets=3 $end
13 compute the right hand side and its associated
14 velocity for the first set of boundary conditions.
15 cbcv
16 print the results.
17 pbcv
18 store the results.
19 sbcv
20 1
21 define the second set of boundary conditions.
22 bcre
23 $bcread alpha=5. $end
24 compute the right hand side and its associated
25 velocity for the second set of boundary conditions.
26 cbcv
27 print the results.
28 pbcv
29 store the results on the next record of the same file.
30 sbcv
31 define the third set of boundary conditions.
32 bcre
33 $bcread alpha=0.,p=1. $end
34 compute the right hand side and its associated
35 velocity for the third set of boundary conditions.
```

```
36      cbcv
37      print the results.
38      pbcv
39      store the results on the next record of the same file.
40      sbcv
41      stop
```

Input for BCDN test case 1 (Concluded)

potfan boundary condition program. version 1.2
=====

dynamic memory = 5000 words

time = 11/29/75 13:35:49

+ sample case 1 for the bcdn code.
+ three sets of boundary conditions.
+ results are printed and stored.
+ batch mode.
+ read the geometry file.

+grea

ntgr= 4 ntg= 8

file 4.gm-pnc has been opened for reading on unit 8

creation time=10/08/75 13:08:00

unit 8 rewound and released

+ define the first set of boundary conditions
+ as the default values.

+bcre

boundary condition information

alpha= 0. beta= 0. p= 0. q= 0. rr= 0.

(rcg(i),i=1,3)= 0. 0. 0. idext= 0

+ compute the right hand side and its associated
+ velocity for the first set of boundary conditions.

+cbcv

+ print the results.

+pbcv

title = test case 4 -- thin, swept, uncambered, untwisted wing with dihedral

Figure 8.1-2. Output for BCDN test case 1

Figure 8.1-2. Output for BCDN test case 1 (continued)

```

reduced boundary condition constant vector
number of rows in vector= 4
(bcc(i),i=1,nrows)=
  .68599e+00 .68599e+00 .68599e+00 .68599e+00
+   store the results.
+sbcv
ntbcw= 1 ntbc= 0
file 1.bc-pnc/lib$      has been opened for writing on unit 11
creation time = 11/29/76 13:35:51
+   define the second set of boundary conditions.
+bcre

boundary condition information
alpha= .50000e+01 beta= 0.          p= 0.          a= 0.          rr= 0.
(rcg(i),i=1,3)= 0.          0.          0.          idext= 0
+   compute the right hand side and its associated
+   velocity for the second set of boundary conditions.
+cbcv
+   print the results.
+pbcv

title = test case 4 -- thin, swept, uncambered, untwisted wing with dihedral

reduced boundary condition constant vector
number of rows in vector= 4
(bcc(i),i=1,nrows)=
  .74317e+00 .74317e+00 .74317e+00 .74317e+00
+   store the results on the next record of the same file.
+sbcv
+   define the third set of boundary conditions.
+bcre

boundary condition information
alpha= 0.          beta= 0.          p= .10000e+01 a= 0.          rr= 0.

```

```

(rcg(i),i=1,3)= 0.          0.          0.          idext= 0
+   compute the right hand side and its associated
+   velocity for the third set of boundary conditions.
+cbcv
+   print the results.
+pbcv

title = test case 4 -- thin, swept, uncambered, untwisted wing with dihedral

reduced boundary condition constant vector
number of rows in vector= 4
(bcc(i),i=1,nrows)=
    .91645e+00 .16346e+01 .76638e+00 .15274e+01
+   store the results on the next record of the same file.
+sbcv
unit 11 endfiled and released
+stop
stop 777

```

Figure 8.1-2. Output for BCDN test case 1 (concluded)

Figure 8.2-1. Input for BCDN test case 2

```
1 true
2   sample case 2 for the bcdn code.
3   two aircraft components.
4   two sets of boundary conditions for each aircraft component.
5   results are printed and stored on two separate boundary condition files.
6   batch mode.
7   read the first geometry file.
8
9   grea
10  4
11  define the first set of boundary conditions for the first
12  aircraft component.
13  bcre
14  $bcread alpha=1.,beta=2.,nsets=2 $end
15  compute the results.
16  cbcv
17  print the results.
18  pbcv
19  store the results on the first boundary condition file.
20  sbcv
21  2
22  define the second set of boundary conditions for the first
23  aircraft component.
24  bcre
25  $bcread alpha=0.,beta=0.,p=1.,a=2.,rr=4. $end
26  compute the results.
27  cbcv
28  print the results.
29  pbcv
30  store the results on the next record of the same boundary
31  condition file.
32  sbcv
33  read the second geometry file.
34  grea
35  5
   define the first set of boundary conditions for the second
```


Figure 8.2-1. Input for BCDN test case 2 (concluded)

```
36     aircraft component.
37 bcre
38 $bcread alpha=1.,beta=2.,nsets=2,c=0.,rcg=0.,0.,0. $end
39     compute the results.
40 cbcv
41     print the results.
42 pbcv
43     store the results on the second boundary condition file.
44 sbcv
45     3
46     define the second set of boundary conditions for the
47     second aircraft component.
48 hcre
49 $bcread alpha=0.,beta=0.,p=1.,q=2.,rr=4. $end
50     compute the results.
51 cbcv
52     print the results.
53 pbcv
54     store the results on the next record of the second boundary
55     condition file.
56 sbcv
57     all done.
58 stop
```

potfan boundary condition program. version 1.2

=====

dynamic memory = 5000 words

time = 11/29/76 14:06:33

+ sample case 2 for the bcdn code.
+ two aircraft components.
+ two sets of boundary conditions for each aircraft component.
+ results are printed and stored on two separate boundary condition files.
+ batch mode.
+ read the first geometry file.

+grea

ntgr= 4 ntg= 8

file 4.gm-pnc has been opened for reading on unit 8

creation time=10/08/76 13:08:00

unit 8 rewound and released

+ define the first set of boundary conditions for the first
+ aircraft component.

+bcre

boundary condition information

alpha= .10000e+01 beta= .20000e+01 $\sigma= 0.$ rr= 0.

(rcg(i),i=1,3)= 0. 0. 0. idext= 0

+ compute the results.

+cbcv

+ print the results.

+pbcv

title = test case 4 -- thin, swept, uncambered, untwisted wing with dihedral

Figure 8.2-2. Output for BCDN test case 2

Figure 8.2-2. Output for BCDN test case 2 (continued)

```
reduced boundary condition constant vector
number of rows in vector= 4
(bcc(i),i=1,nrows)=
    .68897e+00 .68897e+00 .68897e+00 .68897e+00
+   store the results on the first boundary condition file.
+sbcv
ntbcw= 2 ntbc= 0
file 2.bc-pnc/lib$ has been opened for writing on unit 11
creation time = 11/29/76 14:06:36
+   define the second set of boundary conditions for the first
+   aircraft component.
+bcrc

boundary condition information
alpha= 0. beta= 0. p= .10000e+01 q= .20000e+01 rr= .40000e+01
(rcg(i),i=1,3)= 0. 0. 0. idext= 0
+   compute the results.
+cbcv
+   print the results.
+pbcv

title = test case 4 -- thin, swept, uncambered, untwisted wing with dihedral

reduced boundary condition constant vector
number of rows in vector= 4
(bcc(i),i=1,nrows)=
    -.25931e+01 -.49545e+01 -.50411e+01 -.67032e+01
+   store the results on the next record of the same boundary
+   condition file.
+sbcv
unit 11 endfiled and released
+   read the second geometry file.
+grea
ntgr= 5 ntg= 8
```

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

```

file 5.gm-pnc          has been opened for reading on unit  8
creation time=10/08/76  13:08:44
unit  8 rewound and released
+   define the first set of boundary conditions for the second
+   aircraft component.
+bcrc

boundary condition information
alpha= .10000e+01 beta= .20000e+01 p= .10000e+01 q= 0.          rr= .40000e+01
(rcg(i),i=1,3)= 0.          0.          0.          idext=  0
+   compute the results.
+cbcv
+   print the results.
+pbcv

title = test case 5 -- sphere with s the circumferential variable

reduced boundary condition constant vector
number of rows in vector= 50
(bcc(i),i=1,nrows)=
.96867e+00 .97156e+00 .97431e+00 .97675e+00 .97872e+00 .84570e+00 .85209e+00 .85805e+00 .86381e+00 .86811e+00
.64001e+00 .64847e+00 .65819e+00 .66666e+00 .67322e+00 .37169e+00 .38299e+00 .39389e+00 .40432e+00 .41192e+00
.66889e-01 .79166e-01 .90991e-01 .10131e+00 .10996e+00 -.24485e+00 -.23310e+00 -.22140e+00 -.21073e+00 -.20290e+00
-.53222e+00 -.52234e+00 -.51218e+00 -.50231e+00 -.49549e+00 -.76764e+00 -.75951e+00 -.75214e+00 -.74446e+00 -.73928e+00
-.92768e+00 -.92298e+00 -.91848e+00 -.91451e+00 -.91098e+00 -.99705e+00 -.99634e+00 -.99551e+00 -.99474e+00 -.99411e+00
+   store the results on the second boundary condition file.
+sbcv
ntbcw=  3 ntbc=  0
file 3.bc-pnc/lib$          has been opened for writing on unit 11
creation time = 11/29/76  14:06:40
+   define the second set of boundary conditions for the
+   second aircraft component.
+bcrc

```

```

boundary condition information
alpha= 0.          beta= 0.          n= .10000e+01  a= .20000e+01  rr= .40000e+01
(rcg(i),i=1,3)= 0.          0.          0.          idext= 0
+ compute the results.
+cbcv
+ print the results.
+pbcv

```

```

title = test case 5 -- sphere with s the circumferential variable

```

```

reduced boundary condition constant vector
number of rows in vector= 50
(bcc(i),i=1,nrows)=

```

```

.97202e+00 .97209e+00 .97208e+00 .97199e+00 .97203e+00 .85237e+00 .85236e+00 .85209e+00 .85216e+00 .85241e+00
.64919e+00 .64863e+00 .64924e+00 .64962e+00 .64961e+00 .38301e+00 .38330e+00 .38281e+00 .38330e+00 .38303e+00
.78760e-01 .79201e-01 .78855e-01 .78241e-01 .78428e-01 -.23375e+00 -.23365e+00 -.23363e+00 -.23343e+00 -.23400e+00
-.52275e+00 -.52324e+00 -.52334e+00 -.52254e+00 -.52309e+00 -.76089e+00 -.76048e+00 -.76087e+00 -.76012e+00 -.76040e+00
-.92394e+00 -.92383e+00 -.92392e+00 -.92406e+00 -.92395e+00 -.99711e+00 -.99731e+00 -.99739e+00 -.99742e+00 -.99738e+00

```

```

+ store the results on the next record of the second boundary
+ condition file.
+sbcv
unit 11 endfiled and released
+ all done.
+stop
stop 777

```

Figure 8.2-2. Output for BCDN test case 2 (concluded)