File with
N75-32027

41176

NASA CR-132710

A THREE-DIMENSIONAL SOLUTION OF FLOWS OVER WINGS WITH LEADING-EDGE VORTEX SEPARATION

PART II

Program Description Document

Ronald G. Coleman, Forrester T. Johnson, and Paul Lu

September 1975

Prepared under contracts NAS1-12185 and NAS1-13833 by
Boeing Commercial Airplane Company
P.O. Box 3707
Seattle, Washington 98124

for
Langley Research Center
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-132710		

4. Title and Subtitle
A THREE-DIMENSIONAL SOLUTION OF FLOWS OVER WINGS WITH LEADING-EDGE VORTEX SEPARATION
Part II—Program Description Document

5. Report Date
September 1975

6. Performing Organization Code

7. Author(s)
Ronald G. Coleman, Forrester T. Johnson, and Paul Lu

8. Performing Organization Report No.
D6-41789

9. Performing Organization Name and Address
Boeing Commercial Airplane Company
P.O. Box 3707
Seattle, Washington 98124

10. Work Unit No

11. Contract or Grant No.
NAS1-12185, NAS1-13833

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington D.C. 20546

13. Type of Report and Period Covered
Contractor Report
May 1973 — June 1975

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

A method of predicting forces, moments, and detailed surface pressures on thin, sharp-edged wings with leading-edge vortex separation in incompressible flow is presented. The method employs an inviscid, incompressible flow model in which the wing and the rolled-up vortex sheets are represented by piecewise continuous quadratic doublet sheet distributions. The Kutta condition is imposed on all wing edges.

Computed results were compared to experimental data and with the predictions of the leading-edge suction analogy for a selected number of wing planforms over a wide range of angle of attack. These comparisons show the method to be very promising, capable of producing not only force predictions but also accurate predictions of detailed surface pressure distributions, loads, and moments.

Experience with the present computer program, however, is limited, and operational limitations related to doublet panel spacing and panel density requirements, behavior at large planform breaks, convergence characteristics, etc., have yet to be extensively explored.

17. Key Words (Suggested by Author(s))
Doublet singularity
Incompressible
Iteration
Leading edge
Nonlinear
Panel method
Potential flow
Three-dimensional
Separation

18. Distribution Statement
Unclassified-Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	182	

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

CONTENTS

A THREE-DIMENSIONAL SOLUTION OF FLOWS OVER WINGS WITH LEADING-EDGE VORTEX SEPARATION
Part II—Program Description Document

Ronald G. Coleman, Forrester T. Johnson, and Paul Lu
Boeing Commercial Airplane Company

SUMMARY

A computer program has been developed for the solution of the subsonic, three-dimensional flow over wings with leading-edge vortex separation. The documentation is divided into two volumes, Part I and Part II.

This volume is Part II of the documentation containing the description of the computer program. It consists of three sections presenting the Program Logic, the Description of Subroutines, and the Program Listing.

INTRODUCTION

A computer program has been developed for the solution of the subsonic, three-dimensional flow over wings with leading-edge vortex separation (ref. 1). The program provides capabilities for calculating forces, moments, and detailed surface pressures on thin, sharp-edged wings of an arbitrary planform. The wing geometry is arbitrary in the sense that leading and trailing edges may be curved or kinked and the wing may have arbitrary camber and twist. The computer program includes a recently developed potential flow computational technique based on the advanced aerodynamic panel method (ref. 2). The numerical method employs an inviscid flow model in which the wing and the rolled-up vortex sheets are represented by piecewise continuous quadratic doublet sheet distribution. The Kutta condition is imposed along all wing edges. An iterative scheme is applied to find the strengths of the doublet distributions as well as shape and position of the free vortex sheet spirals satisfying the nonlinear boundary conditions of the flow problem.

The computer program is written in FORTRAN IV for a SCOPE 3.0 or KRONOS 2.1 operating system of the CDC 6400/6600 computer. The program uses overlay structure and eight disk files (including input and output files), and it requires approximately 120 000 (Octal) central memory words. This program has been designed with the primary objective of verifying new concepts and ideas.

The documentation of the program is divided into two parts:

Part I: Engineering Document

Part II: Program Description Document

The Engineering Document (bound separately) contains a detailed description of the theoretical method and, in particular, a thorough discussion of the following items: flow model as a nonlinear boundary value problem, geometry definition, numerical method, solution procedure, and verification of the method. A user's guide of the computer program is also included in the Engineering Document.

This volume, the Program Description Document, consists of three sections:

1: Program logic describing the basic program structure and listing the names of overlay programs and all subroutines. It includes descriptions and flowcharts of overlay programs, along with a discussion of file usages and common blocks.

2: Purpose, input and output, and a brief discussion of processing performed by the routine for all subroutines.

3: A complete program listing.

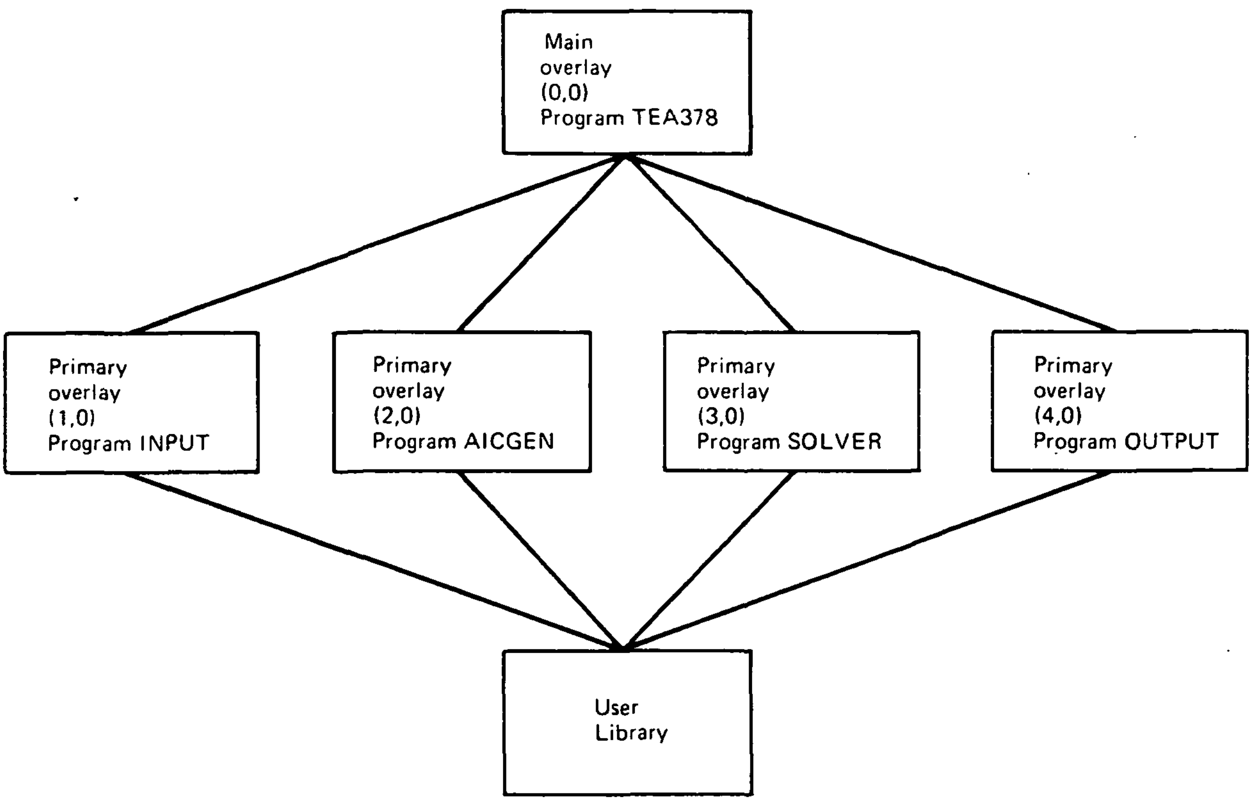
The Program Description Document is written for the purpose of maintaining an experimental-type computer program.

PROGRAM LOGIC

BASIC PROGRAM STRUCTURE

The computer program consists of one main overlay, four primary overlays, and one user library. A description and flowchart of each overlay program are given.

The flowchart of main overlay program TEA378 illustrates the overall functions the program performs. A schematic diagram of basic program structure follows:



NAMES OF PROGRAMS AND SUBROUTINES

The names of programs and subroutines used in each overlay and of those routines included in the user library are given as follows:

Main	OVERLAY(VORTEX,0,0)
Program	TEA378
Subroutines	ITFLOW, FUNC, FGCAL, UPDATE, AJGEN, DFGMU, DFGDT

Primary	OVERLAY(VORTEX,1,0)
Program	INPUT
Subroutines	DWNET, AWNET, GWNET, SWEPTE, SHEGEN

Primary	OVERLAY(VORTEX,2,0)
Program	AICGEN
Subroutines	EDGEIN, KSORT, TGEOMC, GEOMC, SURFIT, TSING, SING, LSQSF, TCNTRL, CONTRL, SURPRO, VINFCC, EIVC, PIVC, GCPCAL, GRDIND, PIDENT, CCAL, ECAL, DPIV, FKCAL, FMNCAL, FNKCAL, INTCAL, SIDECL

Primary	OVERLAY(VORTEX,3,0)
Program	SOLVER

Primary	OVERLAY(VORTEX,4,0)
Program	OUTPUT
Subroutines	SNGCAL, SINFCC

User Library Subroutines	IPTRNS, PTRNS, TRNSFR, TRANS, ZERO, CROSS, UVECT, MMULT, CMAB, LINEQS, TDECOM, BSUBSM, PDSEQS, VIP, VIPA, VIPS

DESCRIPTION AND FLOWCHART OF OVERLAY PROGRAMS

Main OVERLAY(VORTEX,0,0)

Program TEA378

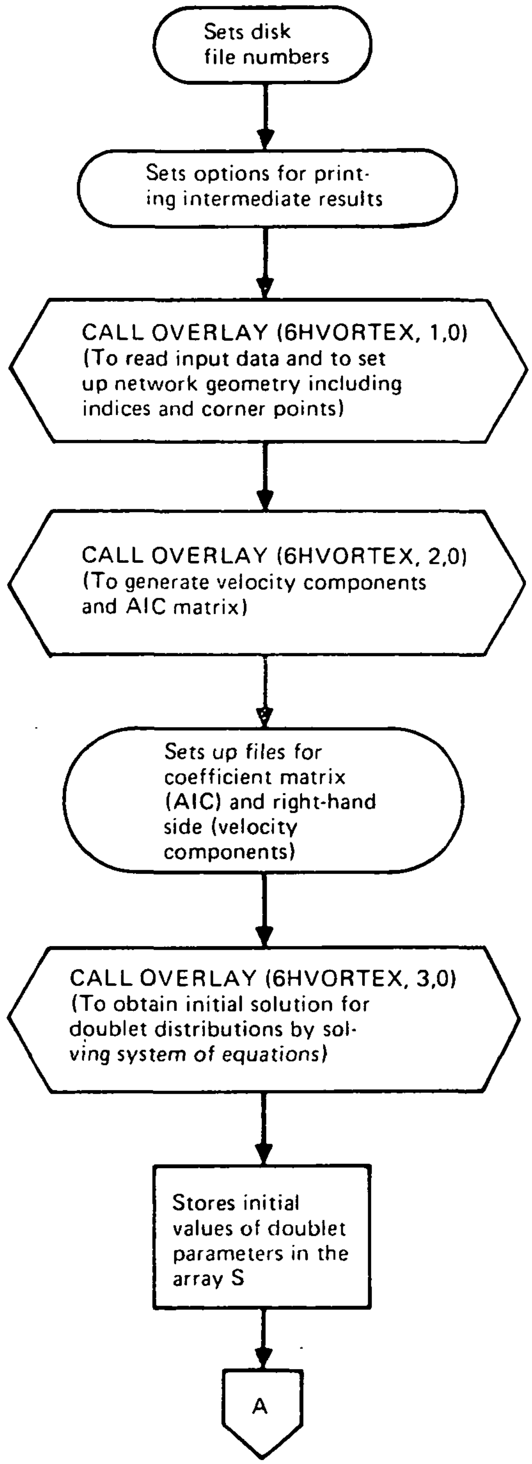
Purpose To call various overlays to perform the following tasks:

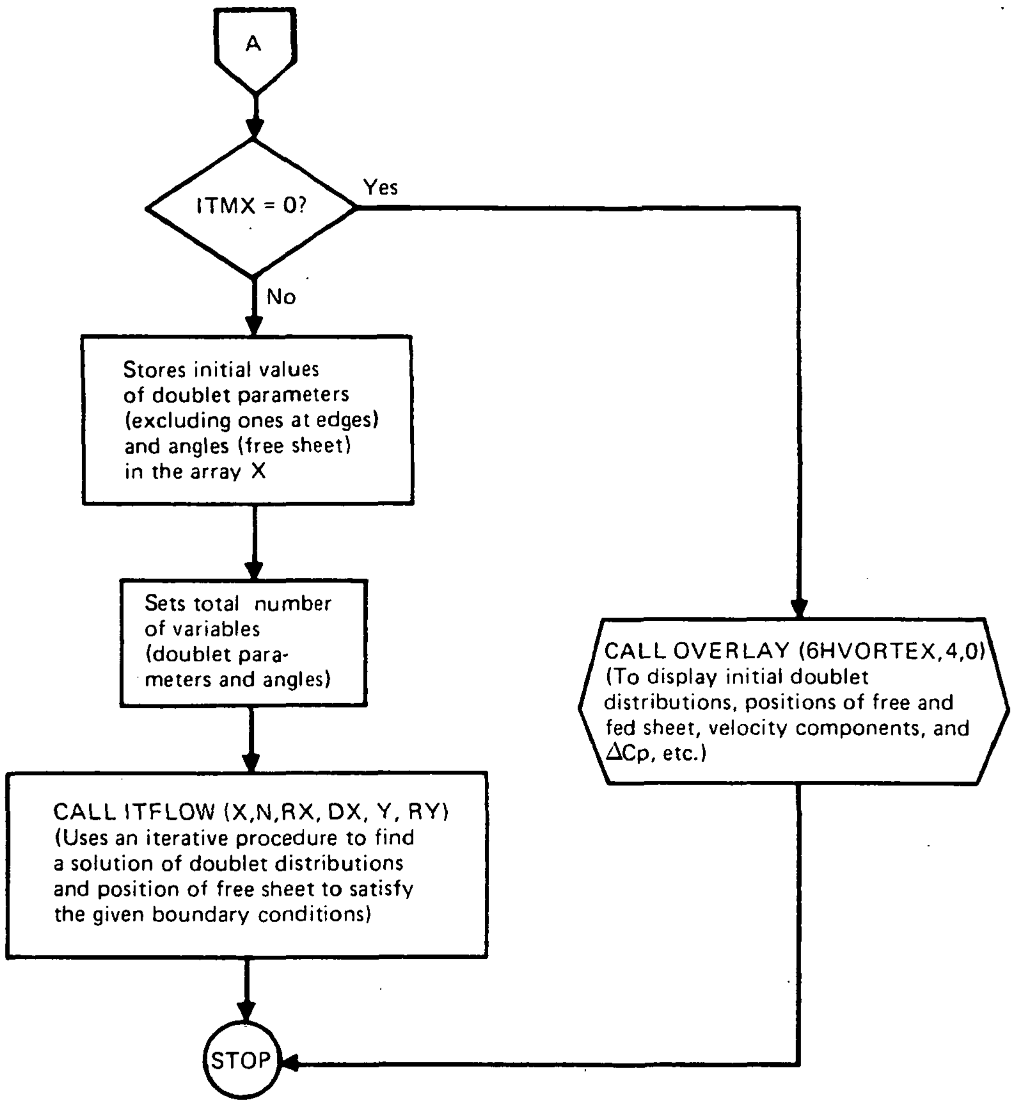
- Reading the input data and setting up geometry definition
- Generating the AIC matrix using an advanced panel-type method
- Solving a system of equations with the generated AIC to obtain initial doublet distributions
- Using the routine ITFLOW to find an iterative solution to the flow problem with nonlinear boundary conditions

Subroutines Called INPUT(OVERLAY-1,0), AICGEN(OVERLAY-2,0), SOLVER (OVERLAY-3,0), OUTPUT(OVERLAY-4,0), ITFLOW

Discussion Program TEA378 is the main overlay. It sets disk file numbers and options for printing intermediate results. The printing options are for checkout purposes. First the program calls INPUT to read input data cards and to set up network panel corner points, including an initial guess for the shape and position of free sheet. Then AICGEN is called to generate velocity components and AIC matrix using a panel-type influence coefficients method (see description of program AICGEN). An initial guess for doublet distributions is obtained by calling SOLVER to solve a system of equations with the generated velocity components and AIC matrix.

Flowchart of Program TEA378



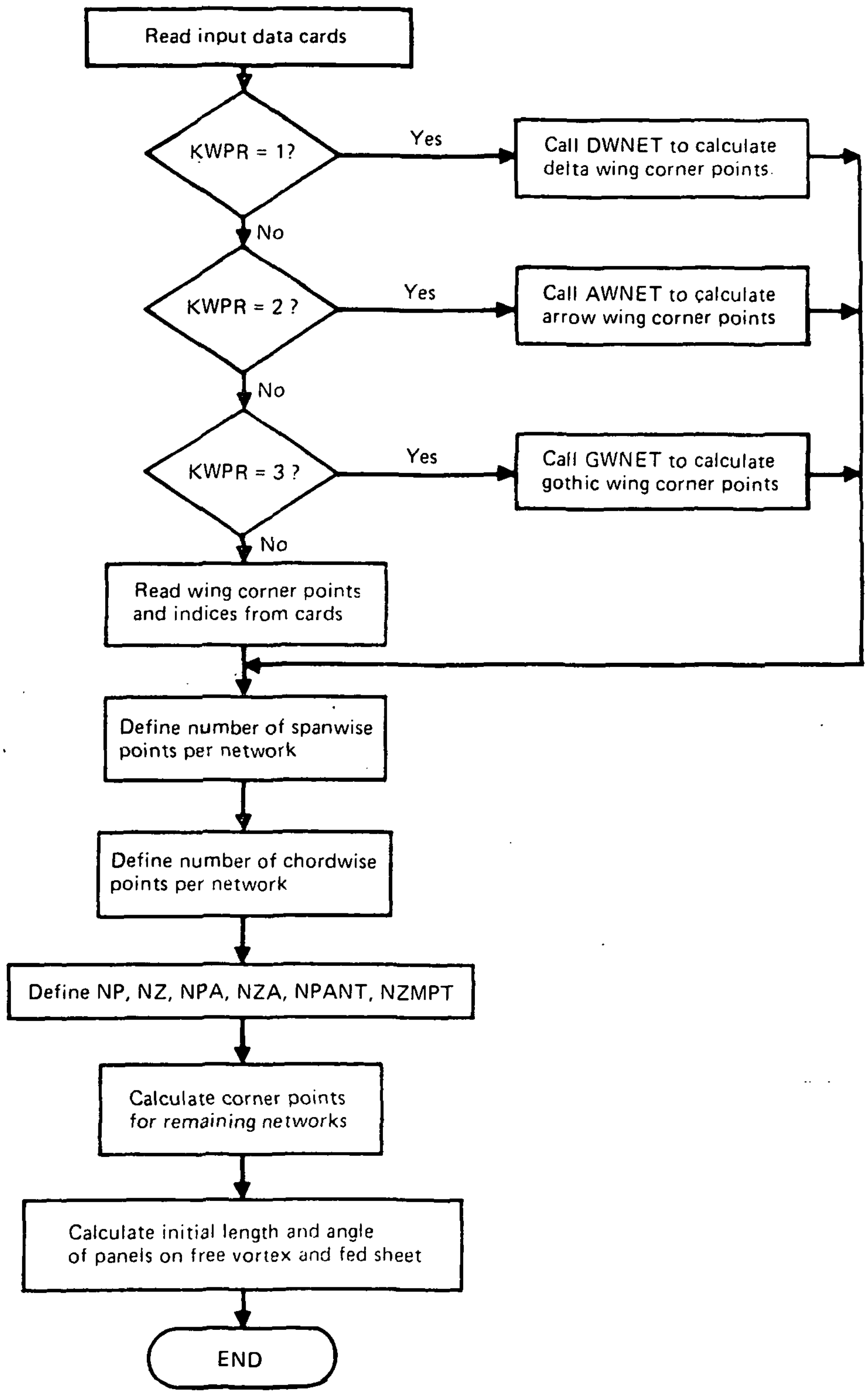

A
ITMX = 0?
Yes
No
Stores initial values of doublet parameters (excluding ones at edges) and angles (free sheet) in the array X
Sets total number of variables (doublet parameters and angles)
CALL ITFLOW (X,N,RX, DX, Y, RY) (Uses an iterative procedure to find a solution of doublet distributions and position of free sheet to satisfy the given boundary conditions)
CALL OVERLAY (6HVORTEX,4,0) (To display initial doublet distributions, positions of free and fed sheet, velocity components, and ΔCp, etc.)
STOP

If the user sets the number of iterations to zero in order to check the input data, initial guesses for doublet distributions, and position of free sheet, the program will call OUTPUT to display initial doublet distributions, positions of free sheet and fed sheet, velocity components, Δc_p, etc. Otherwise, the program proceeds to call the subroutine ITFLOW to find a solution for doublet distributions and the position of free sheet to satisfy the nonlinear boundary conditions.

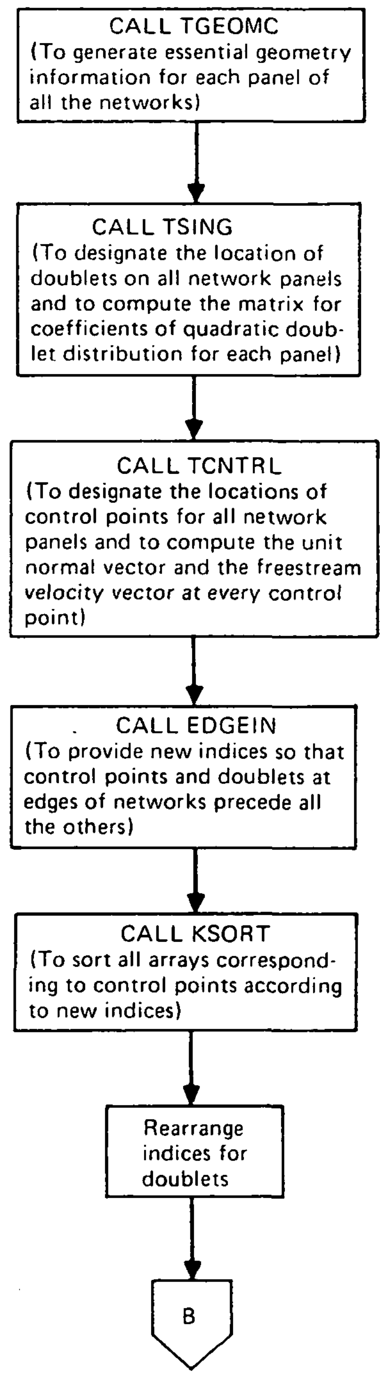
Primary	OVERLAY(VORTEX,1,0)
Program	INPUT
Purpose	To read and print user input data, calculate freestream velocity, calculate all panel corner point coordinates, calculate the initial length and angle of inclination of panels in the Free Vortex and Fed Sheet networks.
Input	See Engineering Document—User's Guide
Output	Common block /DAT3/-AR, NTR, XTR, MSP, YSP, NTC, NLE, YLE, NTE, YTE, MSF /FSVEL/-FSV, FSVM, ALPHA, XPITCH, RCHORD /INDEX/-NT, NM, NN, NP, NZ, NPA, NZA, NNETT, NPANT, NZMPT /MSPNTS/-ZM, ZL /SOLN/-ZA
Subroutines Called	AWNET, DWNET, GWNET, SHEGEN
Discussion	c is set equal to the cosine of the angle of attack in radians, and s is set equal to the sine. They form the components of a freestream velocity vector whose magnitude is calculated by taking the square root of the sum of the squares of c and s. The wing panel corner points may be input by the user following a "$INPUT WING NETWORK" data card or the program will calculate them following a "$DELTA," "$ARROW," or "$GOTHIC WING PREPROCESSOR" data card. Subroutine SHEGEN calculates the Y and Z coordinates of panel corner points on the Free Vortex and Fed Sheet networks. The distance between adjacent panel corner points on each transverse cut in the Free Vortex and Fed Sheet is calculated along with the panel inclination angle with respect to the flat wing.

Primary	OVERLAY(VORTEX,2,0)
Program	AICGEN
Purpose	To calculate essential geometry information for each panel and the locations of doublets and control points for each network and to generate the aerodynamic influence coefficients using an advance panel-type method.

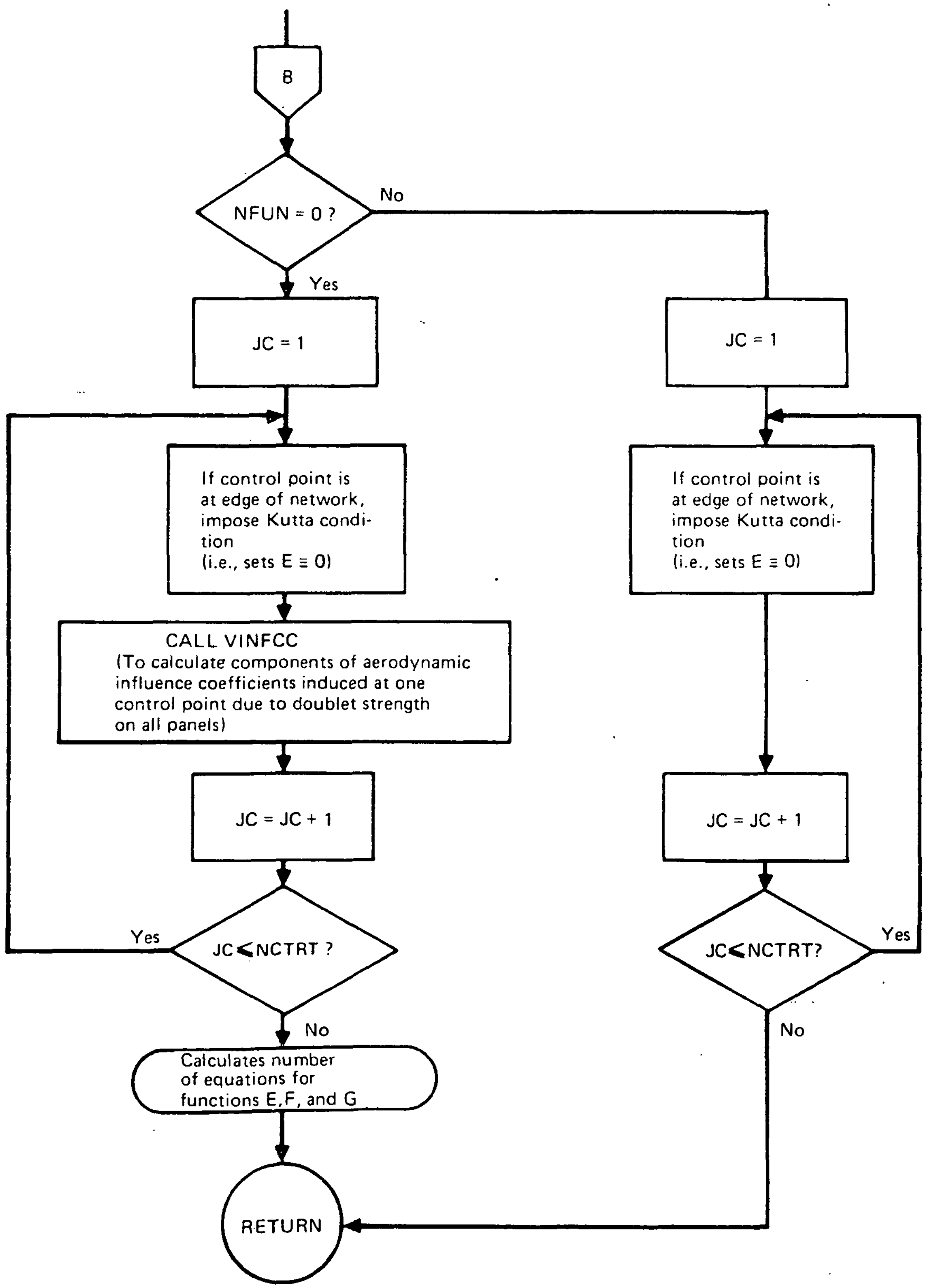
Flowchart of Program INPUT



Flowchart of Program AICGEN

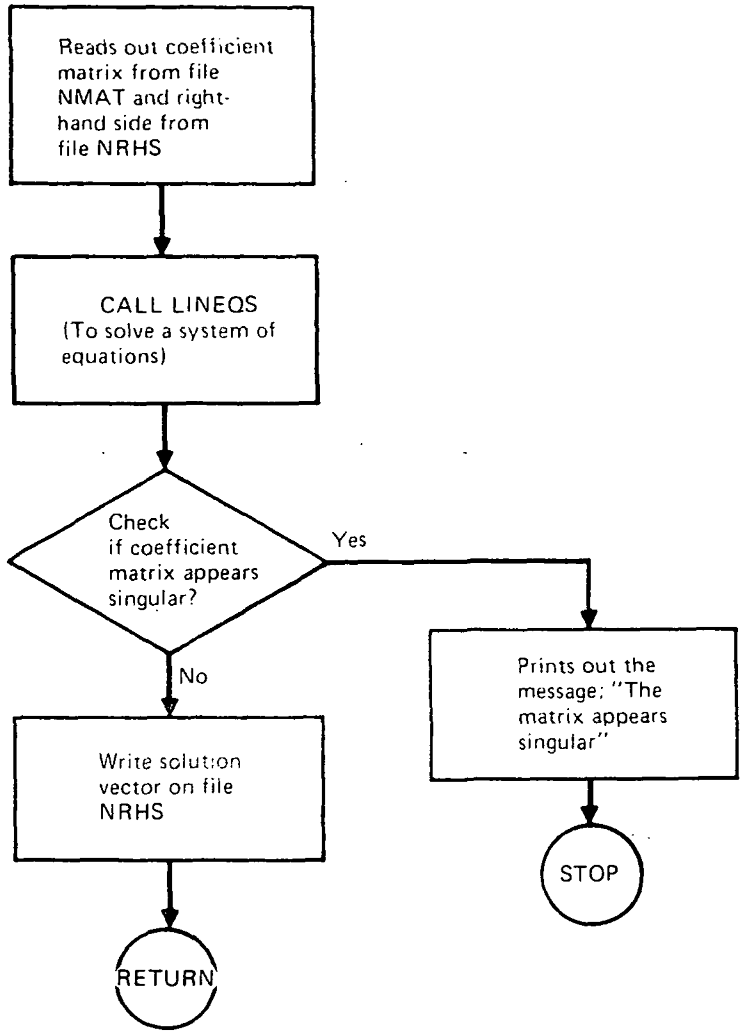


Flowchart of Program AICGEN (Continued)



Input	Common block /INDEX/-NT, NM, NN, NP, NZ, NPA, NZA, NNETT, NPANT, NZMPT /NITF/-NFUN /IPRINT/-IPGEOM, IPSING, IPCNTR, IPEIVC
Output	Common block /CM03/-NPIF, NAIC3, NAIC /BDYCS/-AC, ZCC, ZCR, ZDC, IPC, ITC /INDEX/-NS, NC, NSA, NCA, NSNGT, NCTRT /NINDX/-NEQ, NJC, ITC
Subroutines Called	TGEOMC, TSING, TCNTRL, EDGEIN, KSORT, PTRNS, IPTRNS, VINFCC, VIP
Discussion	The program first calls TGEOMC to provide least squares surface fit for each panel and to generate essential geometry information such as local coordinates of panel corner points and center, coordinate transformations, moments, etc. The subroutines TSING and TCNTRL are called to designate the locations of doublets and control points for each network according to different network types. The coefficients of quadratic doublet distribution for each panel are computed by using least squares method. The unit normal vector and the freestream velocity normal to the panel surface at each control point are also calculated. As discussed in the Engineering Document, equation (46), the stream surface boundary condition at edge points of networks where the Kutta condition has to be satisfied, gives a linear relation between the set of doublets (μ_e) at the edges of the network and the set of all remaining doublets (μ_r). Hence, doublets μ_e can be expressed in terms of all remaining doublets μ_r, and only μ_r will have to be treated as the explicit unknown parameters in the iterative procedure. For the convenience of computation, indices of doublets are rearranged so that μ_e's precede μ_r's. Indices of control points are also rearranged in a similar way. The components of aerodynamic influence coefficients (AIC) induced at each control point, due to doublet distributions of all panels, are computed by calling VINFCC. This process is repeated first for the control points at the edges of networks and then for all other control points. The program provides an option for skipping the last part of the computation. Thus, when subroutine ITFLOW in the main overlay calls AICGEN to update AIC at every KIT iteration, it sets NFUN=0, and the new AIC is generated. Otherwise, NFUN≠0 and the computation of new AIC is skipped.
Primary	OVERLAY(VORTEX,3,0)
Program	SOLVER
Purpose	To solve a linear system of equations A·X=B

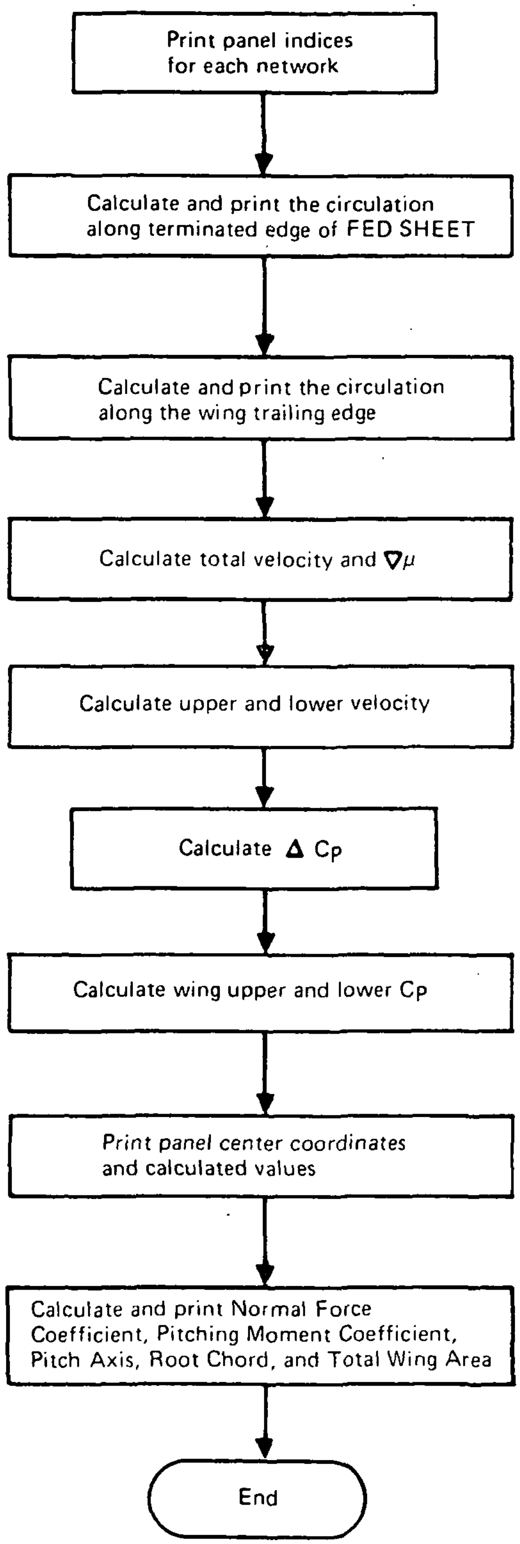
Flowchart of Program SOLVER



Input	Common block /NEQS/-NE, NR, NMAT, NRHS
Output	Common block /NEQS/-NRHS
Subroutines Called	LINEQS
Discussion	The program reads out coefficient matrix and right-hand side from two separate files and stores them in two different arrays A and B. The routine LINEQS is then called to solve the system of equations. If the coefficient matrix is not singular, the solution vector is written on the file that originally stores the right-hand side. Otherwise, an error message "The matrix appears singular" is printed, and execution of the computer program is terminated. The program has been set up with the consideration that an out-of-core equation solver can replace the present in-core one without changing the data structure significantly.

Primary	OVERLAY (VORTEX,4,0)
Program	OUTPUT
Purpose	To print the program results, including: circulation along terminated edge of the fed sheet network; circulation along the wing trailing edge; X, Y and Z coordinates of panel center, velocity on upper and lower surfaces, and Δc_p for each panel in the wing and free vortex networks; upper and lower c_p and area for each panel in the wing network; normal force coefficient; pitching moment coefficient; pitch axis x value; root chord length; and total wing area.
Input	Common Block /BDYCS/-ZC /CM03/-NPIF, NAIC3 /FSVEL/-FSV,XPITCH /INDEX/-NM,NPA,NZA /MSPNTS/-ZM /NFAJ/-NEQ,NF,NG /PANDQ/-AR,ART,C /SOLN/-S
Output	See Purpose
Subroutines Called	MMULT,PTRNS,SNGCAL,UVECT,VIP
Discussion	The Fed Sheet terminated edge points are calculated as the midpoints between each pair of outboard Fed Sheet panel corner points. SNGCAL is called to calculate the circulation at each edge point. The wing trailing-edge points are calculated as the midpoints between each pair of forward wake panel corner points. SNGCAL is again called to calculate the circulation at each point.

Flowchart of Program OUTPUT



The total velocity is calculated by multiplying the AIC matrix (DVDFS) by the vector, consisting of values S of doublet parameters and adding the components of the freestream velocity vector.

SNGCAL is called to calculate the value and 1st, 2nd derivatives of doublet strength at the panel center points of the wing and free sheet networks. The components of sheet vorticity (first derivatives of doublet strength) are transformed into the global panel system to define $\nabla\mu$(VELFS).

Upper and lower velocity is calculated by adding to and subtracting from the total velocity components, one-half of the $\nabla\mu$ components.

Δc_p is calculated by multiplying two times the vector inner product of $\nabla\mu$ and the total velocity vector.

c_p upper and lower for the wing network are calculated by the equation:

$$(\mathrm{VSQ} \pm \mathrm{HDCP} + 0.25*\mathrm{GMUSQ}) \qquad (1)$$

where VSQ is the velocity vector inner product, GMUSQ is the $\nabla\mu$ vector inner product, and HDCP is one-half of Δc_p.

The final calculations performed are for a summation table for the wing network consisting of normal force coefficient (c_N), pitching moment coefficient (c_m), pitch axis X value (XPITCH), root chord length (RCHORD) and total wing area (SW). The product (CNF) of the normal vector, Δ c_p, and wing panel area is used to calculate c_N and c_m.

c_N is defined as two times the sum of CNF at each wing panel divided by the total wing area.

c_m is defined as two times the sum of CNF at each wing panel times the X of the panel center minus XPITCH, divided by RCHORD times total wing area.

XPITCH is the X value of the pitch axis input by the user.

RCHORD is defined as the X value where the trailing edge breaks away from the planform centerline minus the X value of the first transverse cut.

Total wing area is simply the sum of the individual wing panel areas contained in array C in common block PANDQ.

FILE USAGE

There are eight disk files used in the computer program. The files are passed through common block /CM03/. Their disk numbers are assigned in program TEA378 (Main Overlay). Files NTSIN and NTSOUT are the standard input and output files. The following illustration gives where and how the other six files are used.

File	Where	Usage
NTGD	TEA378	Writing on components of freestream velocity at control points; reading out initial values of doublet parameters
	ITFLOW	Writing on residuals; reading out corrections
	DFGMU	Writing on $\partial G/\partial \mu_e$ and $\partial G/\partial \mu_r$
	DFGDT	Writing on $\partial G/\partial \theta$
	AICGEN	Transferring panel information
NPIF	FGCAL	Reading out panel information
	DFGMU	Reading out panel information
	DFGDT	Reading out panel information
	AICGEN	Storing panel information
	VINFCC	Reading out panel information
	OUTPUT	Reading out panel information
NAIC3	FGCAL	Reading out components of influence coefficient
	DFGMU	Reading out components of influence coefficient
	AICGEN	Storing components of influence coefficient
	OUTPUT	Reading out components of influence coefficient
NAIC	FGCAL	Reading out rows of AIC matrix
	AICGEN	Storing rows of AIC matrix
NJAC	ITFLOW	Reading out rows of Jacobian
	DFGMU	Writing on $\partial F/\partial \mu_e$ and $\partial F/\partial \mu_r$
	DFGDT	Storing rows of Jacobian
NSCR	ITFLOW	Writing on rows of updated Jacobian
	DFGMU	Writing on partial derivatives with respect to doublet parameters
	DFGDT	Reading out partial derivatives with respect to doublet parameters

COMMON BLOCK DEFINITION

Variables of the more essential common blocks shared by main and primary overlays and some subroutines are defined below.

Common Block	Variables	Description	Where Set
ADR	RTD	Number of degrees in 1 rad	TEA378
	DTR	Number of radians in 1 deg	
BDYCS	ZC	Array of X,Y,Z coordinates of control points	TCNTRL
	ZCC	Array of normal vectors at control points on the panel surface	
	ZCR	Array of components of freestream velocity normal to the panel surface at control points	
	ZDC	Array of perturbation distances for control points at edges of networks	
	IPC	Array of panel indices for control points	
	ITC	Array of codes (= 1 for control points at edges of networks; =0 otherwise)	
CM03		See file usage	TEA378
DAT3	AR	Aspect ratio	INPUT
	NTR	Number of transverse cuts	
	XTR	Array of transverse cut x values	
	MSP	Number of spanwise wing percent values	
	YSP	Array of spanwise wing percent values	
	NTC	Number of transverse cuts intersecting root chord	
	NLE	Number of points defining wing leading edge	
	YLE	Array of Y coordinates of leading-edge points	

Common Block	Variables	Description	Where Set
	NTE	Number of wing trailing-edge, transverse-cut intersections	
	YTE	Array of Y coordinates of points defined by NTE	
	MFS	Number of spanwise cuts in the free vortex network	
EEQS	EMUE	Array of $\partial E/\partial \mu_e$	FGCAL
	EMU	Array of $\partial E/\partial \mu_r$ Array of $(\partial E/\partial \mu_e)^{-1}\,(\partial E/\partial \mu_r)$	 DFGMU
	IPR	Array of indices of row pivoting for equation solver	FGCAL
FLATP	NFLTP	=1 for flat panel; =0 otherwise	INPUT
FSVEL	FSV	Freestream velocity vector	INPUT
	FSVM	Freestream velocity magnitude	
	ALPHA	Wing angle of attack in radians	
	XPITCH	X value of pitch axis	
	RCHORD	Length of root chord	
ICONST	PI	π	AICGEN
	PI2	2π	
	PI4I	$1/4\pi$	
INDEX	NT	Array containing network type indices 2,4,6,5,7	INPUT
	NM	Array containing the number of spanwise panel points in each network	
	NN	Array containing the number of transverse cuts in each network	
	NP	Array containing the number of panels in each network	
	NS	Array containing the number of singularity parameters in each network	SING
	NC	Array containing the number of control points in each network	CONTRL
	NZ	Array containing the number of panel points in each network	

Common Block	Variables	Description	Where Set
	NPA	Array containing cumulative sum of array NP	
	NSA	Array containing cumulative sum of array NS	TSING
	NCA	Array containing cumulative sum of array NC	TCNTRL
	NZA	Array containing cumulative sum of array NZ	
	NNETT	Number of networks	
	NPANT	Number of total panels	
	NSNGT	Number of total singularity parameters	TSING
	NCTRT	Number of total control points	TCNTRL
	NZMPT	Number of total panel corner points in all networks	
INTQ	H	Array of H integrals	INTCAL
	HZ	Z distance of field point above panel	
	IH	Indicates necessity of using special expansion when field point is near panel surface	
	MXQ	Maximum value of m+n for H integrals required to compute influence coefficients	DPIV
	MXK	Maximum value of k for H integrals required to compute influence coefficients	
IPRINT	IPNPUT	Controls printout of intermediate results from program INPUT in overlay (1,0)	TEA378
	IPGEOM	Controls printout of intermediate results from subroutines GEOMC and SURFIT in overlay (2,0)	
	IPSING	Controls printout of intermediate results from subroutine SING in overlay (2,0)	
	IPCNTR	Controls printout of intermediate results from subroutine CONTRL in overlay (2,0)	

Common Block	Variables	Description	Where Set
	IPEIVC	Controls printout of intermediate results from subroutine EIVC in overlay (2,0)	
	IPOUTP	Controls printout of intermediate results from program OUTPUT in overlay (4,0)	
LSQSFC	ZK	X,Y,Z coordinates of corner points used in least squares fit	SURFIT,SING
	WTK	Weights used in least squares fit	
	AK	Generalized inverse from least squares fit	LSQSF
	NO	=2 for quadratic fit (6 terms) $<$2 for linear fit (3 terms)	SURFIT,SING
	NPK	Number of data points used in least squares fit	
MSPNTS	ZM	Array containing panel corner point X,Y,Z coordinates	INPUT
	ZL	Array containing panel lengths along transverse cuts	
NEQS	NE	Number of equations to be solved	TEA378,ITFLOW
	NR	Number of right-hand sides	
	NMAT	Name of file storing the coefficient matrix	
	NRHS	Name of file storing right-hand side or solution vector	
NFAJ	NEQ	Number of equations corresponding to control points at edges of networks	AICGEN
	NF	Number of equations corresponding to all other control points (excluding those at edges)	
	NG	Number of equations corresponding to number of panels of the free sheet network	
NINDX	NEQ	Number of equations corresponding to control points at edges of networks	EDGEIN
	NJC	NJC(j) gives new index for control point j	

Common Block	Variables	Description	Where Set
	IJC	IJC(k) gives the control point which has new index k	
NITF	NFUN	Number of functions called at every KIT iteration	ITFLOW
	JT	Iteration number	
	ITMX	Maximum number of iterations	INPUT
	KIT	Number of iterations to generate new AIC	ITFLOW
	ITPRIN	Printing output occurs at every ITPRIN iteration	INPUT
PANDQ	CP	X,Y,Z coordinates of panel corner points	GEOMC
	PC	Average X,Y,Z coordinates of four corner points	SURFIT
	RO	X,Y,Z coordinates of origin of local panel coordinate system	
	AR	Rotation matrix for transforming from global X,Y,Z coordinates to local panel ξ,η,ζ coordinates	
	ART	Transposition of AR	GEOMC
	P	ξ,η coordinates of panel corner points	SURFIT
	A	Panel principal curvature in ξ direction	
	B	Panel principal curvature in η direction	
	DIAM	Length of longest diagonal of panel	GEOMC
	C	Array of (ξ,η) moments	CCAL
	AST	Matrix defining dependence of coefficients of quadratic doublet distribution on free parameters	SING
	IIS	Array containing indices of free parameters on which panel doublet coefficients depend	
	INS	Number of free singularity parameters determining coefficients of panel quadratic doublet distribution	

Common Block	Variables	Description	Where Set
	ITS	Panel singularity type	
	NPDQ	Total number of panel-defining quantities in PANDQ common block, not used in this program	Not set
PINDX	KP	Index for keeping track of the position of record (panel information) on disk to be written	Wherever routines PTRNS and IPTRNS are called
	KQ	Index for keeping track of the position of record (panel information) on disk to be read	
	NPWR	Disk file on which panel information is to be written	
	NPRD	Disk file from which panel information is to be read	
PIVINT	XX	Local panel coordinates of control point	PIVC
	PP	Local coordinates of panel corner points	
	AA	Coefficients of the quadratic surface	
	BB	fit of the panel	
	DDIAM	Maximum diagonal of the panel	
	CC	Moments for the panel	
	DVDV	Integrals from routine DPIV	
	NTST	Number of coefficients in panel doublet distribution	
	NCF	Not used in this program	Not set
SIDEQ	QSIDE	Collection of geometric quantities describing relationship of field point to panel side	SIDECL
SKAIC1	AKS1	Coordinate of field point relative to panel corner point expressed in local system	INTCAL
	AET1	Coordinate of field point relative to panel corner point expressed in local system	
	AKS2	Coordinate of field point relative to panel corner point expressed in local system	

Common Block	Variables	Description	Where Set
	AET2	Coordinate of field point relative to panel corner point expressed in local system	
	DRM	Length of panel edge	
	EL1	Distance from panel corner point to projection of field point on panel edge line	
	EL2	Distance from panel corner point to projection of field point on panel edge line	
	ELM	Minimum value of distance from panel edge to projection of field point on panel edge line	
	ANK	Component of unit normal to panel edge	
	ANE	Component of unit normal to panel edge	
	A	Distance from projection of field point on panel plane to panel edge line	
	AA	Square of A	
	GG	Square of distance from field point to panel edge line	
	S1	Distance from field point to end point of a panel side	
	S2	Distance from field point to end point of a panel side	
	S1I	Inverse of S1	
	S2I	Inverse of S2	
	HM	Magnitude of HZ	
	HH	Square of HZ	
SKAIC2	GAK	Accumulates A times FK over four panel sides	FKCAL
	GKNK	Accumulates ANK times FNK over four panel sides	FNKCAL
	GENK	Accumulates ANE times FNK over four panel sides	
	GKMN	Accumulates ANK times FMN over four panel sides	FMNCAL

Common Block	Variables	Description	Where Set
	GEMN	Accumulates ANE times FMN over four panel sides	
	GAMN	Accumulates A times FMN over four panel sides	
	H111	Accumulates H(1,1,1) over four panel sides	FKCAL
	FK	F integrals for M=1 and N=1	
	FNK	F integrals for M=1	FNKCAL
	FMN	F integrals for K=1	FMNCAL
	E	E functions used for computation of F integrals	ECAL
SKAICI	MXFK	Maximum value of K for F integrals	INTCAL
	MXFKN	Maximum value of K for F integrals when field point is near panel edge and special expansion is required	FKCAL
	MXFNK	Not used in this program	Not set
	MXKM2	Used as upper limit for certain K loops	INTCAL
	MXKM4	Used as upper limit for certain K loops	
	MXQM1	Used as upper limit for certain M or N loops	
SKAICL	LMXQ2	Logical variable used to circumvent unnecessary calculations	INTCAL
	LMXQ3	Logical variable used to circumvent unnecessary calculations	
	LMXQ4	Logical variable used to circumvent unnecessary calculations	
	LMXK3	Logical variable used to circumvent unnecessary calculations	
	LMXK5	Logical variable used to circumvent unnecessary calculations	
	LMKEX	Logical variable determining necessity of using special expansion for field point near panel surface	

Common Block	Variables	Description	Where Set
SKRCH1	ZA	Array of network points serving as control points or locations of free singularity parameters	GCPCAL
	IA	Index array which counts nonidentical points in ZA	GRDIND
SOLN	S	Array of values of doublet parameters	TEA378,ITFLOW
	ZA	Array containing panel inclination angles	INPUT
SYMM	NSYMM	=1 for axisymmetric =0 otherwise	INPUT

DESCRIPTION OF SUBROUTINES

The subroutines are arranged in alphabetical order.

<u>Subroutine</u>	<u>AJGEN(X,N)</u>
Purpose	To obtain the analytic Jacobian for perturbation variables (doublet parameters excluding those at edges and angles) assuming D(AIC)/D(THETA)=0
Input	Calling sequence X—Array of values for the variables N—Number of variables Common block /INDEX/—NM, NN, NZ /MSPNTS/—ZM /ADR/—DTR
Output	Common block /SOLN/—S, ZA
Subroutines Called	DFGMU, DFGDT
Discussion	The routine stores values of doublet parameters (excluding those at edges) and angles in array S and ZA, respectively. Routines are called to generate the partial derivatives of functions F and G with respect to doublet parameters MU, excluding those at edges (DFGMU) and angles THETA (DFGDT).

<u>Subroutine</u>	<u>AWNET</u>
Purpose	To calculate the coordinates of all panel corner points in an arrow wing planform configuration
Input	Common block /DAT3/—AR, NTR, XTR, MSP, YSP, NTC
Output	Common block /DAT3/—YLE, NTE /MSPNTS/—ZM
Subroutines Called	SWEPTE
Discussion	The Y coordinates of the panel corner points at the intersection of the leading edge and transverse cuts are computed by multiplying the X value of the transverse cut by one-fourth the aspect ratio. The Y coordinates of the panel corner points between the leading edge and root chord on the transverse cuts are computed by multiplying the Y coordinate at the leading edge by the array of percent values YSP.

Subroutine SWEPTE is called to calculate the Y coordinates of all panel points aft of the root chord.

The X coordinates of the panel corner points are the X values of the transverse cuts input by the user. All Z coordinates are set to zero.

Subroutine	BSUBSM(A,NR,N,IPR,B,M)
Purpose	To perform back substitutions using the factorization obtained from a decomposition routine and find the solution for a system of equations
Input	Calling sequence A —The lower triangle of the array consists of a lower triangular matrix L and the upper triangle consists of an upper triangular matrix U. They are obtained from a decomposition routine such as TDECOM NR —Maximum row dimension of arrays A and B N —Order of the coefficient matrix IPR—Array consists of numbers of pivotal row, as derived from the subroutine TDECOM B —Array consists of M right-hand sides of the linear system M —Number of right-hand sides
Output	Calling sequence B —Solution vectors
Subroutines Called	VIPS
Discussion	The routine first uses pivotal information given in the array IPR to exchange elements of right-hand sides. It then performs forward substitution by solving the lower triangular system of equations LY=B and backward substitution by solving the upper triangular system of equations UX=Y. X is the desired solution of the given system of equations. The routine is a modified version of a routine in the subroutine library of the Boeing Computer Services company.

Subroutine	CCAL(P,C)
Purpose	To calculate for each panel the quadrilateral moment integrals used in the computation of the source and doublet far-field velocity influence coefficients. (See sec. B.4, app. B, of the Engineering Document.)
Input	Calling sequence P—Coordinates of four corner points of quadrilateral
Output	Calling sequence C—Array of moment integrals
Subroutines	ECAL, ZERO

Called

Discussion — The routine computes the quadrilateral moment integrals C(M,N)=I(SIGMA, KSE**(M-1)*ETA**(N-1), DKSE*DETA) for M=1, MXQ and N=1, MXQ-M+1. A description of the calculations performed is contained in section B.4 of appendix B of the Engineering Document. The relevant equations are (B-93) through (B-102). The relevant procedure is procedure 6. The code closely follows the development and notation of this portion of appendix B.

Subroutine — **CMAB(A,B,R,NRA,NCA,NCB)**

Purpose — To multiply two matrices whose elements are stored compactly by row (compass)

Input — Calling sequence
A—Location of first matrix
B—Location of second matrix
R—Location of resultant matrix
NRC—Number of rows in first matrix
NCA—Number of columns in first matrix
NCB—Number of columns in second matrix

Output — Calling sequence
R—Resultant matrix

Subroutines Called — None

Discussion — Performs the matrix operation (R)=(B)(A).

Subroutine — **CONTRL(NT,NM,NN,NC,NPA,ZM,ZC,ZCC,ZCR,ZDC,IPC,ITC)**

Purpose — To compute control point defining quantities for each network

Input — Calling sequence
NT—Network type
NM—Number of spanwise cuts in the network
NN—Number of transverse cuts in the network
NPA—Total number of panels in all previous networks
ZM—Coordinates of corner points in the network

Common block
/IPRINT/—IPCNTR
/FSVEL/—FSV
/PANDQ/—PC

Output — Calling sequence
NC—Number of control points on the network
ZC—Coordinates of control points on the network
ZCC—Surface normal vector at control points

ZCR—Normal components of freestream velocity
ZDC—Relocation distance of control point
IPC—Sequence number of panel to which control point belongs
ITC—Network edge control point indicator

Subroutines Called — GCPCAL, GRDIND, PTRNS, SURPRO, MMULT

Discussion — The routine calculates quantities associated with the control points and boundary conditions of the problem. Separate computations are performed for each network type. First, the control points (points at which the boundary conditions are applied) are located. This is done by averaging certain combinations of corner points and then projecting the resultant points onto the panel surfaces. Those control points located on a network edge are withdrawn slightly from the edge and not projected onto their panel surfaces to avoid numerical difficulty later. The control points are ordered and indexed along with auxiliary quantities which are computed as well. Such quantities include the panel normal at the control point, the component of freestream velocity in this direction (for use in applying the boundary conditions), and the distances the edge control points are withdrawn.

Subroutine — CROSS(A,B,C)

Purpose — To calculate the cross product of two vectors

Input — Calling sequence
A—First vector
B—Second vector

Output — Calling sequence
C—Resultant vector

Subroutines Called — None

Discussion — CROSS performs the following calculations:
C(1)=(A(2)*B(3))-(A(3)*B(2))
C(2)=(A(3)*B(1))-(A(1)*B(3))
C(3)=(A(1)*B(2))-(A(2)*B(1))

Subroutine — DFGDT(ZM,NM,NN)

Purpose — To calculate partial derivatives of functions F and G with respect to panel inclination angles of free sheet, assuming D(AIC)/D(THETA)=0

Input — Calling sequence
ZM—Coordinates of corner points of free sheet network
NM—Number of spanwise cuts of network
NN—Number of transverse cuts of network

	Common block /CM03/—NSCR /BDYCS/—ZC /FSVEL/—FSV /NFAJ/—NEQ, NF, NG /ADR/—DTR
Output	Common block /CM03/—NJAC
Subroutines Called	PTRNS, CROSS, UVECT, VIP, UNIPAN, MMULT
Discussion	A detail discussion of the formula used in the computation is given in the Engineering Document (see app. C, geometry update coefficients). The routine first finds a normal vector N for the panel. It then computes partial derivatives of N with respect to angle THETA and forms partial derivatives of N·V and of pressure jump with respect to THETA. Finally, it stores all partial derivatives in proper position of the Jacobian.

Subroutine	DFGMU
Purpose	To calculate partial derivatives of functions F and G with respect to doublet parameters (excluding those at edges)
Input	Common block /CM03/—NPIF, NAIC3 /BDYCS/—ZC /INDEX/—NSNGT /FSVEL/—FSV /NFAJ/—NEQ, NF, NG /SOLN/—S /EEQS/—EMUE, EMU, IPR
Output	Common block /CM03/—NSCR
Subroutines Called	BSUBSM, PTRNS, MMULT, UNIPAN, VIPS
Discussion	The formula and notation used here are discussed in detail in the Engineering Document (see app. D, doublet strength update coefficients). The routine reads in DE/DMUE and DE/DMU and calculates (DE/DMUE)(-1)*(DE/DMU), where E is the function consisting of only those equations corresponding to control points at edges. Then, it obtains partial derivatives of N·V on wing and on free sheet with respect to doublet parameters. Partial derivatives of pressure jump V·GRAD(MU) with respect to doublet parameter are also calculated. Finally, partial derivatives with respect to doublet parameters excluding those at edges are formed.

Subroutine	DPIV
Purpose	To calculate the velocity influence coefficients induced at a field point by a doublet panel
Input	Common block /ICONST/—PI2, PI4I /PIVINT/—X, P, A, B, DIAM, C, NTST
Output	Common block /PIVINT/—DV
Subroutines Called	INTCAL, ZERO
Discussion	The routine computes the doublet panel velocity influence coefficients at a specified field point. A description of the method and calculations performed is contained in appendix B of the Engineering Document. If the field point is sufficiently distant from the panel, a far-field approximation is employed. The approximation and computational method is presented in section B.4 of appendix B and the related code comprises the part of DPIV between statement 120 and statement 500. The loop 450 contains the bulk of the calculations, and its purpose is to compute the J vectors of equation (B-91). For this calculation the terms on the right side of equation (B-91) have been expanded; hence, the code does not directly correlate with this formula. Another evaluation procedure is employed when the field point is near the panel. A description of this procedure is presented in sections B.2 and B.3 of appendix B. The related code comprises the part of DPIV between statements 500 and 900. The loop 750 calculates the vector J, defined by equation (B-34), with the H integrals computed by the routine INTCAL. The loop 800 transforms the influence coefficients relative to the expansion of doublet strength about the projection of the field point to coefficients relative to the expansion of doublet strength about the origin.

Subroutine	DWNET
Purpose	To calculate the coordinates of all panel corner points in a delta wing planform configuration
Input	Common block /DAT3/—AR, NTR, XTR, MSP
Output	Common block /DAT3/—YLE /MSPNTS/—ZM

Subroutines Called — None

Discussion — The Y coordinates of panel corner points at the intersection of the leading edge and the transverse cuts are computed by multiplying the X value of the transverse cut by one-fourth the aspect ratio.

The Y coordinates of the panel corner points between the leading edge and root chord on the transverse cuts are computed by multiplying the Y coordinate at the leading edge by the array of percent values YSP.

The X coordinates of the panel corner points are the X values of the transverse cuts input by the user. All Z coordinates are set to zero.

Subroutine — ECAL(X1,X2,A1,A2,E,N)

Purpose — To evaluate E(I) = A2*X2**(I-1) - A1*X1**(I-1); I = 1,N. (See eq. (B-59), app. B, of the Engineering Document.)

Input — Calling sequence
X1—(see Purpose)
X2—(see Purpose)
A1—(see Purpose)
A2—(see Purpose)
N—(see Purpose)

Output — Calling sequence
E—(see Purpose)

Subroutines Called — None

Discussion — The routine calculates the quantities:
E(I)=A2*X2**(I-1)-A1*X1**(I-1) for I=1,N using the recursion formula E(I)=(X1+X2)*E(I-1)-X1*X2*E(I-2) and the initial conditions E(1)=A2-A1 and E(2)=A2*X2-A1*X1.

Subroutine — EDGEIN

Purpose — To provide new indices for the control points and doublets so that the corresponding equations (downwash condition) and doublets at edges of network will precede all the others

Input — Common block
/BDYCS/—ITC
/INDEX/—NCTRT

Output — Common block
/NINDX/—NEQ, NJC, IJC

Subroutines Called — None

Discussion — The routine obtains the number of equations corresponding to control points at edges. Then it assigns indices according to whether control points are at edge or interior.

Subroutine	EIVC(ZC,ZNC,ZDC,IPINF)
Purpose	To calculate the velocity induced by a doublet panel on a network edge control point
Input	Calling sequence ZC—Coordinates of control point ZNC—Unit normal to surface at control point ZDC—Distance from control point to panel edge
	Common Block /IPRINT/—IPEIVC /ZIP/—IPZ, IP, JCZ /PANDQ/—CP, PC, RO, AR, P, DIAM /SYMM/—NSYMM
Output	Calling sequence IPINF—Indicates whether panel is close enough to control point to induce a substantial downwash
	Common block /PIVM/—DVDS
Subroutines Called	ZERO, CROSS, UNIPAN
Discussion	The routine calculates the velocity induced by a doublet panel (and its image if configuration is symmetrical) on a network edge control point. The influence is computed by accumulating the influence of each panel edge. The influence of a panel edge is ignored unless a point on the edge is within a small sphere around the control point. In this case, the influence, resulting from both the doublet strength and its derivative perpendicular to the edge (evaluated at that edge point), is computed. The resultant velocity is then distributed among the coefficients of the doublet distribution on the panel.

Subroutine	FGCAL(FVZ,GVZ)
Purpose	To solve for doublet parameters at edges and to calculate functions F and G
Input	Common Block /CM03/—NPIF, NAIC3, NAIC /BDYCS/—ZC, ZCR /FSVEL/—FSV /NFAJ/—NEQ, NF, NG /SOLN/—S
Output	Calling sequence FVZ—Values of F GVZ—Values of G
	Common block /EEQS/—EMUE, EMU, IPR

Subroutines Called	VIPS, LINEQS, PTRNS, MMULT, VIP, UNIPAN
Discussion	The routine reads rows of AIC matrix to form coefficients of function E. The solution for doublet parameters (MUE) at edges are found by using function E and given values of all other doublet parameters (MU). Since E is a function of doublet parameters only, DE/DMUE and DE/DMU are simply the coefficients of E. If the matrix DE/DMUE is singular, an error message will be printed and the execution of the computer program will be terminated. Components of influence coefficients are read in and multiplied by values of doublet parameters to form perturbation velocity. The latter is added to freestream velocity to become the average velocity vector V. The DOT product N•V is then calculated for every interior control point on wing (forming part of function F) and on free sheet (forming function G). The jump in pressure coefficients V•GRAD(MU) (see Engineering Document) on free sheet is also calculated (forming the other part of function F).

Subroutine	FKCAL
Purpose	To calculate certain F integrals used to compute the H integrals involved in the formulas for the source and doublet panel induced velocity influence coefficients. (See sec. B.3 of app. B of the Engineering Document.)
Input	Common block /SKAICL/—LMKEX /SKAIC1/—EL1, EL2, ELM, A, AA, GG, S1, S2, S1I, S2I, HM /SKAICI/—MXFK
Output	Common block /SKAICI/—MXFKN SKAIC2/—GAK,H111
Subroutines Called	ECAL
Discussion	The routine computes the integrals F (1,1,K) for K=1,MXFK where F(1,1,K)=I(L,1./RHO**K,DL). A description of the calculations performed is contained in section B.3 of appendix B of the Engineering Document. The relevant equations are (B-60), (B-61), (B-68), and (B-69). The relevant procedures are 4 and 5. The routine also computes the arctangent terms of step 1 (eq. (B-41) of procedure 1. The code closely follows the development and notation of section B.3. Note that FNK(N,K)=F(1,N,K).

Subroutine	FMNCAL
Purpose	To calculate certain F integrals used to compute the H integrals involved in the formulas for the source and doublet panel induced velocity influence coefficients. (see sec. B.3 of app. B of the Engineering Document.)
Input	Common block /INTQ/—MXQ /SKAICL/—LMXQ2, LMXQ3 /SKAIC1/—AKS1, AET1, AKS2, AET2, ANK, ANE, A, AA, S1, S2, HH
Output	Common block /SKAIC2/—GKMN, GEMN, GAMN
Subroutines Called	ECAL
Discussion	The routine computes the integrals F(M,N,1) for N=1,MXQ and M=1,MXQ-N+1 where F(M,N,1)=I(L,KSE**(M-1)*ETA**(N-1)/RHO, DL). A description of the calculations performed is contained in section B.3 of appendix B of the Engineering Document. The relevant equations are (B-62), (B-63), (B-64), and (B-65). The relevant procedures are 4 and 5. The code closely follows the development and notation of section B.3. Note that FMN(M,N)=F(M,N,1).

Subroutine	FNKCAL
Purpose	To calculate certain F integrals used to compute the H integrals involved in the formulas for the source and doublet panel induced velocity influence coefficients. (see sec. B.3 of app. B of the Engineering Document.)
Input	Common block /SKAICL/—LMXQ3, LMXQ4, LMXK5 /SKAIC1/—ANK, ANE, AA, S1I, S2I, HH /SKAICI/—MXKM2, MXQM1
Output	Common block /SKAIC2/—GKNK, GENK
Subroutines Called	ECAL
Discussion	The routine computes the integrals F(1,N,K) for N=2,MXQ and K=3,MXK-2,2 where F(1,N,K)=I(L,ETA**(N-1)/RHO**K,DL). A description of the calculations performed is contained in section B.3 of appendix B of the Engineering Document. The relevant equations are (B-66) and (B-67). The relevant procedures are procedures 4 and 5. The code closely follows the development and notation of section B.3. Note that FNK(N,K)=F(1,N,K).

Subroutine	FUNC(X,N,RX)
Purpose	To evaluate function F (N·V on wing and V·GRAD(MU) on free sheet) and G (N·V on free sheet)
Input	Calling sequence X—Array of values for the variables N—Number of variables Common block /NFAJ/—NEG, NF, NG /NITF/—NFUN /SOLN/—ZA /ADR/—DTR
Output	Calling sequence RA—Array of values of functions
Subroutines Called	UPDATE, AICGEN(OVERLAY-2,0), FGCAL
Discussion	The routine stores values of doublet parameters (excluding those at edge) and angles in arrays S and ZA, respectively. It uses new angles to update the corner points of free sheet, fed sheet, and part of the wake network. AICGEN(OVERLAY-2,0) is then called to designate locations of doublets and control points and to generate velocity components and AIC matrix using the updated corner points. If perturbation in angle is not significant, UPDATE and AICGEN are skipped. Finally, the routine calls FGCAL to calculate values of functions F and G.

Subroutine	GCPCAL(NM,NN,NM1,NN1,ZM,ZA)
Purpose	To construct an NM+1 by NN+1 grid of points from corner point data
Input	Calling sequence NM—Number of corner points in a row NN—Number of corner points in a column NM1—Number of grid points in a row (NM+1) NN1—Number of grid points in a column (NN+1) ZM—Coordinates of corner points
Output	Calling sequence ZA—Coordinates of grid points
Subroutines Called	None
Discussion	The routine computes an NM+1 by NN+1 grid of points derived from corner point data. The points in the grid consist of the average of each set of four adjacent corner points, the average of each set of two adjacent edge corner points, and the four extreme corner points. These points are obtained by computing approximate averages of the corner points.

Subroutine	GEOMC(NT,NM,NN,NPA,ZM)
Purpose	To calculate geometric defining quantities for each panel in a network
Input	Calling sequence NT—Network type NM—Number of spanwise cuts in network NN—Number of transverse cuts in network NPA—Total number of panels in all previous networks ZM—Coordinates of corner points in the network Common block /IPRINT/—IPGEOM
Output	Common block /PANDQ/—CP,PC,RO,AR,ART,P,A,B,DIAM,C
Subroutines Called	SURFIT, CCAL, IPTRNS
Discussion	The routine calculates and stores geometric defining quantities for each panel of a network. First, the four grid points defining the panel corner points are found. Together with adjacent grid points, these corner points are fed into SURFIT, which defines the actual panel surface and the local panel coordinate system. Then CCAL is called to calculate panel moments. Finally, all the panel-defining quantities are stored on a file.

Subroutine	GRDIND(NM,NN,Z,I,IS)
Purpose	To order nonidentical points of an NM by NN grid of points via an index array
Input	Calling sequence NM—Number of grid points in a row NN—Number of grid points in a column Z—Coordinates of grid points
Output	Calling sequence I—Index array containing sequence number of each grid point IS—Total number of nonidentical points in a grid
Subroutines Called	PIDENT
Discussion	The routine sequences an NM by NN grid of points. The sequencing proceeds in the order ((M=1,NM),N=1,NN), where (M,N) is the point in row M and column N. Any point identical with the point in the same row and previous column or with the point in the same column and previous row is assigned the same sequence number as that point. The sequence numbers of the grid points are stored in an NM x NN index array and returned as output along with the total number of nonidentified points.

Subroutine	GWNET
Purpose	To calculate the coordinates of all panel corner points in a gothic wing planform configuration
Input	Common block /DAT3/—NTR,XTR,MSP,YSP,NTC,YLE
Output	Common block /DAT3/—NTE /MSPNTS/—ZM
Subroutines Called	SWEPTE
Discussion	The Y coordinates of panel corner points at the intersection of the leading edge and transverse cuts are input by the user. The Y coordinates of panel corner points between the leading edge and root chord on the transverse cuts are computed by multiplying the Y coordinate at the leading edge by the array of percent values YSP. Subroutine SWEPTE is called to calculate the Y coordinates of all panel points aft of the root chord. The X coordinates of the panel corner points are the X values of the transverse cuts input by the user. All Z coordinates are set to zero.

Subroutine	INTCAL
Purpose	To compute the H integrals involved in the formulas for the source and doublet panel induced velocity influence coefficients. (See sec. B.3 of app. B of the Engineering Document.)
Input	Common block /INTQ/—MXQ,MXK /PIVINT/—X,P,AC,BC,DIAM
Output	Common block /INTQ/—H,HZ,IH
Subroutines Called	SIDECL, ZERO, TRNSFR, FKCAL, FMNCAL, FNKCAL
Discussion	The routine calculates the integrals H(M,N,K)=1(SIGMA,KSE**(M-1) *ETA**(N-1)/RHO**K,DKSE*DETA) for M=1,MXQ and N=1,MXQ-M+1 and K=1,MXK,2. A description of the calculations performed is contained in section B.3 of appendix B of the Engineering Document. The routine can be divided into three parts. In the first part, preliminary quantities concerning the geometric relationship of the field point to the quadrilateral are calculated. In the second part, the F integrals are calculated for each side of the quadrilateral and accumulated. In the third part, procedure 1, 2, or 3 is executed.

Subroutine	IPTRNS(IP)
Purpose	To write panel information on disk
Input	Calling sequence IP—Panel number of information to be written Common block /PANDQ/—CP,PC,RO,AR,ART,P,A,B,DIAM,C,AST,IIS,INS,ITS /PINDX/—KP,NPWR
Output	Common block /PINDX/—KP
Subroutines Called	None
Discussion	Writes 197 words of panel information from common block PANDQ onto disk file specified by NPWR.

Subroutine	ITFLOW(X,N,RX,DX,Y,RY)
Purpose	To perform iterative scheme using quasi-Newton algorithm for the solution of a set of nonlinear equations
Input	Calling sequence X—Array of initial values for the variables N—Number of variables DX,Y,RY—Scratch arrays Common block /NFAJ/—NEQ,NF /NITF/—ITMX,ITPRIN
Output	Calling sequence X—Array of solution vector RX—Array of residual vector
Subroutines Called	VIP, FUNC, OUTPUT(OVERLAY-4,0), AJGEN, SOLVER (OVERLAY-3,0)
Discussion	The routine calls FUNC to evaluate residuals RX and calls AJGEN to set up the Jacobian AJ. The system of equations AJ*DX=-RX is solved and a new approximate solution is found using corrections DX. Residuals and Jacobian are evaluated at the new solution. The procedure is repeated until the sum of squares of residuals satisfies a predetermined tolerance TOL or the given maximum number of iterations ITMX is reached. The routine includes a procedure of generating new AIC after every kit iteration. The Jacobian will be calculated by calling AJGEN only when new AIC is generated. Otherwise, it will be updated by using a formula of quasi-Newton scheme (see Engineering Document). Number of iteration, sum of squares of residuals, and step size are printed for every ITPRIN iteration. For iteration study and checkout purpose, some other intermediate print statements are included (see listing).

Subroutine	KSORT
Purpose	To sort the column of a two-dimensional array using the given key index array
Input	Calling sequence A—Array of which the column is to be sorted M—Number of rows of A N—Number of columns of A KEY—Array consists of given key indices W—Working array of same dimension as A
Output	Calling sequence A—The sorted array
Subroutines Called	None
Discussion	The contents of array A are stored in a working array using the indices given in key array. Working array is then transferred back to array A.

Subroutine	LINEQS(A,NR,N,IPR,B,M,D1)
Purpose	To solve a system of linear equations $A*X=b$
Input	Calling sequence A—Array consists of elements of the coefficient matrix NR—Maximum row dimension of arrays A and B N—Order of the coefficient matrix B—Array consists of M right-hand sides of the linear system M—Number of right-hand sides
Output	Calling sequence A—The lower triangle of the array consists of a lower triangular matrix L, and the upper triangle consists of an upper triangular matrix U. (Since U is unit upper triangular, its diagonal elements are not stored.) IPR—Array gives numbers of pivotal row (a record of interchanges) B—Solution vectors D1—=+1 or -1 according to the number of interchanges being even or odd. It also indicates successful return; = 0 indicates that the coefficient matrix appears singular.
Subroutines Called	TDECOM, BSUBSM
Discussion	Routine TDECOM is first called by LINEQS to perform the decomposition of the coefficient matrix A into a lower triangular matrix L and an upper triangular matrix U. The result is then used in BSUBSM for carrying out back substitutions and obtaining the solution to the system of equations. This routine is a modified version of a routine in the subroutine library of the Boeing Computer Services company.

Subroutine	LSQSF
Purpose	To find the generalized inverse from a least squares fit
Input	Common block /LSQSFC/—ZK,WTK,NO,NPK
Output	Common block /LSQSFC/—AK
Subroutines Called	TRANS, MMULT, PDSEQS
Discussion	The routine first forms the weighted normal equations. It then calls routine using the Cholesky scheme to solve the system of equations and finds the generalized inverse. If the system of equations is not positive definite, an error message will be printed and execution of the computer program will be terminated.

Subroutine	MMULT(A,B,C,L,M,N)
Purpose	To multiply two matrices
Input	Calling sequence A—Array containing elements of matrix A B—Array containing elements of matrix B L—Number of rows in A and C M—Number of columns in A and rows in B N—Number of columns in B and C
Output	Calling sequence C—Resultant matrix
Subroutines Called	CMAB
Discussion	MMULT calls CMAB to calculate (C) (A)(B).

Subroutine	PANUNI(ART,RO,Y,X)
Purpose	To transform point coordinates from the local panel system to the universal system
Input	Calling sequence ART—Local to global panel system transformation matrix RO—X,Y,Z coordinates of panel center (universal) Y—X,Y,Z coordinates of point to be transformed (local)
Output	Calling sequence X—X,Y,Z coordinates of transformed point (universal)
Subroutines Called	MMULT
Discussion	The local panel coordinates are multiplied by the matrix ART using subroutine MMULT to produce the global panel coordinates which, when added to the universal panel center, produce the universal coordinates.

Subroutine	PDSEQS(A,NR,N,DN,B,M,D1)
Purpose	To solve a system of equations A*X=B, where A is a positive definite symmetric matrix, using Cholesky decomposition
Input	Calling sequence A—Array of which the upper triangle is the upper triangle of a given positive definite symmetric matrix NR—Maximum row dimension of arrays A and B N—Order of the positive definite coefficient matrix B—Array consists of M right-hand sides of the linear system M—Number of right-hand sides
Output	Calling sequence B—Solution vectors A—Array of which the upper triangle is same as input, the lower triangle contains the lower triangular matrix L from Cholesky decomposition with diagonal elements excluded DN—The reciprocals of diagonal elements of L D1—=1 for successful return =0 indicates that the given coefficient matrix appears not positive definite
Subroutines Called	None
Discussion	The routine first performs the Cholesky decomposition of the given matrix A into a lower triangular matrix L and its transpose. It then solves the given system of equations by back substitutions.

Subroutine	PIDENT(P,Q,IDENT)
Purpose	To determine whether the points P and Q are to be considered numerically identical
Input	Calling sequence P—Coordinates of first point Q—Coordinates of second point
Output	Calling sequence IDENT—Logical variable equal to true if P and Q are considered identical, and false otherwise
Subroutines Called	None
Discussion	The routine determines whether the points P and Q are considered numerically identical. The criterion for identity is that the distance from P to Q must be smaller than or equal to 1.E-12 times the sum of the lengths of P and Q.

<u>Subroutine</u>	<u>PIVC</u>
Purpose	To obtain doublet panel influence coefficients for a given control point
Input	Calling sequence Z—X,Y,Z coordinates of a given control point Common block /PANDQ/—RO,AR,ART,P,A,B,DIAM,C /SYMM/—NSYMM /ZIP/—IPZ,IP
Output	Common block /PIVM/—DVDS
Subroutines Called	UNIPAN, DIPV, MMULT
Discussion	The routine first transfers some of the panel information to be used by the integration routine. It then calls the integration routine to provide influence coefficients for a given control point induced by doublet distribution of the specified panel and its image (when NSYMM is set to 1). The influence coefficients are modified to account for the case when the given control point is located on the influencing panel itself (see Engineering Document—Aerodynamic Influence Coefficients).

<u>Subroutine</u>	<u>PTRNS(IP)</u>
Purpose	To read panel information from disk
Input	Calling sequence IP—Panel number of information to be read Common block /PINDX/—KQ,NPRD
Output	Common block /PANDQ/—CP,PC,RO,AR,ART,P,A,B,DIAM,C,AST,IIS,INS,ITS /PINDX/—KQ
Subroutines Called	None
Discussion	Reads 197 words of panel information from disk file specified by NPRD into common block PANDQ.

<u>Subroutine</u>	<u>SHEGEN(ALPHA,X,S,N,Y,Z)</u>
Purpose	To provide an initial guess of the free and fed sheet geometry at a particular transverse cut

Input	Calling sequence ALPHA—Angle of attack of the wing (in radians) X—X coordinate of transverse cut (APEX is X=0.0) S—Y coordinate of leading edge on transverse cut N—Desired number of free sheet panels in transverse cut Calling sequence
Output	Y—Y coordinate of corner points defining shape of free and fed sheets on given transverse cut Z—Z coordinates of corner points defining shape of free and fed sheets on given transverse cut
Subroutines Called	None
Discussion	The routine computes an initial guess of the free and fed sheet geometry at a particular transverse cut. (See starting solution section of Engineering Document for method. Points describing the curves of figure 17 are stored in the array YZVAL.) Each curve represents the free and fed sheet geometry for one of eight values of A. Points describing the free and fed sheet geometry for an arbitrary value of A are obtained by linear interpolation (or extrapolation). Linear interpolation is then employed on this new set of points to construct a representation of the free sheet by the number of points specified in the input data.

Subroutine	SIDECL(W,DSMIN,D)
Purpose	To compute geometric quantities associated with the relationship of the field point to the quadrilateral Σ for use in computing the H integrals. (See fig. 30 and sec. B.3 of app. B of the Engineering Document.)
Input	Common block /PIVINT/—X,P
Output	Calling sequence W—Point on quadrilateral closest to projection of field point onto quadrilateral plane DSMIN—Minimum distance of projection of field point onto quadrilateral plane to perimeter of quadrilateral D—Distance from W to projection of field point onto quadrilateral plane Common block /SIDEQ/—QSIDE /SKAIC1/—AKS1,AET1,AKS2,AET2,DRM,EL1,EL2,ELM,ANK,ANE, A,AA

Subroutines Called	TRNSFR
Discussion	The routine computes geometric quantities associated with the relationship of the quadrilateral Σ to the projection of the field point onto the quadrilateral plane. In particular, the routine determines whether the projection lies inside or outside of the quadrilateral as well as calculates the minimum distance from the projection to the perimeter of the quadrilateral. Other quantities computed include those quantities displayed in figure 31 and discussed in section B.3 of appendix B of the Engineering Document. The quantities associated with the quadrilateral in general are returned via the call list, whereas the quantities associated with each side of the quadrilateral are stored in a common block array, a side at a time.

<u>Subroutine</u>	<u>SINFCC(Z)</u>
Purpose	Given the X,Y,Z coordinates of a point SINFCC defines a matrix (DSDFS), which when multiplied by a vector consisting of values of all doublet parameters, gives the value and 1st,2nd derivatives of doublet strength at the given point
Input	Calling sequence Z—X,Y,Z coordinates of the given point Common block /INDEX/—NSNGT /PANDQ/—RO,AR,AST,IIS,INS
Output	Common block /SNGC/—DSDFS
Subroutines Called	UNIPAN
Discussion	Subroutine UNIPAN converts the input point from the universal to local panel coordinate system. A six-by-six matrix is formed by the general equation representing the doublet strength distribution at the given point on a panel and its derivatives. A six-by-sixteen matrix (AST) for coefficients of quadratic doublet distribution on the panel also exists. The matrix is computed in subroutine SING. The matrix DSDFS is formed by multiplying these two matrices.

Subroutine	SINGNT.NM.NN,NS.NSA,NPA.ZM)
Purpose	To calculate the singularity distribution defining quantities for a given network
Input	Calling sequence NT—Network type NM—Number of spanwise cuts in the network NN—Number of transverse cuts in the network NSA—Total number of singularity parameters in all previous networks NPA—Total number of panels in all previous networks ZM—Coordinates of corner points in the network Common block /IPRINT/—IPSING /PANDQ/—RO,AR
Output	Calling sequence NS—Number of singularity parameters in the network Common block /PANDQ/—AST,IIS,INS,ITS
Subroutines Called	GCPCAL. GRDIND. PTRNS. UNIPAN, LSQSF. IPTRNS
Discussion	The routine calculates the dependence of each panel singularity strength distribution on the free singularity parameters of the network. Separate computations are performed for each network type. First, the locations of the free singularity parameters on the network are computed and indexed. For each panel, the singularity parameters affecting the distribution of singularity strength on that panel are isolated. Each such parameter is assigned a weight (large if the parameter actually lies on the panel). The panel singularity distribution is then obtained by fitting a quadratic form (if the singularity is of doublet type) to the parameters by the method of least squares. The matrix that relates the coefficients of the distribution to the singularity parameters is then stored on a file along with indices identifying the parameters.

Subroutine	SNGCAL(Z,TSC)
Purpose	To calculate the value and 1st,2nd derivatives of doublet strength at the specified point
Input	Calling sequence Z—X.Y,Z coordinates of the given point Common block /SOLN/—S
Output	Calling sequence TSC—Array consists of the value and 1st.2nd derivatives of doublet strength
Subroutines Called	SINFCC, MMULT

Discussion	SNGCAL calls subroutine SINFCC to produce the matrix DSDFS. MMULT multiplies this matrix by the vector consisting of values of all doublet parameters previously obtained to produce the value and 1st,2nd derivatives of doublet strength at the given point.

<u>Subroutine</u>	<u>SURFIT</u>
Purpose	To define panel surface and local panel coordinate system
Input	Common block /FLATP/—NFLTP /LSQSFC/—ZK,WTK,NO,NPK /PANDQ/—CP /IPRINT/—IPGEOM
Output	Common block /PANDQ/—PC,RO,AR,P,A,B
Subroutines Called	CROSS, UVECT, TRANS, UNIPAN, LSQSF, MMULT
Discussion	The routine defines a panel surface and local panel coordinate system. As a first approximation to the panel surface, the routine takes the quadrilateral formed by projecting the panel corner points onto the plane through the midpoints of the line segments joining these corner points. A local coordinate system is constructed with the origin at the average of the quadrilateral corner points and with one axis normal to the quadrilateral. To obtain a second order approximation to the panel surface, the routine calculates a paraboloid passing through the corner points with curvature obtained by least squaring the paraboloid to adjacent corner points. The local coordinate system is then rotated and translated in such a manner that the paraboloid can be represented in canonical form. An iterative process is required to eliminate linear terms without translating the origin.

<u>Subroutine</u>	<u>SURPRO(Z,ZP,UN)</u>
Purpose	To find the location of the projection of a point onto a panel surface as well as the surface normal at this location
Input	Calling sequence Z—Global coordinates of point to be projected Common block /PANDQ/—RO,AR,ART
Output	Calling sequence ZP—Global coordinates of location of projection UN—Global coordinates of unit normal to panel surface at this location
Subroutines Called	UNIPAN, UVECT, PANUNI, MMULT

Discussion

The routine calculates the projection of a point onto a panel surface as well as the surface normal vector at the projected point. All input and output vectors are assumed to be given in global coordinates. The routine converts to local coordinates, projects and converts back to global coordinates. In the event that the given point does not lie above or below the panel, the projection is made onto the paraboloid of which the panel is a part.

Subroutine: SWEPTE(X,S,N,Y,M,YP)

Purpose: To calculate the Y coordinates of the panel corner points aft of the root chord for swept trailing-edge designs

Input: Calling sequence

X —Array of transverse cut X values starting with the last cut that intersects the root chord

S —Array of Y coordinates of the leading edge on the transverse cuts specified by X

N —Number of transverse cuts aft of the last transverse cut to intersect the root chord plus one

Y —Array of Y coordinates of panel corner points lying on the last transverse cut that intersects the root chord

M —Number of spanwise percent values input by the user

Output: Calling sequence

YP—Array of Y coordinates of panel corner points aft of the root chord

Subroutines Called: None

Discussion

Given the coordinates of two points defining a line and one coordinate of a third point on the line, the unknown coordinate of the third point can be calculated by triangulation.

One of the points defining the line is the leading-edge/trailing-edge intersection point. The other point is the panel corner point lying on the last transverse cut that intersects the root chord.

The X value of the third point is the value of the transverse cut.

Subroutine: TCNTRL

Purpose: To designate the location of control points for all network panels and to compute the unit normal vector and the normal component of freestream velocity vector at every control point

Input: Common block

/INDEX/—NT,NM,NN,NPA,NZA,NNETT

/MSPNTS/—ZM

Output: Common block

/BDYCS/—ZC,ZCC,ZCR,ZDC,IPC,ITC

/INDEX/—NCA,NCTRT

Subroutines Called	CONTRL
Discussion	The routine calls CONTRL to calculate the location of control points for all panels and to compute the unit normal vector and the normal component of freestream velocity vector at every control point on all panels for each network. It also finds the cumulative number of control points and the total number of control points.

<u>Subroutine</u>	<u>TDECOM(A,NR,N,V,IPR,D1)</u>
Purpose	To decompose a square matrix into lower and upper triangular matrices with partial pivoting and row equilibration
Input	Calling sequence A —Array consists of elements of a given matrix NR —Maximum row dimension of array A N —Order of the given matrix V —Scratch array, may be same array as IPR to save storage
Output	Calling sequence A —The lower triangle of the array consists of a lower triangular matrix L, and the upper triangle consists of an upper triangular matrix U (since U is unit upper triangular, its diagonal elements are not stored) IPR—Array gives numbers of pivotal row (a record of interchanges) D1 —=+1 or -1 according to the number of interchanges being even or odd. It also indicates successful decomposition =0 indicates that the given matrix appears singular
Subroutines Called	VIP, VIPS
Discussion	The routine performs the crout factorization of a given matrix with partial pivoting and row equilibration. The upper and lower triangular matrices resulting from the decomposition are stored in the array A which originally consisted of elements of the given matrix. If one of the pivots appears to be too small, D1 is set to zero and an error exit is taken. This routine is a modified version of a routine in the subroutine library of the Boeing Computer Services company.

<u>Subroutine</u>	<u>TGEOMC</u>
Purpose	To generate essential geometry information for each panel of all the networks
Input	Common block /INDEX/—NT,NM,NN,NPA,NZA,NNETT /MSPNTS/—ZM

Output	See output of subroutine GEOMC
Subroutines Called	GEOMC
Discussion	The routine calls GEOMC to calculate essential geometry for all panels of each network.

<u>Subroutine</u>	<u>TRANS(A,AT,M,N)</u>
Purpose	To form the transpose of a matrix A and store the result in a matrix B
Input	Calling sequence A—Array containing matrix elements to be transposed M—Number of rows in A and columns in B N—Number of columns in A and rows in B
Output	Calling sequence AT—Array containing elements of the transpose of the given matrix
Subroutines Called	None
Discussion	AT(J,I) is set to A(I,J) as I varies from 1 to M and J varies from 1 to N.

<u>Subroutine</u>	<u>TRNSFR(X,Y,N)</u>
Purpose	To move a number of elements from one array to another
Input	Calling sequence X—Location of the first array element to be moved N—Number of elements to be moved
Output	Calling sequence Y—Array of elements identical to the first N elements in array X
Subroutines Called	None
Discussion	Y(I) is set to X(I) as I varies from 1 to N.

<u>Subroutine</u>	<u>TSING</u>
Purpose	To designate the location of doublets on all network panels and to compute the matrix for coefficients of quadratic doublets distribution for each panel
Input	Common block /INDEX/—NT,NM,NN,NPA,NZA,NNETT /MSPNTS/—ZM
Output	Common block /INDEX/—NS,NSA,NSNGT

Subroutines Called	SING
Discussion	The routine calls SING to calculate the location of doublets on panels and to compute the matrix for coefficients of quadratic doublet distribution for every panel of each network. It also finds the cumulative number of doublets. Finally, the total number of doublets is obtained.

<u>Subroutine</u>	<u>UNIPAN(AR,RO,X,Y,)</u>
Purpose	To transform point coordinates from the universal system to the local panel system
Input	Calling sequence AR—Global to local panel system transformation matrix RO—X,Y,Z coordinates of panel center (universal) X—X,Y,Z coordinates of point to be transformed (universal)
Output	Calling sequence Y—X,Y,Z coordinates of transformed point (local)
Subroutines Called	MMULT
Discussion	The coordinates of the panel center are subtracted from the coordinates of the point to be transformed. This global array is then multiplied by the matrix AR using subroutine MMULT to produce the local panel coordinates.

<u>Subroutine</u>	<u>UPDATE</u>
Purpose	To update corner points of free sheet, fed sheet, and the part of wake attached to those sheets
Input	Common block /INDEX/—NM,NN,NP,NZ /MSPNTS/—ZM,ZL /SOLN/—ZA
Output	Common block /MSPNTS/—ZM
Subroutines Called	None
Discussion	Corner points are updated using given values of angle and fixed chord length of panels in transverse cut obtained previously in INPUT(OVERLAY-1,0). It is assumed that panel corner points move only in transverse cuts. The routine assumes that NM(3)=2, NN(4)=2, and NM(5)=NN(5)=2.

Subroutine	UVECT(A)
Purpose	To calculate the direction cosines of a vector
Input	Calling sequence A—Direction numbers of a vector
Output	Calling sequence A—Direction cosines of a vector
Subroutines Called	None
Discussion	UVECT performs the following calculations—A(I) / SQRT(A(1)*A(1)+A(2)*A(2)+A(3)*A(3)), where I varies from 1 to 3.

Subroutine	VINFCC
Purpose	To generate the three components of aerodynamic influence coefficients for a given control point induced by all panels doublet distribution
Input	Calling sequence Z—X,Y,Z coordinates of a given control point ZN—Ncrmal vector at the control point on panel surface ZD—Perturbation distance for control point at edges JPC—Index of panel of which components of AIC are to be transformed to its local coordinates Common block /CMO3/—NPIF /INDEX/—NPANT,NSNGT
Output	Common block /PINC/—DVDFS
Subroutines Called	PTRNS, EIVC, PIVC, MMULT
Discussion	For every panel, the routine calls PTRNS to transfer panel information. Depending on the given control point being at the edge or interior of the panel, the routine calls EIVC or PIVC to evaluate the integrals. The latter is then multiplied by the generalized inverse from least squares fit of quadratic doublet distribution obtained in subroutine SING to form the three components of aerodynamic influence coefficients. If JPC is specified, the components of AIC will be transformed to local coordinates of that particular panel.

Subroutine	VIP(A,INCA,B,INCB,N,C) VIPA(A,INCA,B,INCB,N,C) VIPS(A,INCA,B,INCB,N,C)
Purpose	To perform vector inner product calculation (VIP) and to add (VIPA) to or subtract (VIPS) from an incoming value (COMPASS)

Input	Calling sequence A—Vector A INCA—Increment between successive elements of A B—Vector B INCB—Increment between successive elements of B N—Number of elements to be multiplied C—An incoming value to be added to (VIPA) or to be subtracted from (VIPS)
Output	Calling sequence C—Result: C=A·B(VIP), C=C+A·B(VIPA), and C=C-A·B(VIPS)
Subroutines Called	None
Discussion	The inner product of two vectors A and B is calculated and stored in C(VIP). The result is added to (VIPA) or subtracted from (VIPS), an incoming value C, and the sum or the difference is stored back in C. This routine is a modified version of a compass routine in the subroutine library of the Boeing Computer Services company.

Subroutine	ZERO(A,N)
Purpose	To set the elements of an array to zero (COMPASS)
Input	Calling sequence A—Location of first element to be set to zero N—Number of elements to be set to zero
Output	Calling sequence A—Array of zero elements
Subroutines Called	None
Discussion	A(I) is set to zero as I varies from 1 to N.

PROGRAM LISTING

Overlay programs and user library are listed in order as previously shown in the section "Names of Programs and Subroutines."

```
      OVERLAY(VORTEX,0,0)
      PROGRAM TEA378(INPUT=502,OUTPUT,TAPE1,TAPE2,TAPE3,TAPE4,
     1                   TAPE5=INPUT,TAPE6=OUTPUT,TAPE7,TAPE8)
C******
C      PROGRAM   TEA378
C
C      PURPOSE  TO CALL VARIOUS OVERLAYS TO PERFORM THE FOLLOWING TASKS
C               (1) READING THE INPUT DATA AND SETTING UP GEOMETRY DEFINI
C               -TION,
C               (2) GENERATING AIC MATRIX BY AN ADVANCED PANEL-TYPE METHO
C               (3) SOLVING SYSTEM OF EQUATIONS WITH THE GENERATED AIC TO
C               OBTAIN INITIAL DOUBLET DISTRIBUTIONS,
C               AND TO USE THE ROUTINE ITFLOW TO FIND AN ITERATIVE SOLU-
C               TION OF THE FLOW PROBLEM WITH NONLINEAR BOUNDARY CONDI-
C               TIONS.
C
C      SUBROUTINES
C      CALLED   INPUT(OVERLAY-1,0),AICGEN(OVERLAY-2,0),SOLVER(OVERLAY-3,0
C               ),OUTPUT(OVERLAY-4,0),ITFLOW
C
C      DISCUSSION  SEE PROGRAM DOCUMENT 1.3 DESCRIPTION AND FLOW CHART OF
C               OVERLAY PROGRAMS.
C******
      COMMON  /CM03/NTSIN,NTSOUT,NTGD,NPIF,NAIC3,NAIC,NJAC,NSCR
      COMMON/BDYCS/DZC(6,125),ZCF(125),ZDC(125),IZC(125,2)
      COMMON/INDEX/DN(9,7),DNA(10,4),NNETT,NPANT,NSNGT,NCTRT,NZMPT
      COMMON /MSPNTS/ZM(3,175),ZL(75)
      COMMON/FLATP/NFLTP
      COMMON/FSVEL/FSV(3),FSVM,ALPHA,XPITCH,RCHORD
      COMMON/SYMM/NSYMM
      COMMON /NEQS/NE,NR,NMAT,NRHS
      COMMON /NFAJ/NEQ,NF,NG
      COMMON /NITF/NFUN,JT,ITMX,KIT,ITPRIN
      COMMON /SOLN/S(125),ZA(75)
      COMMON /ADR/RTD,DTR
      COMMON /IPRINT/IPNPUT,IPGEOM,IPSING,IPCNTR,IPEIVC,IPOUTP
      DIMENSION X(130),RX(130),DX(130),Y(130),RY(130)
C                              CONSTANTS FOR CONVERTING RADIAN TO DEGREE
C                              AND VICE VERSA
      RTD = 57.29577951  $  DTR = .0174532925
C                              SETS DISK FILE NUMBERS
      NTSIN = 5  $  NTSOUT = 6
      NTGD = 1  $  NPIF = 2
      NAIC3 = 3  $  NAIC = 4
      NJAC = 7  $  NSCR = 8
C                              SETS PRINTING CODES FOR INTERMEDIATED
C                              RESULTS, = 1 FOR PRINTOUT $ = 0 OTHERWISE
      IPNPUT = 0  $  IPOUTP = 0
      IPGEOM = IPSING = IPCNTR = IPEIVC = 0
C                              CALLS OVERLAY(1,0) TO SET UP NETWORK
C                              INDICES AND CORNER POINTS
      CALL OVERLAY(6HVORTEX,1,0)
```

```
C                                  CALLS OVERLAY(2,0) TO GENERATE
C                                  VELOCITY COMPONENTS AND AIC MATRIX
      NFUN = 0
      CALL OVERLAY(6HVORTEX,2,0)
C                                  OBTAINS INITIAL VALUES FOR DOUBLET
C                                  PARAMETERS BY SOLVING SYSTEM OF EQUATIONS
C                                  WITH AIC MATRIX
      REWIND NTGD
      NE = NCTRT   $   NR = 1
      DO 10 JC=1,NCTRT
   10 WRITE(NTGD)   ZCR(JC)
      NMAT = NAIC   $   NRHS = NTGD
      CALL OVERLAY(6HVORTEX,3,0)
      REWIND NRHS
      READ(NRHS)   (S(I),I=1,NSNGT)
      IF(ITMX.NE.0)   GO TO 15
C                                  DISPLAYS INITIAL DOUBLET DISTRIBUTION,
C                                  POSITICON OF FREE SHEET, VELOCITY COMPONENT
C                                  AND DELTA CP WHEN NO ITERATION IS REQUESTED
      NFUN = 100   $   IPOUTP = 1
      CALL OVERLAY(6HVORTEX,4,0)
      GO TO 40
C                              ITERATIVE SOLUTION
   15 CONTINUE
      IPNPUT = 0   $    IPOUTP = 0
      IPGEOM = IPSING = IPCNTR = IPEIVC = 0
C                                  STORES INITIAL GUESSES (DOUBLET PARAMETERS
C                                  EXCLUDING THOSE AT EDGES AND ANGLES)
      DO 20 I=1,NF
   20 Y(I) = S(NEQ+I)
      DO 30 I=1,NG
   30 X(NF+I) = ZA(I)*RTD
      N = NF + NG
      CALL ITFLOW(X,N,RX,DX,Y,PY)
   40 CONTINUE
      END
```

```
      SUBROUTINE AJGEN(X,N)
C******
C     SUBROUTINE AJGEN
C
C     PURPOSE  TO OBTAIN THE ANALYTIC JACOBIAN FOR PERTURBATION VARIABLE
C              -S (DOUBLET PARAMETERS EXCLUDING THOSE AT EDGES AND ANGLE
C              -S) ASSUMING D(AIC)/D(THETA) = 0
C
C     INPUT    CALLING SEQUENCE
C              X - ARRAY OF VALUES FOR THE VARIABLES
C              N - NUMBER OF VARIABLES
C              COMMON BLOCK
C              /INDEX/ - NM,NN,NZ
C              /MSPNTS/ - ZM
C              /ADR/ - DTR
C
C     OUTPUT   COMMON BLOCK
C              /SOLN/ - S,ZA
C
C     SUBROUTINES
C     CALLED   DFGMU,DFGDT
C
C     DISCUSSION  THE ROUTINE STORES VALUES OF DOUBLET PARAMETERS(EXCLUD
C              -ING THOSE AT EDGES) AND ANGLES IN ARRAY S AND ZA RESPEC-
C              TIVELY. ROUTINES ARE CALLED TO GENERATE THE PARTIAL DERI-
C              VATIVES OF FUNCTIONS F AND G WITH RESPECT TO DOUBLET
C              PARAMETERS MU EXCLUDING THOSE AT EDGES (DFGMU) AND ANGLES
C              THETA (DFGDT)
C******
      DIMENSION X(1)
      COMMON/INDEX/NT(9),NM(9),NN(9),NP(9),NS(9),NC(9),NZ(9),
     CNPA(10),NSA(10),NCA(10),NZA(10),NNETT,NPANT,NSNGT,NCTPT,NZMPT
      COMMON /MSPNTS/ZM(3,175),ZL(75)
      COMMON /NFAJ/NEQ,NF,NG
      COMMON /SOLN/S(125),ZA(75)
      COMMON /ADR/RTD,DTR
      DO 10 I=1,NF
   10 S(NEQ+I) = X(I)
      DO 20 I=1,NG
   20 ZA(I) = X(NF+I)*DTR
C                            OBTAINS PARTIAL DERIVATIVES W.R.T. MU
      CALL DFGMU
C                            OBTAINS PARTIAL DERIVATIVES W.R.T. THETA
      NZMP = NZ(1) + 1
      CALL DFGDT(ZM(1,NZMP),NM(2),NN(2))
      RETURN
      END
```

```
      SUBROUTINE DFGDT(ZM,NM,NN)
C******
C     SUBROUTINE DFGDT
C
C     PURPOSE   TO CALCULATE PARTIAL DERIVATIVES OF FUNCTIONS F AND G
C               WITH RESPECT TO PANEL INCLINATION ANGLES OF FREE SHEET
C               ASSUMING D(AIC)/D(THETA) = 0
C
C     INPUT     CALLING SEQUENCE
C               ZM - COORDINATES OF CORNER POINTS OF FREE SHEET NETWORK
C               NM - NUMBER OF SPANWISE CUTS OF NETWORK
C               NN - NUMBER OF TRANSVERSE CUTS OF NETWORK
C               COMMON BLOCK
C               /CM03/ - NSCR
C               /BDYCS/ - ZC
C               /FSVEL/ - FSV
C               /NFAJ/ - NEQ,NF,NG
C               /ADR/ - DTR
C
C     OUTPUT    COMMON BLOCK
C               /CM03/ - NJAC
C
C     SUBROUTINES
C     CALLED    PTRNS,CROSS,UVECT,VIP,UNIPAN,MMULT
C
C     DISCUSSION  A DETAIL DISCUSSION OF THE FORMULA USED IN THE COMPUTA
C               -TION IS GIVEN IN ENGINEERING DOCUMENT (SEE APPENDIX -
C               GEOMETRY UPDATE COEFFICIENTS). THE ROUTINE FIRST FINDS A
C               NORMAL VECTOR N FOR THE PANEL. IT THEN COMPUTES PARTIAL
C               DERIVATIVES OF N WITH RESPECT TO ANGLE THETA, AND FORMS
C               PARTIAL DERIVATIVES OF N.V AND OF PRESSURE JUMP WITH RES-
C               PECT TO THETA. FINALLY IT STORES ALL PARTIAL DERIVATIVES
C               IN PROPER POSITION OF THE JACOBIAN.
C******
      COMMON  /CM03/NTSIN,NTSOUT,NTGD,NPIF,NAIC3,NAIC,NJAC,NSCR
      COMMON/BDYCS/ZC(3,125),ZCC(3,125),ZCR(125),ZDC(125),IZC(125,2)
      COMMON/PANDQ/CP(3,4),PC(3),RC(3),AR(3,3),ART(3,3),P(2,4),DUM(3),
     CC(6,6),AST(6,16),IIS(16),INS,ITS,NPDQ
      COMMON/FSVEL/FSV(3),FSVM
      COMMON /PINDX/KP,KQ,NPWR,NPRD
      COMMON /NFAJ/NEQ,NF,NG
      COMMON /SOLN/S(125),ZA(75)
      COMMON /EEOS/EMUE(2500),EMU(3750),IPR(50)
      COMMON /ADR/RTD,DTR
      DIMENSION ZM(3,NM,NN)
      DIMENSION A(3),B(3),DADT(3),DBDT(3),DN1(3),DN2(3)
      DIMENSION CN(3),W(3),GMU(3),DNDTH(3)
      DIMENSION DNVDTH(50),DCPDTH(50),AJ(130)
      EQUIVALENCE (DNVDTH,EMUE),(DCPDTH,EMUE(101)),(AJ,EMU)
      EQUIVALENCE (X,W(1)),(Y,W(2))
      REWIND NSCR
      REWIND NJAC
```

```
      REWIND NTGD
      NFG = NF + NG
      NFW = NF - NG
C                           PROVIDES ZERO FOR D(AIC)/D(THETA)
C     DO 310 J=1,NG
C 310 AJ(NF+J) = 0.
      CALL ZERO(AJ(NF+1),NG)
      DO 320 I=1,NFW
      READ(NSCR)  (AJ(J),J=1,NF)
      WRITE(NJAC) (AJ(J),J=1,NFG)
  320 CONTINUE
      KQ = 0
      REWIND NPIF  $  NPRD = NPIF
      IPL = NFW
      DADT(1)=DBDT(1)=0.
      NMM1=NM-1
      NNM1=NN-1
      DO 200 MP=1,NNM1
      DO 200 IP=1,NMM1
      ITH=0
      IPL = IPL + 1
      CALL PTRNS(IPL)
C                           CALCULATES CROSS PRODUCT OF VECTORS A AND B
C                           TO FORM AN UNIT NORMAL VECTOR
      DO 65 I=1,3
      ABE=ZM(I,IP,MP)-ZM(I,IP+1,MP+1)
      ABM=ZM(I,IP+1,MP)-ZM(I,IP,MP+1)
      A(I)=ABE+ABM
   65 B(I)=ABE-ABM
      CALL CROSS(A,B,CN)
      SN=SQRT(CN(1)**2+CN(2)**2+CN(3)**2)
      CALL UVECT(CN)
      CALL VIP(CN,1,FSV,1,3,VL3)
      TVL3 = 2.*VL3
C                           CALCUALTES GRAD(MU)
      TSC2 = 0.  $  TSC3 = 0.
      CALL UNIPAN(AP,RC,ZC(1,NEQ+IPL),W)
      DO 90 IC=1,INS
      IS = IIS(IC)
      DX = AST(4,IC)*X + AST(5,IC)*Y
      DY = AST(5,IC)*X + AST(6,IC)*Y
      DSDFS2 = AST(2,IC) + DX
      DSDFS3 = AST(3,IC) + DY
      TSC2 = TSC2 + DSDFS2*S(IS)
      TSC3 = TSC3 + DSDFS3*S(IS)
   90 CONTINUE
      W(1) = TSC2  $  W(2) = TSC3   $   W(3) = 0.
      CALL MMULT(ART,W,GMU,3,3,1)
C                           STARTS TO CALCUALTE D(N.V)/DTHETA AND
C.                          D(V.GRAD(MU))/DTHETA
      DO 100 NP=1,NNM1
      DO 100 JP=1,NMM1
```

```
      ITH=ITH+1
      IF(JP-IP)10,10,30
C                             THETA INBOARD OF PANEL OUTBOARD EDGE
   10 IF(NP-MP+1)30,40,20
C                             THETA AFT OF PANEL L.E.
   20 IF(NP-MP)30,50,30
C                             SETS D(N,V)/DTHETA = 0 AND
C                             D(V.GRAD(MU))/DTHETA = 0
   30 DNVDTH(ITH)=0.
      DCPDTH(ITH) = 0.
      GO TO 100
C                             THETA AT PANEL L.E.
   40 DADT(2)=ZM(3,JP,MP)-ZM(3,JP+1,MP)
      DADT(3)=ZM(2,JP+1,MP)-ZM(2,JP,MP)
      IF(JP-IP)44,45,44
   44 DBDT(2)=DBDT(3)=0.
      DADT(2)=DADT(2)+DADT(2)
      DADT(3)=DADT(3)+DADT(3)
      GO TO 60
   45 DBDT(2) = ZM(3,JP+1,MP) - ZM(3,JP,MP)
      DBDT(3) = ZM(2,JP,MP) - ZM(2,JP+1,MP)
      GO TO 60
C                             THETA AT PANEL T.E.
   50 DADT(2)=ZM(3,JP+1,MP+1)-ZM(3,JP,MP+1)
      DADT(3)=ZM(2,JP,MP+1)-ZM(2,JP+1,MP+1)
      IF(JP-IP)54,55,54
   54 DBDT(2)=DBDT(3)=0.
      DADT(2)=DADT(2)+DADT(2)
      DADT(3)=DADT(3)+DADT(3)
      GO TO 60
   55 DBDT(2)=ZM(3,JP+1,MP+1)-ZM(3,JP,MP+1)
      DBDT(3)=ZM(2,JP,MP+1)-ZM(2,JP+1,MP+1)
   60 CALL CROSS(DADT,B,DN1)
      CALL CROSS(A,DBDT,DN2)
      DO 70 I=1,3
   70 DNDTH(I)=DN1(I)+DN2(I)
      CALL VIP(CN,1,DNDTH,1,3,DNTHL3)
      DO 80 I=1,3
   80 DNDTH(I) = (DNDTH(I) - DNTHL3*CN(I))/SN
C                                     FORMS D(N.V)/DTHETA
      CALL VIP(DNDTH,1,FSV,1,3,DNVDTH(ITH))
      DNVDTH(ITH) = DNVDTH(ITH)*DTR
C                                     CALCULATES D(DELTA CP)/DTHETA
      CALL VIP(GMU,1,DNDTH,1,3,GMUDN)
      DCPDTH(ITH) = -TVL3*GMUDN
      DCPDTH(ITH) = DCPDTH(ITH)*DTR
  100 CONTINUE
      READ(NSCR)  (AJ(J),J=1,NF)
      WRITE(NJAC) (AJ(J),J=1,NF),(DCPDTH(J),J=1,ITH)
      WRITE(NTGD) (DNVDTH(J),J=1,ITH)
  200 CONTINUE
C                             STORES ALL PARTIAL DERIVATIVES IN PROPER
```

```
C                                 POSITION OF THE JACOBIAN
      REWIND NTGD
      DO 400 I=1,NG
      READ(NSCR)  (AJ(J),J=1,NF)
      READ(NTGD)  (DNVDTH(J),J=1,NG)
      WRITE(NJAC) (AJ(J),J=1,NF),(DNVDTH(J),J=1,NG)
  400 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DFGMU
C******
C     SUBROUTINE DFGMU
C
C     PURPOSE   TO CALCULATE PARTIAL DERIVATIVES OF FUNCTIONS F AND G
C               WITH RESPECT TO DOUBLET PARAMETERS (EXCLUDING THOSE AT
C               EDGES)
C
C     INPUT     COMMON BLOCK
C               /CM03/ - NPIF,NAIC3
C               /BDYCS/ - ZC
C               /INDEX/ - NSNGT
C               /FSVEL/ - FSV
C               /NFAJ/ - NEQ,NF,NG
C               /SOLN/ - S
C               /EEQS/ - EMUE,EMU,IPR
C
C     OUTPUT    COMMON BLOCK
C               /CMC3/ - NSCR
C
C     SUBROUTINES
C     CALLED    BSUBSM,PTRNS,MMULT,UNIPAN,VIPS
C
C     DISCUSSION  THE FORMULA AND NOTATION USED HERE ARE DISCUSSED IN DE
C               -TAIL IN ENGINNERING DOCUMENT (SEE APPENDIX - DOUBLET ST-
C               RENGTH UPDATE COEFFICIENTS). THE ROUTINE READS IN DE/EMUE
C               AND DE/DMU AND CALCULATES (DE/DMUE)(-1)*(DE/DMU) WHERE E
C               IS THE FUNCTION CONSISTING OF ONLY THOSE EQUATIONS CORRES
C               -PONDING TO CONTROL POINTS AT EDGES. THEN IT OBTAINS PAR-
C               TIAL DERIVATIVES OF N.V ON WING AND ON FREE SHEET WITH
C               RESPECT TO DOUBLET PARAMETERS. PARTIAL DERIVATIVES OF PRE
C               -SSURE JUMP V.GRAD(MU) WITH RESPECT TO DOUBLET PARAMETER
C               ARE ALSO CALCULATED. FINALLY, PARTIAL DERIVATIVES WITH RE
C               -SPECT TO DOUBLET PARAMETERS EXCLUDING THOSE AT EDGES ARE
C               FORMED.
C******
      COMMON  /CM03/NTSIN,NTSOUT,NTGD,NPIF,NAIC3,NAIC,NJAC,NSCR
      COMMON/BDYCS/ZC(3,125),ZCC(3,125),ZCR(125),ZDC(125),IZC(125,2)
      COMMON/INDEX/DN(9,7),DNA(10,4),NNETT,NPANT,NSNGT,NCTRT,NZMPT
      COMMON/PANDQ/CP(3,4),PC(3),PC(3),AR(3,3),ART(3,3),P(2,4),A,B,DIAM,
     CC(6,6),AST(6,16),IIS(16),INS,ITS,NPDQ
      COMMON/FSVEL/FSV(3),FSVM
      COMMON /PINC/DVDFS(3,125)
      COMMON /PINDX/KP,KQ,NPWR,NPRD
      COMMON /NFAJ/NEQ,NF,NG
      COMMON /SOLN/S(125),ZA(75)
      COMMON /EEQS/EMUE(2500),EMU(3750),IPR(50)
      DIMENSION VL(3),FSVL(3),W(3),DSDFS2(16),DSDFS3(16)
      DIMENSION FGMUE(50),AJ(100)
      EQUIVALENCE (X,W(1)),(Y,W(2)),(FGMUE,IPR),(AJ,EMUE)
C                         CALCULATES DE = (DE/DMUE)(-1)*(DE/DMU)
C                         USING EMUE AND EMU FROM SUBROUTINE FGCAL
```

```
      CALL RSUBSM(EMUE,NEQ,NEQ,IPR,EMU,NF)
      REWIND NJAC
      REWIND NTGD
      KQ = 0
      REWIND NPIF  $  NPFD = NPIF
      REWIND NAIC3
C                           SKIPS FIRST NEQ RECORDS CORRESPONDING TO
C                           CONTROL POINTS AT EDGES
      DO 10 I=1,NEQ
   10 READ(NAIC3)  DVDFS(1)
      NFW = NF - NG
      DO 100 IJ=1,NF
      IP = IJ
      CALL PTRNS(IP)
      READ(NAIC3)  DVDFS
      DO 30 J=1,NSNGT
      CALL MMULT(AR,DVDFS(1,J),W,3,3,1)
      DO 20 I=1,3
   20 DVDFS(I,J) = W(I)
   30 CONTINUE
      IF(IJ.GT.NFW)  GO TO 40
C                           STORES D(N.V) ON WING
      WRITE(NJAC) (DVDFS(3,J),J=1,NSNGT)
      GO TO 100
C                           STORES D(N.V) ON FREE SHEET
   40 CONTINUE
      WRITE(NTGD) (DVDFS(3,J),J=1,NSNGT)
C                           CALCULATES D(V.GRAD(MU)) ON FREE SHEET
      CALL MMULT(DVDFS,S,VL,3,NSNGT,1)
      CALL MMULT(AR,FSV,FSVL,3,3,1)
      DO 50 I=1,3
   50 VL(I) = VL(I) + FSVL(I)
      TSC2 = 0.  $  TSC3 = 0.
      CALL UNIPAN(AR,RO,ZC(1,NEQ+IJ),W)
      DO 60 IC=1,INS
      IS = IIS(IC)
      DX = AST(4,IC)*X + AST(5,IC)*Y
      DY = AST(5,IC)*X + AST(6,IC)*Y
      DSDFS2(IC) = AST(2,IC) + DX
      DSDFS3(IC) = AST(3,IC) + DY
      TSC2 = TSC2 + DSDFS2(IC)*S(IS)
      TSC3 = TSC3 - DSDFS3(IC)*S(IS)
   60 CONTINUE
      DO 70 IS=1,NSNGT
   70 DVDFS(1,IS) = 2.*(TSC2*DVDFS(1,IS) - TSC3*DVDFS(2,IS))
      DO 80 IC=1,INS
      IS = IIS(IC)
   80 DVDFS(1,IS) = DVDFS(1,IS)
     1             + 2.*(VL(1)*DSDFS2(IC) + VL(2)*DSDFS3(IC))
      WRITE(NJAC) (DVDFS(1,J),J=1,NSNGT)
  100 CONTINUE
C                           FORMS (DF/DMU) - (DF/DMUE)*DE
```

```
      REWIND NJAC
      REWIND NSCR
      DO 130 I=1,NF
      READ(NJAC) (FGMUE(J),J=1,NEQ),(AJ(J),J=1,NF)
      DO 120 J=1,NF
      JN1 = (J-1)*NEQ+1
  120 CALL VIPS(FGMUE,1,EMU(JN1),1,NEQ,AJ(J))
      WRITE(NSCR) (AJ(J),J=1,NF)
  130 CONTINUE
C                             FORMS (DG/DMU) - (DG/DMUE)*DE
      REWIND NTGD
      DO 150 I=1,NG
      READ(NTGD) (FGMUE(J),J=1,NEQ),(AJ(J),J=1,NF)
      DO 140 J=1,NF
      JN1 = (J-1)*NEQ+1
  140 CALL VIPS(FGMUE,1,EMU(JN1),1,NEQ,AJ(J))
      WRITE(NSCR) (AJ(J),J=1,NF)
  150 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FGCAL(FVZ,GVZ)
C******
C     SUBROUTINE FGCAL
C
C     PURPOSE  TO SOLVE FOR DOUBLET PARAMETERS AT EDGES AND TO CALCULATE
C              FUNCTIONS F AND G
C
C     INPUT    COMMON BLOCK
C              /CM03/ - NPIF,NAIC3,NAIC
C              /BDYCS/ - ZC,ZCR
C              /FSVEL/ - FSV
C              /NFAJ/ - NEQ,NF,NG
C              /SOLN/ - S
C
C     OUTPUT   CALLING SEQUENCE
C              FVZ - VALUES OF F
C              GVZ - VALUES OF G
C              COMMON BLOCK
C              /EEQS/ - EMUE,EMU,IPR
C
C     SUBROUTINES
C     CALLED   VIPS,LINEQS,PTRNS,MMULT,VIP,UNIPAN
C
C     DISCUSSION  THE ROUTINE READS ROWS OF AIC MATRIX TO FORM COEFFI-
C              CIENTS OF FUNCTION E. THE SOLUTION FOR DOUBLET PARAMETERS
C              (MUE) AT EDGES ARE FOUND BY USING FUNCTION E AND GIVEN
C              VALUES OF ALL OTHER DOUBLET PARAMETERS (MU). SINCE E IS A
C              FUNCTION OF DOUBLET PARAMETERS ONLY, DE/DMUE  AND DE/DMU
C              ARE SIMPLY THE COEFFICIENTS OF E. IF THE MATRIX DE/DMUE
C              IS SINGULAR, AN ERROR MESSAGE WILL BE PRINTED AND THE EX-
C              ECUTION OF THE COMPUTER PROGRAM WILL BE TERMINATED.
C              COMPONENTS OF INFLUENCE COEFFICIENTS ARE READ IN AND MUL-
C              TIPLIED BY VALUES OF DOUBLET PARAMETERS TO FORM PERTURBA-
C              TION VELOCITY. THE LATTER IS ADDED TO FREE STREAM VELO-
C              CITY TO BECOME THE AVERAGE VELOCITY VECTOR V. THE DOT PRO
C              -DUCT N.V IS THEN CALCULATED FOR EVERY INTERIOR CONTROL
C              POINT ON WING (FORMING PART OF FUNCTION F) AND ON FREE
C              SHEET (FORMING FUNCTION G).
C              THE JUMP IN PRESSURE COEFFIEICENTS V.GRAD(MU) (SEE ENGIN-
C              EERING DOCUMENT) ON FREE SHEET IS ALSO CALCULATED (FORM-
C              ING THE OTHER PART OF FUNCTION F).
C******
      DIMENSION FVZ(1),GVZ(1)
      COMMON  /CM03/NTSIN,NTSOUT,NTGD,NPIF,NAIC3,NAIC,NJAC,NSCR
      COMMON/BDYCS/ZC(3,125),ZCC(3,125),ZCR(125),ZDC(125),IZC(125,2)
      COMMON/INDEX/ON(9,7),ONA(10,4),NNETT,NPANT,NSNGT,NCTRT,NZMPT
      COMMON/PANDQ/CP(3,4),PC(3),PO(3),AR(3,3),ART(3,3),P(2,4),A,B,DIAM,
     CC(6,6),AST(6,16),IIS(16),INS,ITS,NPDQ
      COMMON/FSVEL/FSV(3),FSVM
      COMMON /PINC/DVDFS(3,125)
      COMMON /PINDX/KP,KQ,NPWR,NPRD
      COMMON /NFAJ/NEQ,NF,NG
```

```
      COMMON /SOLN/S(125),ZA(75)
      COMMON /EEQS/EMUE(2500),EMU(3750),IPR(50)
      DIMENSION VG(3),VL(3),W(3)
      EQUIVALENCE (X,W(1)),(Y,W(2))
C                             SOLVES FOR EDGE DOUBLETS BY USING
C                             FUNCTION E AND GIVEN VALUES OF ALL
C                             OTHER DOUBLETS
      REWIND NAIC
      DO 30 I=1,NEQ
      IN = I - NEQ
   30 READ(NAIC) (EMUE(J*NEQ+IN),J=1,NEQ),(EMU(J*NEQ+IN),J=1,NF)
      NEQ1 = NEQ + 1
      DO 40 I=1,NEQ
      S(I) = ZCR(I)
   40 CALL VIPS(EMU(I),NEQ,S(NEQ1),1,NF,S(I))
      CALL LINEQS(EMUE,NEQ,NEQ,IPR,S,1,D1)
      IF(D1.NE.0.)  GO TO 50
      PRINT 1001
 1001 FORMAT(/* (DE/DMUE) APPEARS SINGULAR*)
      STOP
   50 CONTINUE
C                             STARTS TO CALCULATE FUNCTIONS F AND G
      KQ = 0
      REWIND NPIF  $  NPPD = NPIF
      REWIND NAIC3
C                             SKIPS FIRST NEQ RECORDS CORRESPONDING TO
C                             CONTROL POINTS AT EDGES
      DO 60 I=1,NEQ
   60 READ(NAIC3)  DVDES(1)
      NEW = NF - NG
      DO 100 IJ=1,NF
      IP = IJ
      CALL PTENS(IP)
      READ(NAIC3)  DVDES
      CALL MMULT(DVDES,S,VG,3,NSNGT,1)
      DO 70 I=1,3
   70 VG(I) = VG(I) + FSV(I)
      IF(IJ.GT.NEW)  GO TO 80
C                             CALCULATES N.V ON WING
      CALL VIP(AR(3),3,VG,1,3,FVZ(IJ))
      GO TO 100
C                             CALCULATES N.V ON FREE SHEET
   80 IG = IJ - NEW
      CALL MMULT(AR,VG,VL,3,3,1)
      GVZ(IG) = VL(3)
C                             CALCULATES V.GRAD(MU) ON FREE SHEET
      TSC2 = 0.  $  TSC3 = 0.
      CALL UNIPAN(AR,PO,ZC(1,NEQ+IJ),W)
      DO 90 IC=1,INS
      IS = IIS(IC)
      DX = AST(4,IC)*X + AST(5,IC)*Y
      DY = AST(5,IC)*X + AST(6,IC)*Y
```

```
      DSDFS2 = AST(2,IC) + DX
      DSDFS3 = AST(3,IC) + DY
      TSC2 = TSC2 + DSDFS2*S(IS)
      TSC3 = TSC3 - DSDFS3*S(IS)
   90 CONTINUE
      FV7(IJ) = 2.*(TSC2*VL(1) - TSC3*VL(2))
  100 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FUNC(X,N,RX)
C******
C     SUBROUTINE FUNC
C
C     PURPOSE   TO EVALUATE FUNCTION F (N.V ON WING AND V.GRAD(MU) ON
C               FREE SHEET) AND G (N.V ON FREE SHEET)
C
C     INPUT     CALLING SEQUENCE
C               X - ARRAY OF VALUES FOR THE VARIABLES
C               N - NUMBER OF VARIABLES
C               COMMON BLOCK
C               /NFAJ/ - NEQ,NF,NG
C               /NITF/ - NFUN
C               /SOLN/ - ZA
C               /ADR/ - DTR
C
C     OUTPUT    CALLING SEQUENCE
C               RX - ARRAY OF VALUES OF FUNCTIONS
C
C     SUBROUTINES
C     CALLED    UPDATE,AICGEN(OVERLAY-2,0),FGCAL
C
C     DISCUSSION  THE ROUTINE STORES VALUES OF DOUBLET PARAMETERS (EX-
C               CLUDING THOSE AT EDGE) AND ANGLES IN ARRAYS S AND ZA
C               RESPECTIVLEY. IT USES NEW ANGLES TO UPDATE THE CORNER
C               POINTS OF FREE SHEET, FED SHEET AND PART OF THE WAKE NET-
C               WORK. AICGEN(OVERLAY-2,0) IS THEN CALLED TO DESIGNATE LO-
C               CATIONS OF DOUBLETS AND CONTROL POINTS AND TO GENERATE
C               VELOCITY COMPONENTS AND AIC MATRIX USING THE UPDATED CORN
C               -ER POINTS. IF PRETURBATION IN ANGLE IS NOT SIGNIFICANT
C               UPDATE AND AICGEN ARE SKIPPED. FINALLY, THE ROUTINE CALLS
C               FGCAL TO CALCULATE VALUES OF FUNCTIONS F AND G.
C******
      DIMENSION X(1),RX(1)
      COMMON /NFAJ/NEQ,NF,NG
      COMMON /NITF/NFUN,JT,ITMX,KIT,ITPRIN
      COMMON /SOLN/S(125),ZA(75)
      COMMON /ADR/RTD,DTR
      DO 10 I=1,NF
   10 S(NEQ+I) = X(I)
      SUM = 0.
      DO 20 I=1,NG
      XDTR = X(NF+I)*DTR
      DZA = ZA(I) - XDTR
      SUM = SUM + DZA*DZA
   20 ZA(I) = XDTR
      IF(NFUN.EQ.0)  GO TO 30
      IF(SUM.LE.1.0E-30)  GO TO 40
C     PRINT 1001, SUM
C1001 FORMAT(/* SUM OF SQUARES OF CHANGES IN ANGLES (RAD.) =*,E14.6)
C
C                              UPDATES CORNER POINTS
```

```
      CALL UPDATE
C                                  OBTAINS VELOCITY COMPONENTS AND AIC MATRIX
   30 CONTINUE
      CALL OVERLAY(6HVORTEX,2,0)
C                                  GETS VALUES OF FUNCTIONS F AND G
   40 CONTINUE
      CALL FGCAL(FX,FX(NF+1))
      RETURN
      END
```

```
      SUBROUTINE ITFLOW(X,N,RX,DX,Y,RY)
C******
C     SUBROUTINE ITFLOW
C
C     PURPOSE   TO PERFORM ITERATIVE SCHEME USING QUASI-NEWTON ALGORITHM
C               FOR THE SOLUTION OF A SET OF NONLINEAR EQUATIONS
C
C     INPUT     CALLING SEQUENCE
C               X - ARRAY OF INITIAL VALUES FOR THE VARIABLES
C               N - NUMBER OF VARIABLES
C               DX,Y,RY - SCRATCH ARRAYS
C               COMMON BLOCK
C               /NFAJ/ - NEQ,NF
C               /NITE/ - ITMX,ITPRIN
C
C     OUTPUT    CALLING SEQUENCE
C               X - ARRAY OF SOLUTION VECTOR
C               RX - ARRAY OF RESIDUAL VECTOR
C
C     SUBROUTINES
C     CALLED    VIP,FUNC,OUTPUT(OVERLAY-4,0),AJGEN,SOLVER(OVERLAY-3,0)
C
C     DISCUSSION  THE ROUTINE CALLS FUNC TO EVALUATE RESIDUALS RX AND
C               CALLS AJGEN TO SET UP THE JACOBIAN AJ. THE SYSTEM OF EQUA
C               -TIONS AJ*DX = -RX IS SOLVED AND A NEW APPROXIMATE SOLU-
C               TION IS FOUND USING CORRECTIONS DX. RESIDUALS AND JACOBI-
C               AN ARE EVALUATED AT THE NEW SOLUTION. THE PROCEDURE IS RE
C               -PEATED UNTIL THE SUM OF SQUARES OF RESIDUALS SATISFIES A
C               PREDETERMINED TOLERANCE TOL OR THE GIVEN MAXIMUM NUMBER
C               OF ITERATIONS ITMX IS REACHED. THE ROUTINE INCLUDES A PRO
C               -CEDURE OF GENERATING NEW AIC AFTER EVERY KIT ITERATIONS.
C               THE JACOBIAN WILL BE CALCULATED BY CALLING AJGEN ONLY
C               WHEN NEW AIC IS GENERATED. OTHERWISE, IT WILL BE UPDATED
C               BY USING A FORMULA OF QUASI-NEWTON SCHEME (SEE ENGINEER-
C               ING DOCUMENT). NUMBER OF ITERATION, SUM OF SQUARES OF RE-
C               SIDUALS AND STEP SIZE ARE PRINTED FOR EVERY ITPRIN ITERA-
C               TIONS. FOR ITERATION STUDY AND CHECK OUT PURPOSE, SOME
C               OTHER INTERMEDIATE PRINT STATEMENTS ARE INCLUDED (SEE  -
C               LISTING).
C******
      DIMENSION X(1),RX(1),DX(1),Y(1),RY(1)
      COMMON  /CMO3/NTSIN,NTSOUT,NTGD,NPIF,NAIC3,NAIC,NJAC,NSCR
      COMMON /NEQS/NE,NP,NMAT,NRHS
      COMMON /NFAJ/NEQ,NF,NG
      COMMON /SOLN/S(125),ZA(75)
      COMMON /NITE/NFUN,JT,ITMX,KIT,ITPRIN
      DIMENSION AJ(130)
C                               SETS PRINTING CODE (FOR ITERATION STUDY)
      IP = 0
C                               SETS NO. OF ITERATIONS TO GENERATE NEW AIC
      KIT = 5
C                               SETS TOLERANCE FOR CONVERGENCE, AND
```

```
C                                   PERCENTAGE FOR NEWTON STEP
      TOL = 1.0E-4
      GAMA = 0.1
C                                   INITIALIZES ITERATION
      TSRX = 1.0E50
      IT = 0
   10 NFUN = 0
      CALL VIP(X,1,X,1,N,SX)
C                                   CALLS FUNC TO EVALUATE RESIDUALS
      CALL FUNC(X,N,RX)
      NFUN = NFUN + 1
      CALL VIP(RX,1,RX,1,N,SRX)
C                                   CHECKS IF STEP SIZE REDUCTION IS NECESSARY
C                                   AND SETS THE APPROPRIATE CODE
      IC = 0
      IF(SRX.LT.TSRX)  IC = 1
      TSRX = SRX
C                                   PRINTS RESULTS FOR EVERY ITPRIN ITERATIONS
      IF(MOD(IT,ITPRIN).NE.0)  GO TO 15
      WRITE(NTSOUT,5010)  IT,SRX
 5010 FORMAT(1H1,* ITERATION NO.*,I4,9X,*SUM OF SQUARES OF RESIDUALS =*,
     $E18.10)
      IF(IT.NE.0)  WRITE(NTSOUT,5020)  SADS
      DO 13 I=1,NF
   13 S(NEQ+I) = X(I)
      CALL OVERLAY(6HVORTEX,4,0)
   15 IF(IP.EQ.0)  GO TO 18
      PRINT 105
  105 FORMAT(1H1)
      PRINT 101, IT,SRX,NFUN
  101 FORMAT(//* ITERATION NO.*,I3,5X,*SUM OF SQUARES OF RESIDUALS =*,E1
     18.10/* NO. OF FUNCTION CALLED =*,I4)
      PRINT 102, (X(I),I=1,N)
  102 FORMAT(/* VALUES OF VARIABLES*/(5E14.6))
      PRINT 103, (RX(I),I=1,N)
  103 FORMAT(/* RESIDUALS*/(5E14.6))
   18 IF(IT.EQ.0)  GO TO 20
      ITO = IT + 1
      IF(ITO.GE.ITMX)  GO TO 110
      IF(SRX.LT.TOL)  GO TO 110
C
C                              TO OBTAIN THE JACOBIAN
   20 CALL AJGEN(X,N)
      DO 100 K=1,KIT
C
C                              TO SOLVE AJ*DX = -RX
      REWIND NTGD
      NE = N  $  NR = 1
      DO 30 I=1,N
      RXN = -RX(I)
   30 WRITE(NTGD) RXN
      NMAT = NJAC  $  NRHS = NTGD
```

```
      CALL OVERLAY(6HVORTEX,3,0)
      REWIND NRHS
      READ(NRHS)  (DX(I),I=1,N)
C     IF(K.EQ.1)  PRINT 107, (DX(I),I=1,N)
C 107 FORMAT(/* CORRECTIONS*/(5E14.6))
      CALL VIP(DX,1,DX,1,N,SDX)
C
C                              DETERMINES THE STEP SIZE
      RXD = SQRT(SX/SDX)
      CALFA = AMIN1(GAMA*RXD,1.)
      FALFA = CALFA
      IH = 0
   40 DO 50 I=1,N
      DX(I) = CALFA*DX(I)
   50 Y(I) = X(I) + DX(I)
C     PRINT 108, FALFA
C 108 FORMAT(/* (FRACTION OF NEWTON STEP TAKEN =*,E14.6,*)*)
C     PRINT 107, (DX(I),I=1,N)
C     PRINT 102, (Y(I),I=1,N)
C
C                              EVALUATES NEW RESIDUALS
      CALL FUNC(Y,N,RY)
      NFUN = NFUN + 1
      CALL VIP(RY,1,RY,1,N,SRY)
      IF(SRY.LT.SRX)  GO TO 60
      IF(IC.EQ.1.AND.K.EQ.1)  GO TO 60
      IH = IH + 1
C     PRINT 106, IH,SRY
C 106 FORMAT(/I5,* CYCLE OF STEP SIZE REDUCTION*/* SUM OF SQUARES OF RES
C    1IDUALS =*,E14.6)
      IF(IH.GE.3)  GO TO 60
      CALFA = 0.5
      FALFA = 0.5*FALFA
      GO TO 40
C
C                              UPDATES THE JACOBIAN
   60 ADS = SDX*FALFA**2
      RADS = 1./ADS
      REWIND NJAC
      REWIND NSCR
      DO 75 I=1,N
      READ(NJAC)  (AJ(J),J=1,N)
      CALL VIP(AJ,1,DX,1,N,TJD)
      RJD = (RY(I) - RX(I) - TJD)*RADS
      DO 70 J=1,N
      AJ(J) = AJ(J) + RJD*DX(J)
   70 CONTINUE
   75 WRITE(NSCR)  (AJ(J),J=1,N)
      NTH=NSCR
      NSCR=NJAC
      NJAC=NTH
C
```

```
C                                RESETS VALUE OF THE VARIABLES AND THE RESIDUALS
      DO 80 I=1,N
      X(I) = Y(I)
   80 RX(I) = RY(I)
      SFX = SRY
      JT = IT + K
C                                  PRINTS RESULTS FOR EVERY ITPRIN ITERATIONS
      SADS = SQRT(ADS)
      IF(K.EQ.KIT)  GO TO 100
      IF(MOD(JT,ITPRIN).NE.0)  GO TO 90
      WRITE(NTSOUT,5010)  JT,SFX
      WRITE(NTSOUT,5020)  SADS
 5020 FORMAT(/*  STEP SIZE (LENGTH OF CORRECTION VECTOR) =*,F14.6)
      DO 85 I=1,NF
   85 S(NEC+I) = X(I)
      CALL OVERLAY(6HVORTEX,4,0)
   90 IF(IP.EQ.0)  GO TO 100
      PRINT 101, JT,SRX,NFUN
      PRINT 104, RXO,FALFA
  104 FORMAT(/* RATIO OF LENGTH OF INITIAL VECTOR TO LENGTH OF FULL NEWT
     1ON STEP =*/E14.6/* FRACTION OF NEWTON STEP TAKEN =*,E14.6)
C     PRINT 102, (X(I),I=1,N)
C     PRINT 103, (RX(I),I=1,N)
  100 CONTINUE
      IT = IT + KIT
      IF(IT.LE.ITMX)  GO TO 10
  110 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE UPDATE
C******
C     SUBROUTINE UPDATE
C
C     PURPOSE   TO UPDATE CORNER POINTS OF FREE SHEET, FED SHEET AND THE
C               PART OF WAKE ATTACHED TO THOSE SHEETS
C
C     INPUT     COMMON BLOCK
C               /INDEX/ - NM,NN,NP,NZ
C               /MSPNTS/ - ZM,ZL
C               /SOLN/ - ZA
C
C     OUTPUT    COMMON BLOCK
C               /MSPNTS/ - ZM
C
C     SUBROUTINES
C     CALLED    NONE
C
C     DISCUSSION  CORNER POINTS ARE UPDATED USING GIVEN VALUES OF ANGLE
C               AND FIXED CHORD LENGTH OF PANELS IN TRANSVERSE CUT OBTAIN
C               -ED PREVIOUSLY IN INPUT(OVERLAY-1,0). IT IS ASSUMED THAT
C               PANEL CORNER POINTS MOVE ONLY IN TRANSVERSE CUTS.
C               THE ROUTINE ASSUMES THAT NM(3)=2, NN(4)=2, AND NM(5)=
C               NN(5)=2.
C******
      COMMON/INDEX/NT(9),NM(9),NN(9),NP(9),NS(9),NC(9),NZ(9),
     CNPA(10),NSA(10),NCA(10),NZA(10),NNETT,NPANT,NSNGT,NCTRT,NZMPT
      COMMON /MSPNTS/ZM(3,175),ZL(75)
      COMMON /SOLN/S(125),ZA(75)
C                               UPDATES CORNER POINTS OF FREE SHEET
      NM2 = NM(2)   $   NN2 = NN(2)
      NZM = NZ(1) + NM2
      DO 100 J=2,NN2
      J2 = J-2
      JA = J2*(NM2 - 1)
      JM = NZM + J2*NM2
      DO 100 I=2,NM2
      I1 = I-1
      IA = JA + I1
      IM = JM + I1
      ZLP = ZL(IA)   $   ZAP = ZA(IA)
      ZM(2,IM+1) = ZM(2,IM) + ZLP*COS(ZAP)
      ZM(3,IM+1) = ZM(3,IM) + ZLP*SIN(ZAP)
  100 CONTINUE
C                               UPDATES CORNER POINTS OF FED SHEET
      NZ1 = NZ(1)   $   NZ2 = NZ1 + NZ(2)
      NP2 = NP(2)   $   NP3 = NP(3)
      DO 200 I=1,NP3
      IM = NZM + I*NM2
      IMN = NZ2 + (2*I+1)
      ZM(2,IMN) = ZM(2,IM)
      ZM(3,IMN) = ZM(3,IM)
```

```
      IA = NP2 + I
      ZLP = ZL(IA)   $  ZAP = ZA(IA)
      ZM(2,IMN+1) = ZM(2,IMN) + ZLP*COS(ZAP)
      ZM(3,IMN+1) = ZM(3,IMN) + ZLP*SIN(ZAP)
  200 CONTINUE
C                          UPDATES FORZEN WAKE ATTACHED TO FREE SHEET
      NM3 = NM(3)   $   NM4 = NM(4)
      NZ3 = NZ2 + NZ(3)   $   NZ4 = NZ3 + NZ(4)
      NZMA = NZ2 - NM2
      NZMB = NZ3 + NM4 - NM2
      DO 300 I=2,NM2
      IM = NZMA + I
      IMN = NZMB + I
      ZM(2,IMN+NM4) = ZM(2,IMN) = ZM(2,IM)
      ZM(3,IMN+NM4) = ZM(3,IMN) = ZM(3,IM)
  300 CONTINUE
C                          UPDATES FROZEN WAKE ATTACHED TO FED SHEET
      NM5 = NM(5)
      NZMC = NZ3 - NM3
      DO 400 I=1,NM5
      IM = NZMC + I
      IMN = NZ4 + I
      ZM(2,IMN+2) = ZM(2,IMN) = ZM(2,IM)
      ZM(3,IMN+2) = ZM(3,IMN) = ZM(3,IM)
  400 CONTINUE
      RETURN
      END
```

```
      OVERLAY(VORTEX,1,0)
      PROGRAM INPUT
C******
C     PROGRAM INPUT
C
C     PURPOSE  READ AND ECHO USER INPUT DATA
C              CALCULATE FREE STREAM VELOCITY
C              CALCULATE COORDINATES OF ALL PANEL CORNER POINTS
C              CALCULATE INITIAL LENGTH AND ANGLE OF PANELS ON
C                THE FREE VORTEX AND FED SHEET
C
C     INPUT    DATA CARDS (SEE ENGINEERING DOCUMENT - USER GUIDE)
C
C     OUTPUT   COMMON BLOCK
C              /DAT3/ - AR,NTR,XTR,MSP,YSP,NTC,NLE,YLE,NTE,YTE,MSF
C              /FSVEL/ - FSV,FSVM,ALPHA,XPITCH,RCHORD
C              /INDEX/ - NT,NM,NN,NP,NZ,NPA,NZA,NNETT,NPANT,NZMPT
C              /MSPNTS/ - ZM,ZL
C              /SOLN/ - ZA
C
C     SUBROUTINES
C     CALLED   SHEGEN,DWNET,AWNET,GWNET
C
C     DISCUSSION  SEE PROGRAM DOCUMENT 1.3 DESCRIPTION AND FLOW CHART OF
C              OVERLAY PROGRAMS.
C******
      COMMON  /CMO3/NTSIN,NTSOUT,NTGD,NPIF,NAIC3,NAIC,NJAC,NSCF
      COMMON/INDEX/NT(9),NM(9),NN(9),NP(9),NS(9),NC(9),NZ(9),
     CNPA(10),NSA(10),NCA(10),NZA(10),NNETT,NPANT,NSNGT,NCTRT,NZMPT
      COMMON/MSPNTS/ZM(3,175),ZL(75)
      COMMON /SOLN/S(125),ZA(75)
      COMMON/FLATP/NFLTP
      COMMON/FSVEL/FSV(3),FSVM,ALPHA,XPITCH,RCHORD
      COMMON/SYMM/NSYMM
      COMMON /ADR/RTD,DTR
      COMMON /NITF/NFUN,JT,ITMX,KIT,ITPRIN
      COMMON /DAT3/AR,NTR,XTR(10),MSP,YSP(10),NTC,NLE,YLE(10),
     1           NTE,YTE(10),MFS
      COMMON /IPRINT/IPNPUT,IPGEOM,IPSING,IPCNTR,IPEIVC,IPOUTP
      DIMENSION  ILE(10),ITE(10),YF(15),ZF(15)
      DIMENSION  IDICT(14),ICARD(20)
      DATA  NDICT/14/
      DATA  IDICT/4H$ALP,4H$ASP,4H$TRA,4H$SPA,4H$CEN,4H$DEL,4H$ARF,
     1            4H$GOT,4H$INP,4H$FRE,4H$PIT,4H$ITE,4H$PRI,4H$END/
C                       SETS NSYMM = 1 FOR AXISYMMETRIC
C                            NSYMM = 0 OTHERWISE
      NSYMM = 1
C                           SETS NFLTP = 1 FOR FLAT PANEL
C                                NFLTP = 0 FOR CURVED PANEL
      NFLTP = 1
C
C                       PRINTS TITLE AND DATA CARDS
```

```
      WRITE(NTSOUT,5010)
 5010 FORMAT(1H1//57X,*A COMPUTER PROGRAM*/65X,*FOR*/42X,*A THREE DIMENS
     1IONAL SOLUTION OF FLOWS OVER WINGS*/48X,*WITH LEADING EDGE VORTEX
     2SEPARATION*////53X,*- LIST OF INPUT DATA CARDS -*)
      NCARD = 0  $  LCPR = 42
      IDEND = IDICT(14)
   10 WRITE(NTSOUT,5020)
 5020 FORMAT(//34X,*NO.*,4X,*CARD IMAGES*//)
   20 READ(NTSIN,5030)  ICARD
 5030 FORMAT(20A4)
      IF(EOF,NTSIN)  40,30
   30 NCARD = NCARD + 1
      WRITE(NTSOUT,5040)  NCARD,ICARD
 5040 FORMAT(30X,I6,5X,20A4)
      IF(ICARD.EQ.IDEND)  GO TO 40
      IF(MOD(NCARD,LCPR).NE.0)  GO TO 20
      WRITE(NTSOUT,5050)
 5050 FORMAT(1H1)
      GO TO 10
   40 NBSP = NCARD-3
      DO 50 I=1,NBSP
      BACKSPACE NTSIN
   50 CONTINUE
C                             READS INPUT VARIABLES
   60 READ(NTSIN,5030)  ICARD
      IF(EOF,NTSIN)  65,70
   65 WRITE(NTSIN,5055)
 5055 FORMAT(//* - END OF FILE ENCOUNTERED - */* - END CARD ASSUMED PROC
     1ESSING WILL CONTINUE*)
      GO TO 220
   70 DO 80 IGO=1,NDICT
      IF(ICARD.EQ.IDICT(IGO))  GO TO 90
   80 CONTINUE
      WRITE(NTSOUT,5060)  ICARD
 5060 FORMAT(///* - THE FOLLOWING INPUT DATA CARD DOES NOT MATCH ANY DES
     1IGNATED KEYWORD -*//2X,20A4)
      STOP
   90 GOTO (100,110,120,130,140,150,160,170,180,190,195,200,210,220),IGO
C                             READS ANGLE OF ATTACK IN DEGREES
  100 READ(NTSIN,5070)  ALPHAD
 5070 FORMAT(6E10.0)
      ALPHA = ALPHAD*DTR
      GO TO 60
C                             READS ASPECT RATIO
  110 READ(NTSIN,5070)  AR
      GO TO 60
C                             READS NO. AND X COORD. OF TRANSVERSE CUTS
  120 READ(NTSIN,5070)  TRAN
      NTR = TRAN
      READ(NTSIN,5070)  (XTR(I),I=1,NTR)
      GO TO 60
C                             READS NO. AND PERCENT VALUES OF SPANWISE CUTS
```

```
  130 READ(NTSIN,5070)   SPAN
      MSP = SPAN
      READ(NTSIN,5070)   (YSP(I),I=1,MSP)
      GO TO 60
C                          READS NO. OF TRANSVERSE CUTS ALONG CENTERLINE
  140 READ(NTSIN,5070)   CTRA
      NTC = CTRA
      GO TO 60
C                          SETS CODE FOR DELTA WING PREPROCESSOR
  150 READ(NTSIN,5070)   DUMMY
      KWPR = 1
      GO TO 60
C                          SETS CODE FOR ARROW WING PREPROCESSOR
  160 READ(NTSIN,5070)   DUMMY
      KWPR = 2
      GO TO 60
C                          READS Y VALUES OF LEADING EDGE CORNER POINTS
C                          AND SETS CODE FOR GOTHIC WING PREPROCESSOR
  170 READ(NTSIN,5070)   (YLE(I),I=1,NTR)
      KWPR = 3
      GO TO 60
C                          READS INPUT CORNER POINTS AND INDICES FOR
C                          GENERAL TYPE OF WING NETWORK
  180 READ(NTSIN,5070)   FNZ
      NZW = FNZ
      IF(NZW.EQ.MSP*NTR) GO TO 182
      WRITE(NTSOUT,5080)
 5080 FORMAT(//* NO. OF INPUT CORNER POINTS FOR WING NETWORK IS NOT EQUA
     1L TO*/* THE PRODUCT OF NO. OF TRANSVERSE CUTS AND NO. OF SPANWISE
     2CUTS*)
      STOP
  182 DO 184 J=1,NTR
      JN = (J-1)*MSP
      XJ = XTR(J)
      DO 183 I=1,MSP
  183 ZM(1,JN+I) = XJ
      READ(NTSIN,5070)   (ZM(2,JN+I),ZM(3,JN+I),I=1,MSP)
  184 CONTINUE
      READ(NTSIN,5070)   FNLE
      NLE = FNLE
      READ(NTSIN,5070)   (YLE(I),I=1,NLE)
      DO 186 I=1,NLE
      ILE(I) = K = YLE(I)
      YLE(I) = ZM(2,K)
  186 CONTINUE
      READ(NTSIN,5070)   FNTE
      NTE = FNTE
      READ(NTSIN,5070)   (YTE(I),I=1,NTE)
      DO 188 I=1,NTE
  188 ITE(I) = YTE(I)
      KWPR = 0
      GO TO 60
```

```
C                                READS NO. OF SPANWISE CUTS FOR THE FREE SHEET
C                                NETWORK
  190 READ(NTSIN,5070)   SFS
      MFS = SFS
      GO TO 60
C                                   READS X VALUE OF PITCH AXIS
  195 READ(NTSIN,5070)   XPITCH
      GO TO 60
C                                READS MAX. NO. OF ITERATIONS ALLOWED FOR THE
C                                NONLINEAR EQUATIONS SOLVER
  200 READ(NTSIN,5070)   TMX
      ITMX = TMX
      GO TO 60
C                                READS PRINTING OPTION
  210 READ(NTSIN,5070)   PRINT
      ITPRIN = PRINT
      IF(ITPRIN.EQ.0)  ITPRIN = 5
      GO TO 60
C
C                                CALCULATES FREE STREAM VELOCITY AND ROOT CHORD
  220 CONTINUE
      FSV(1) = COS(ALPHA)
      FSV(2) = 0.
      FSV(3) = SIN(ALPHA)
      FSVM=SQRT(FSV(1)**2+FSV(2)**2+FSV(3)**2)
      RCHORD = XTR(NTC)
      IF(KWPR.EQ.0)  GO TO 260
C
C                                USES PREPROCESSOR TO GENERATE CORNER POINTS FOR
C                                WING NETWORK
      GO TO (230,240,250), KWPR
C                                   CALLS DELTA WING PREPROCESSOR
  230 CALL DWNET
      NLE = NTR   $  NTE = MSP
      NMZ = (NTR-1)*MSP
      DO 235 I=1,NTE
      ITE(I) = NMZ + I
  235 CONTINUE
      GO TO 260
C                                   CALLS ARROW WING PREPROCESSOR
  240 CALL AWNET
      NLE = NTR   $  NTE = NTR-NTC+1
      NMZ = (NTC-1)*MSP + 1
      DO 245 I=1,NTE
      ITE(I) = NMZ + (I-1)*MSP
  245 CONTINUE
      GO TO 260
C                                   CALLS GOTHIC WING PREPROCESSOR
  250 CALL GWNET
      NLE = NTR   $  NTE = NTR-NTC+1
      NMZ = (NTC-1)*MSP + 1
      DO 255 I=1,NTE
```

```
      ITE(I) = NMZ + (I-1)*MSP
  255 CONTINUE
C
C                               SETS UP NETWORKS INDICES
  260 NM(1) = MSP   $   NN(1) = MAX0(NTC,NLE)
      NM(2) = MES   $   NN(2) = NLE
      NM(3) = 2     $   NN(3) = NLE
      NM(4) = NTE+NM(2)-1   $   NN(4) = 2
      NM(5) = 2     $   NN(5) = 2
C
C                               DESIGNATES NETWORK TYPE
      NNETT = 5
      NT(1) = 2   $   NT(2) = 4   $   NT(3) = 6
      NT(4) = 5   $   NT(5) = 7
C
C                               CALCULATES OTHER NETWORK DATA
      NPA(1) = 0   $   NZA(1) = 0
      DO 270 K=1,NNETT
      NP(K) = (NM(K)-1)*(NN(K)-1)
      NZ(K) = NM(K)*NN(K)
      NPA(K+1) = NPA(K) + NP(K)
      NZA(K+1) = NZA(K) + NZ(K)
  270 CONTINUE
      NZ(NNETT+1) = 0
      NPANT = NPA(NNETT+1)
      NZMPT = NZA(NNETT+1)
      NZ1 = NZA(2)   $   NZ2 = NZA(3)   $   NZ3 = NZA(4)
      NZ4 = NZA(5)   $   NZ5 = NZA(6)
      M1 = NM(1)   $   M2 = NM(2)   $   M3 = NM(3)
      M4 = NM(4)   $   M5 = NM(5)
C
C                               CALCULATES CORNER POINTS FOR FREE SHEET, FED
C                               SHEET AND THE ATTACHED FROZEN WAKE
      NN1 = NN(1)
      XWAKE = 50.*XTR(NN1)
      IF(XWAKE.LE.100.0)  XWAKE = 100.
C                                   FOR NETWORK NO. 2
      X1 = ZM(1,M1)   $   Y1 = ZM(2,M1)   $   Z1 = ZM(3,M1)
      DO 280 I=1,M2
      ZM(1,NZ1+I) = X1
      ZM(2,NZ1+I) = Y1
      ZM(3,NZ1+I) = Z1
  280 CONTINUE
C                                   FOR NETWORK NO. 3
      X1 = ZM(1,NZ1+M2)   $   Y1 = ZM(2,NZ1+M2)   $   Z1 = ZM(3,NZ1+M2)
      DO 290 I=1,M3
      ZM(1,NZ2+I) = X1
      ZM(2,NZ2+I) = Y1
      ZM(3,NZ2+I) = Z1
  290 CONTINUE
C                                   FOR NETWORK NO. 2
      NMS = M2 - 1
```

```
      DO 310 K=2,NLE
      K1 = K-1
      NMK = NZ1 + K1*M2
      CALL SHEGEN(ALPHA,XTR(K),YLE(K),NMS,YF,ZF)
      XK = XTR(K)
      ZLEK = ZM(3,K*M1)
      DO 300 J=1,M2
      JM = NMK + J
      ZM(1,JM) = XK
      ZM(2,JM) = YF(J)   $   ZM(3,JM) = ZF(J) + ZLEK
  300 CONTINUE
C                                   FOR NETWORK NO. 3
      NMZ = NZ2 + K1*M3
      ZM(1,NMZ+1) = ZM(1,NMZ+2) = XK
      ZM(2,NMZ+1) = YF(M2)   $   ZM(3,NMZ+1) = ZF(M2) + ZLEK
      ZM(2,NMZ+2) = YF(M2+1)   $   ZM(3,NMZ+2) = ZF(M2+1) + ZLEK
  310 CONTINUE
C                                   FOR NETWORK NO. 4
      NZMA = NZ2 - M2   $   NZMB = NZ3 + NTE - 1
      DO 320 I=2,M2
      IM = NZMA + I   $   IMN = NZMB + I
      ZM(1,IMN) = ZM(1,IM)   $   ZM(1,IMN+M4) = XWAKE
      ZM(2,IMN+M4) = ZM(2,IMN) = ZM(2,IM)
      ZM(3,IMN+M4) = ZM(3,IMN) = ZM(3,IM)
  320 CONTINUE
C                                   FOR NETWORK NO. 5
      NZMC = NZ3 - M3
      DO 330 I=1,M5
      IM = NZMC + I   $   IMN = NZ4 + I
      ZM(1,IMN) = ZM(1,IM)   $   ZM(1,IMN+2) = XWAKE
      ZM(2,IMN+2) = ZM(2,IMN) = ZM(2,IM)
      ZM(3,IMN+2) = ZM(3,IMN) = ZM(3,IM)
  330 CONTINUE
C
C                             MATCHES LEADING EDGE CORNER POINTS
      NMZ = NZ1 + 1
      DO 340 I=2,NLE
      IM = I*M1   $   IMN = NMZ + (I-1)*M2
      ZM(2,IM) = ZM(2,IMN)
      ZM(3,IM) = ZM(3,IMN)
  340 CONTINUE
C
C                             SETS UP CORNER POINTS FOR THE PART OF
C                             THE WAKE NETWORK ATTACHED TO WING
      DO 350 I=1,NTE
      IM = ITE(I)   $   IMN = NZ3 + I
      ZM(1,IMN) = ZM(1,IM)   $   ZM(1,IMN+M4) = XWAKE
      ZM(2,IMN+M4) = ZM(2,IMN) = ZM(2,IM)
      ZM(3,IMN+M4) = ZM(3,IMN) = ZM(3,IM)
  350 CONTINUE
C
C                             PRINTS OUT CORNER POINTS
```

```
      IF(IPNPUT.EQ.0)   GO TO 1040
      PRINT 1000
 1000 FORMAT(*1CHECK TEST PROBLEM DATA*)
      DO 1030 I=1,NNETT
      J1 = NZA(I) + 1   $   J2 = NZA(I+1)
      PRINT 1010, I
 1010 FORMAT(//* NETWORK NO.*,I3)
      PRINT 1020, (ZM(1,J),ZM(2,J),ZM(3,J),J=J1,J2)
 1020 FORMAT(12F10.5)
 1030 CONTINUE
 1040 CONTINUE
C
C                         CALCULATES INITIAL LENGTH AND ANGLE OF SPAN-
C                         WISE SECTION OF PANELS FOR FREE AND FED SHEETS
C                         ASSUMING NM(3) = 2
      IF(IPNPUT.NE.0)   PRINT 1050
 1050 FORMAT(*1 ANGLE AND LENGTH*/)
      NM2 = NM(2)   $   NN2 = NN(2)
      NZM = NZ(1) + NM(2)
      DO 360 J=2,NN2
      J2 = J-2
      JA = J2*(NM2 - 1)
      JM = NZM + J2*NM2
      DO 360 I=2,NM2
      I1 = I-1
      IA = JA + I1
      IM = JM + I1
      Y1 = ZM(2,IM)   $   Y2 = ZM(2,IM+1)
      Z1 = ZM(3,IM)   $   Z2 = ZM(3,IM+1)
      DY = Y2 - Y1    $   DZ = Z2 - Z1
      ZLP = SQRT(DY**2 + DZ**2)
      ZAP = ATAN2(DZ,DY)
      ZL(IA) = ZLP
      ZA(IA) = ZAP
      IF(IPNPUT.NE.0)   PRINT 1060, IA,IM,ZAP,ZLP,Y1,Z1,Y2,Z2,DY,DZ
 1060 FORMAT(2I5,2E12.4,6F8.3)
  360 CONTINUE
      NP2 = NP(2)   $   NP3 = NP(3)
      DO 370 I=1,NP3
      IA = NP2 + I
      IM = NZ2 + (2*I+1)
      Y1 = ZM(2,IM)   $   Y2 = ZM(2,IM+1)
      Z1 = ZM(3,IM)   $   Z2 = ZM(3,IM+1)
      DY = Y2 - Y1    $   DZ = Z2 - Z1
      ZLP = SQRT(DY**2 + DZ**2)
      ZAP = ATAN2(DZ,DY)
      ZL(IA) = ZLP
      ZA(IA) = ZAP
      IF(IPNPUT.NE.0)   PRINT 1060, IA,IM,ZAP,ZLP,Y1,Z1,Y2,Z2,DY,DZ
  370 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE AWNET
C******
C     SUBROUTINE AWNET
C
C     PURPOSE  TO CALCULATE THE COORDINATES OF ALL PANEL CORNER POINTS
C              IN AN ARROW WING PLANFORM CONFIGURATION
C
C     INPUT    COMMON BLOCK
C              /DAT3/ - AR,NTR,XTR,MSP,YSP,NTC
C
C     OUTPUT   COMMON BLOCK
C              /DAT3/ - YLE,NTE
C              /MSPNTS/ - ZM
C
C     SUBROUTINES
C     CALLED   SWEPTE
C
C     DISCUSSION  THE Y COORDINATES OF THE PANEL CORNER POINTS AT THE
C              INTERSECTION OF THE LEADING EDGE AND TRANSVERSE CUTS ARE
C              COMPUTES BY MULTIPLYING THE X VALUE OF THE TRANSVERSE CUT
C              BY ONE-FOURTH THE ASPECT RATIO.
C                 THE Y COORDINATES OF THE PANEL CORNER POINTS BETWEEN
C              THE LEADING EDGE AND ROOT CHORD ON THE TRANSVERSE CUTS
C              ARE COMPUTED BY MULTIPLYING THE Y COORDINATE AT THE
C              LEADING EDGE BY THE ARRAY OF PERCENT VALUES YSP.
C                 SUBROUTINE SWEPTE IS CALLED TO CALCULATE THE Y
C              COORDINATES OF ALL PANEL POINTS AFT OF THE ROOT CHORD.
C                 THE X COORDINATES OF THE PANEL CORNER POINTS ARE THE
C              X VALUES OF THE TRANSVERSE CUTS INPUT BY THE USER. ALL Z
C              COORDINATES ARE SET TO ZERO.
C******
      COMMON/MSPNTS/ZM(3,175),ZL(75)
      COMMON /DAT3/AR,NTR,XTR(10),MSP,YSP(10),NTC,NLE,YLE(10),
     1            NTE,YTE(10),MES
      DIMENSION  YW(10),XY(10,10)
C                          FINDS D (= S/X)
      D = AR/4.
C                          OBTAINS CORNEN POINTS COORD. FOR THE UPPER PART
      X1 = XTR(1)
      YLE(1) = D*X1
      DO 10 I=1,MSP
      ZM(1,I) = X1
      ZM(2,I) = ZM(3,I) = 0.
   10 CONTINUE
      DO 30 J=2,NTR
      JN = (J-1)*MSP
      XJ = XTR(J)
      SEMI = D*XJ
      YLE(J) = SEMI
      DO 20 I=1,MSP
      ZM(1,JN+I) = XJ
      ZM(2,JN+I) = SEMI*YSP(I)
```

```
   20 ZM(3,JN+I) = 0.
   30 CONTINUE
C                          OBTAINS CORNER POINTS COORD. FOR THE LOWER PART
C                          WITH SWEPT TRAILING EDGE
      KT1 = NTC-1
      NTE = NTR - KT1
      DO 40 I=1,NTE
      YLE(KT1+I) = D*XTR(KT1+I)
   40 CONTINUE
      NMZ = KT1*MSP
      DO 50 J=1,MSP
      YW(J) = ZM(2,NMZ+J)
   50 CONTINUE
      CALL SWEPTE(XTR(NTC),YLE(NTC),NTE,YW,MSP,XY)
      NTE1 = NTE-1
      DO 60 J=1,NTE1
      JM = NMZ + J*MSP
      XJ = XTR(NTC+J)
      DO 60 I=1,MSP
      IM = JM + I
      ZM(1,IM) = XJ
      ZM(2,IM) = XY(I,J)
      ZM(3,IM) = 0.
   60 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DWNET
C******
C     SUBROUTINE DWNET
C
C     PURPOSE  TO CALCULATE THE COORDINATES OF ALL PANEL CORNER POINTS
C              IN A DELTA WING PLANFORM CONFIGURATION
C
C     INPUT    COMMON BLOCK
C              /DAT3/ - AR,NTR,XTR,MSP
C
C     OUTPUT   COMMON BLOCK
C              /DAT3/ - YLE
C              /MSPNTS/ - ZM
C
C     SUBROUTINES
C     CALLED   NONE
C
C     DISCUSSION  THE Y COORDINATES OF PANEL CORNER POINTS AT THE
C              INTERSECTION OF THE LEADING EDGE AND THE TRANSVERSE CUTS
C              ARE COMPUTED BY MULTIPLYING THE X VALUE OF THE TRANSVERSE
C              CUT BY ONE FOURTH THE ASPECT RATIO.
C                 THE Y COORDINATES OF THE PANEL CORNER POINTS BETWEEN
C              THE LEADING EDGE AND ROOT CHORD ON THE TRANSVERSE CUTS
C              ARE COMPUTED BY MULTIPLYING THE Y COORDINATE AT THE
C              LEADING EDGE BY THE ARRAY OF PERCENT VALUES YSP.
C                 THE X COORDINATES OF THE PANEL CORNER POINTS ARE THE
C              X VALUES OF THE TRANSVERSE CUTS INPUT BY THE USER. ALL
C              Z COORDINATES ARE SET TO ZERO.
C******
      COMMON/MSPNTS/ZM(3,175),ZL(75)
      COMMON /DAT3/AR,NTR,XTR(10),MSP,YSP(10),NTC,NLE,YLE(10),
     1              NTE,YTE(10),MES
C                           FINDS D (= S/X)
      D = AR/4.
C                           OBTAINS CORNEN POINTS COORD.
      X1 = XTR(1)
      YLE(1) = D*X1
      DO 10 I=1,MSP
      ZM(1,I) = X1
      ZM(2,I) = ZM(3,I) = 0.
   10 CONTINUE
      DO 30 J=2,NTR
      JN = (J-1)*MSP
      XJ = XTR(J)
      SEMI = D*XJ
      YLE(J) =SEMI
      DO 20 I=1,MSP
      ZM(1,JN+I) = XJ
      ZM(2,JN+I) = SEMI*YSP(I)
   20 ZM(3,JN+I) = 0.
   30 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE GWNET
C******
C     SUBROUTINE GWNET
C
C     PURPOSE   TO CALCULATE THE COORDINATES OF ALL PANEL CORNER POINTS
C               IN A GOTHIC WING PLANFORM CONFIGURATION
C
C     INPUT     COMMON BLOCK
C               /DAT3/ - NTR,XTR,MSP,YSP,NTC,YLE
C
C     OUTPUT    COMMON BLOCK
C               /DAT3/ - NTE
C               /MSPNTS/ - ZM
C
C     SUBROUTINES
C     CALLED    SWEPTE
C
C     DISCUSSION  THE Y COORDINATES OF PANEL CORNER POINTS AT THE
C               INTERSECTION OF THE LEADING EDGE AND TRANSVERSE CUTS ARE
C               INPUT BY THE USER.
C                  THE Y COORDINATES OF PANEL CORNER POINTS BETWEEN THE
C               LEADING EDGE AND ROOT CHORD ON THE TRANSVERSE CUTS ARE
C               COMPUTED BY MULTIPLYING THE Y COORDINATE AT THE LEADING
C               EDGE BY THE ARRAY OF PERCENT VALUES YSP.
C                  SUBROUTINE SWEPTE IS CALLED TO CALCULATE THE Y
C               COORDINATES OF ALL PANEL POINTS AFT OF THE ROOT CHORD.
C                  THE X COORDINATES OF THE PANEL CORNER POINTS ARE THE
C               X VALUES OF THE TRANSVERSE CUTS INPUT BY THE USER. ALL
C               Z COORDINATES ARE SET TO ZERO.
C******
      COMMON/MSPNTS/ZM(3,175),ZL(75)
      COMMON /DAT3/AR,NTR,XTR(10),MSP,YSP(10),NTC,NLE,YLE(10),
     1              NTE,YTE(10),MES
      DIMENSION  YW(10),XY(10,10)
C                           OBTAINS CORNEN POINTS COORD. FOR THE UPPER PART
      X1 = XTR(1)
      DO 10 I=1,MSP
      ZM(1,I) = X1
      ZM(2,I) = ZM(3,I) = 0.
   10 CONTINUE
      DO 30 J=2,NTR
      JN = (J-1)*MSP
      XJ = XTR(J)
      SEMI = YLE(J)
      DO 20 I=1,MSP
      ZM(1,JN+I) = XJ
      ZM(2,JN+I) = SEMI*YSP(I)
   20 ZM(3,JN+I) = 0.
   30 CONTINUE
C                           OBTAINS CORNER POINTS COORD. FOR THE LOWER PART
C                           WITH SWEPT TRAILING EDGE
      KT1 = NTC-1
```

```
      NTE = NTR - KT1
      NMZ = KT1*MSP
      DO 50 J=1,MSP
      YW(J) = ZM(2,NMZ+J)
   50 CONTINUE
      CALL SWEPTE(XTR(NTC),YLE(NTC),NTE,YW,MSP,XY)
      NTE1 = NTE-1
      DO 60 J=1,NTE1
      JM = NMZ + J*MSP
      XJ = XTR(NTC+J)
      DO 60 I=1,MSP
      IM = JM + I
      ZM(1,IM) = XJ
      ZM(2,IM) = XY(I,J)
      ZM(3,IM) = 0.
   60 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SHEGEN(ALPHA,X,S,N,Y,Z)
C******
C     SUBROUTINE SHEGEN (ALPHA,X,S,N,Y,Z)
C
C     PURPOSE  TO PROVIDE AN INITIAL GUESS OF THE FREE AND FED SHEET
C              GEOMETRY AT A PARTICULAR TRANSVERSE CUT
C
C     INPUT    CALLING SEQUENCE
C              ALPHA - ANGLE OF ATTACK OF THE WING (IN RADIANS)
C              X - X COORDINATE OF TRANSVERSE CUT (APEX IS X=0.0)
C              S - Y COORDINATE OF LEADING EDGE ON TRANSVERSE CUT
C              N - DESIRED NUMBER OF FREE SHEET PANELS IN TRANSVERSE CUT
C
C     OUTPUT   CALLING SEQUENCE
C              Y - Y COORDINATE OF CORNER POINTS DEFINING SHAPE OF
C                  FREE AND FED SHEETS ON GIVEN TRANSVERSE CUT
C              Z - Z COORDINATES OF CORNER POINTS DEFINING SHAPE OF
C                  FREE AND FED SHEETS ON GIVEN TRANSVERSE CUT
C
C     SUBROUTINES
C     CALLED   NONE
C
C     DISCUSSION  THE ROUTINE COMPUTES AN INITIAL GUESS OF THE FREE AND
C              FED SHEET GEOMETRY AT A PARTICULAR TRANSVERSE CUT.(SEE
C              STARTING SOLUTION SECTION OF ENGINEERING DOCUMENT FOR
C              METHOD) POINTS DESCRIBING THE CURVES OF FIGURE 17 ARE
C              STORED IN THE ARRAY YZVAL. EACH CURVE REPRESENTS THE
C              FREE AND FED SHEET GEOMETRY FOR ONE OF EIGHT VALUES OF A.
C              POINTS DESCRIBING THE FREE AND FED SHEET GEOMETRY FOR AN
C              ARBITRARY VALUE OF A ARE OBTAINED BY LINEAR INTERPOLATION
C              (OR EXTRAPOLATION). LINEAR INTERPOLATION IS THEN EMPLOYED
C              ON THIS NEW SET OF POINTS TO CONSTRUCT A REPRESENTATION
C              OF THE FREE SHEET BY THE NUMBER OF POINTS SPECIFIED IN
C              THE INPUT DATA.
C******
      DIMENSION Y(N),Z(N)
      DIMENSION AVAL(8),YZVAL(2,9,8),YZ(2,9),D(8)
C        SET NUMBER OF Y-Z CURVES AND NUMBER OF POINTS
C        REPRESENTING EACH CURVE
      DATA NA,NP /8,9/
C        VALUES OF A FOR EACH CURVE
      DATA AVAL /.2,.6,1.,1.4,1.8,2.2,2.6,3.0/
C        VALUES OF Y AND Z FOR POINTS ON CURVES
      DATA YZVAL /
     C1.,0.,.993,.02,.98,.045,.963,.07,.94,.088,.92,.097,.897,.097,
     C.878,.09,.892,.05,
     C1.,0.,1.012,.033,.998,.098,.973,.154,.92,.224,.853,.268,.79,
     C.277,.74,.269,.789,.146,
     C1.,0.,1.03,.047,1.028,.134,1.,.232,.958,.314,.87,.403,.764,.445,
     C.687,.44,.708,.246,
     C1.,0.,1.046,.06,1.06,.156,1.039,.29,.99,.408,.89,.53,.785,.58,
     C.68,.587,.666,.33,
```

```
     C1.,0.,1.061,.07,1.087,.188,1.076,.339,1.019,.502,.924,.624,.812,
     C.689,.704,.703,.65,.398,
     C1.,0.,1.075,.08,1.112,.222,1.11,.392,1.055,.565,.96,.7,.844,.779,
     C.735,.8,.644,.454,
     C1.,0.,1.09,.09,1.14,.252,1.142,.435,1.095,.617,1.008,.756,.877,
     C.858,.77,.882,.645,.5,
     C1.,0.,1.102,.099,1.162,.277,1.172,.468,1.132,.647,1.05,.8,.908,
     C.922,.805,.95,.648,.541
     C/
C         COMPUTE VALUE OF A FROM INPUT DATA
       A=ALPHA*X/S
C         SELECT TWO DATA CURVES FOR USE IN INTERPOLATING
C         (OR EXTRAPOLATING) NEW CURVE FOR COMPUTED VALUE OF A
      DO 10 K=2,NA
      L=K-1
      IF(A.LT.AVAL(K)) GO TO 12
 10   CONTINUE
 12   DELTA=(A-AVAL(L))/(AVAL(L+1)-AVAL(L))
C         CALCULATE POINTS DESCRIBING CURVE FOR COMPUTED
C         VALUE OF A
      DO 20 I=1,2
      DO 20 J=1,NP
 20   YZ(I,J)=YZVAL(I,J,L)+DELTA*(YZVAL(I,J,L+1)-YZVAL(I,J,L))
C         SET INITIAL POINT ON FREE SHEET
      Y(1)=YZ(1,1)
      Z(1)=YZ(2,1)
C         GET LAST POINT ON FREE SHEET
C         (AND INITIAL POINT ON FED SHEET)
      Y(N+1)=YZ(1,NP-1)
      Z(N+1)=YZ(2,NP-1)
C         SET LAST POINT ON FED SHEET
      Y(N+2)=YZ(1,NP)
      Z(N+2)=YZ(2,NP)
      D(1)=0.
      NPM1=NP-1
C         CALCULATE DISTANCES BETWEEN POINTS ON NEW CURVE
      DO 30 I=2,NPM1
      D(I)=D(I-1)+SQRT((YZ(1,I)-YZ(1,I-1))**2+(YZ(2,I)-YZ(2,I-1))**2)
 30   CONTINUE
      DIST=D(NPM1)/FLOAT(N)
C         LINEARLY INTERPOLATE DESIRED NUMBER OF POINTS
C         FOR FREE SHEET REPRESENTATION
      DO 40 I=2,N
      DIS=FLOAT(I-1)*DIST
      DO 35 J=2,NPM1
      K=J-1
      IF(DIS.LT.D(J)) GO TO 38
 35   CONTINUE
 38   DELTA=(DIS-D(K))/(D(K+1)-D(K))
      Y(I)=YZ(1,K)+DELTA*(YZ(1,K+1)-YZ(1,K))
      Z(I)=YZ(2,K)+DELTA*(YZ(2,K+1)-YZ(2,K))
 40   CONTINUE
```

```
      N2 = N + 2
C        SCALE POINTS TO ACCOUNT FOR MAGNITUDE OF LOCAL SEMI-SPAN
      DO 50 I=1,N2
      Y(I) = Y(I)*S
      Z(I) = Z(I)*S
50    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SWEPTE(X,S,N,Y,M,YP)
C******
C     SUBROUTINE SWEPTE
C
C     PURPOSE  TO CALCULATE THE Y COORDINATES OF THE PANEL CORNER POINTS
C              AFT OF THE ROOT CHORD FOR SWEPT TRAILING EDGE DESIGNS
C
C     INPUT    CALLING SEQUENCE
C              X - ARRAY OF TRANSVERSE CUT X VALUES STARTING WITH THE
C                  LAST CUT THAT INTERSECTS THE ROOT CHORD
C              S - ARRAY OF Y COORDINATES OF THE LEADING EDGE ON THE
C                  TRANSVERSE CUTS SPECIFIED BY X
C              N - NUMBER OF TRANSVERSE CUTS AFT OF THE LAST TRANSVERSE
C                  CUT TO INTERSECT THE ROOT CHORD PLUS ONE
C              Y - ARRAY OF Y COORDINATES OF PANEL CORNER POINTS LYING
C                  ON THE LAST TRANSVERSE CUT THAT INTERSECTS THE ROOT
C                  CHORD.
C              M - NUMBER OF SPANWISE PERCENT VALUES INPUT BY THE USER
C
C     OUTPUT   CALLING SEQUENCE
C              YP - ARRAY OF Y COORDINATES OF PANEL CORNER POINTS AFT OF
C                   THE ROOT CHORD.
C
C     SUBROUTINES
C     CALLED   NONE
C
C     DISCUSSION  GIVEN THE COORDINATES OF TWO POINTS DEFINING A LINE
C              AND ONE COORDINATE OF A THIRD POINT ON THE LINE, THE
C              UNKNOWN COORDINATE OF THE THIRD POINT CAN BE CALCULATED
C              BY TRIANGULATION.
C                 ONE OF THE POINTS DEFINING THE LINE IS LEADING EDGE-
C              TRAILING EDGE INTERSECTION POINT. THE OTHER POINT IS THE
C              PANEL CORNER POINTS LYING ON THE LAST TRANSVERSE CUT THAT
C              INTERSECTS THE ROOT CHORD.
C                 THE X VALUE OF THE THIRD POINT IS THE VALUE OF THE
C              TRANSVERSE CUT.
C******
      DIMENSION X(1),S(1),Y(1),YP(10,1)
      DX = X(N) - X(1)
      N1 = N - 1
      IF(N.EQ.2)  GO TO 15
      DO 10 J=2,N1
      XX = X(J) - X(1)
      J1 = J - 1
      DO 10 I=1,M
      YS = Y(I)
      TM = (S(N) - YS)/DX
      YP(I,J1) = TM*XX + YS
   10 CONTINUE
   15 CONTINUE
      DO 20 I=1,M
   20 YP(I,N1) = S(N)
      RETURN
      END
```

```
      OVERLAY(VORTEX,2,0)
      PROGRAM AICGEN
C******
C     PROGRAM  AICGEN
C
C     PURPOSE  TO CALCULATE ESSENTIAL GEOMETRY INFORMATION FOR EACH
C              PANEL AND THE LOCATIONS OF DOUBLETS AND CONTROL POINTS
C              FOR EACH NETWORK AND TO GENERATE THE AERODYNAMICS INFLUEN
C              -CE COEFFICIENTS USING AN ADVANCED PANEL-TYPE METHOD
C
C     INPUT    COMMON BLOCK
C              /INDEX/ - NT,NM,NN,NP,NZ,NPA,NZA,NNETT,NPANT,NZMPT
C              /NITE/ - NFUN
C              /IPRINT/ - IPGEOM,IPSING,IPCNTR,IPEIVC
C
C     OUTPUT   COMMON BLOCK
C              /CM03/ - NPIF,NAIC3,NAIC
C              /BDYCS/ - ZC,ZCC,ZCP,ZDC,IPC,ITC
C              /INDEX/ - NS,NC,NSA,NCA,NSNGT,NCTRT
C              /NINDX/ - NEQ,NJC,IJC
C              /ICONST/ - PI,PI2,PI4I
C
C     SUBROUTINES
C     CALLED   TGEOMC,TSING,TCNTRL,EDGEIN,KSORT,PTRNS,IPTRNS,VINFCC,VIP
C
C     DISCUSSION  SEE PROGRAM DOCUMENT 1.3 DESCRIPTION AND FLOW CHART OF
C              OVERLAY PROGRAMS
C******
      COMMON  /CM03/NTSIN,NTSOUT,NTGD,NPIF,NAIC3,NAIC,NJAC,NSCF
      COMMON/BDYCS/ZC(3,125),ZCC(3,125),ZCP(125),ZDC(125),IPC(125),
     1            ITC(125)
      COMMON/INDEX/NT(9),NM(9),NN(9),NP(9),NS(9),NC(9),NZ(9),
     *NPA(10),NSA(10),NCA(10),NZA(10),NNETT,NPANT,NSNGT,NCTRT,NZMPT
      COMMON/PANDQ/CP(3,4),PC(3),FC(3),AP(3,3),ART(3,3),P(2,4),A,B,DIAM,
     CC(6,6),AST(6,16),IIS(16),INS,ITS,NPDQ
      COMMON /NINDX/NEQ,NJC(125),IJC(125)
      COMMON /PINC/DVDFS(3,125)
      COMMON /PINDX/KP,KQ,NPWP,NPPD
      COMMON /NFAJ/NEQ,NF,NG
      COMMON /NITE/NFUN,JT,ITMX,KIT,ITPRIN
      COMMON /ZIP/IPZ,IP,ITZ,JCZ
      COMMON/ICONST/PI,PI2,PI4I
      COMMON /IPRINT/IPNPUT,IPGEOM,IPSING,IPCNTR,IPEIVC,IPOUTP
      DIMENSION AIC(125)
C                             SETS CONSTANTS PI, ETC.
      PI = 3.1415926535897931
      PI2 = 2.*PI
      PI4I = 0.25/PI
C                             CALLS ROUTINE TO GENERATE ESSENTAIL GEOME-
C                             TRY INFORMATION FOR EACH PANEL OF ALL
C                             THE NETWORKS
      KP = 0
```

```
      REWIND NPIF   $   NPWR = NPIF
      CALL TGEOMC
C                                    CALLS ROUTINE TO DESIGNATE THE LOCATION OF
C                                    DOUBLETS ON ALL NETWORKS PANELS AND TO
C                                    COMPUTE THE MATRIX FOR COEFFICIENTS OF
C                                    DOUBLET DISTRIBUTION FOR EACH PANEL
      KQ = 0
      REWIND NPIF   $   NPRD = NPIF
      KP = 0
      REWIND NTGD   $   NPWR = NTGD
      CALL TSING
C                                    CALLS ROUTINE TO DESIGNATE THE LOCATION OF
C                                    CONTROL POINTS FOR ALL NETWORK PANELS AND
C                                    TO COMPUTE THE UNIT NORMAL VECTOR AND THE
C                                    NORMAL COMPONENT OF FREE STREAM VELOCITY
C                                    VECTOR AT EVERY CONTROL POINT
      KQ = 0
      REWIND NTGD   $   NPRD = NTGD
      CALL TCNTRL
C                                    TO REARRANGE INDICES SO THAT THOSE CON-
C                                    TROL POINTS AND DOUBLETS AT EDGES OF NET-
C                                    WORKS PRECEDE ALL THE OTHERS
      CALL EDGEIN
      CALL KSORT(ZC,3,NCTRT,NJC,DVDFS)
      CALL KSORT(ZCC,3,NCTRT,NJC,DVDFS)
      CALL KSORT(ZCR,1,NCTRT,NJC,AIC)
      CALL KSORT(ZDC,1,NCTRT,NJC,AIC)
      CALL KSORT(IPC,1,NCTRT,NJC,AIC)
      CALL KSORT(ITC,1,NCTRT,NJC,AIC)
      KQ = 0
      REWIND NTGD   $   NPRD = NTGD
      KP = 0
      REWIND NPIF   $   NPWR = NPIF
      DO 100 IP=1,NPANT
      CALL PTRNS(IP)
      DO 50 I=1,INS
      IS = IIS(I)
   50 IIS(I) = NJC(IS)
      CALL IPTRNS(IP)
  100 CONTINUE
      IF(NFUN.EQ.0)  GO TO 500
C                                    TO SKIP AIC CALCULATION IF NFUN.NE.0
      DO 300 JC=1,NCTRT
      IF(ITC(JC).EQ.1)  ZCR(JC) = 0.
  300 CONTINUE
      GO TO 900
C                              GENERATES AIC MATRIX
  500 CONTINUE
C     PRINT 3001, JT
C3001 FORMAT(//* ITERATION NO.*,I4,5X,*NEW AIC MATRIX GENERATED*)
      REWIND NAIC3
      REWIND NAIC
```

```
      IF(IPEIVC.NE.0)   PRINT 1003
 1003 FORMAT(*1FROM EIVC*/)
      JPC=0
      DO 700 JC=1,NCTRT
      IPZ = IPC(JC)
      ITZ = ITC(JC)
      JCZ = JC
      IF(ITZ.EQ.1)  ZCR(JC) = 0.
      CALL VINFCC(ZC(1,JC),ZCC(1,JC),ZDC(JC),JPC)
      WRITE(NAIC3)  DVDFS
      DO 650 IS=1,NSNGT
      CALL VIP(ZCC(1,JC),1,DVDFS(1,IS),1,3,AIC(IS))
  650 CONTINUE
      WRITE(NAIC)   AIC
  700 CONTINUE
C                               CALCULATES NUMBER OF EQUATIONS FOR E, F AND G
      NEQ = NEP
      NF = NSNGT - NEQ
      NG = NP(2)
  900 RETURN
      END
```

```
      SUBROUTINE CCAL(P,C)
C******
C     SUBROUTINE CCAL (P,C)
C
C     PURPOSE  TO CALCULATE FOR EACH PANEL THE QUADRILATERAL MOMENT
C              INTEGRALS USED IN THE COMPUTATION OF THE SOURCE AND
C              DOUBLET FAR FIELD VELOCITY INFLUENCE COEFFICIENTS. (SEE
C              SECTION  B.4  , APPENDIX B OF THE ENNGINEERING DOCUMENT.)
C
C     INPUT    CALLING SEQUENCE
C              P - COORDINATES OF FOUR CORNER POINTS OF QUADRILATERAL
C
C     OUTPUT   CALLING SEQUENCE
C              C - ARRAY OF MOMENT INTEGRALS
C
C     SUBROUTINES
C     CALLED   ECAL,ZERO
C
C     DISCUSSION  THE ROUTINE COMPUTES THE QUADRILATERAL MOMENT
C              INTEGRALS C(M,N)=I(SIGMA,KSE**(M-1)*ETA**(N-1),DKSE*DETA)
C              FOR M=1,MXQ AND N=1,MXQ-M+1. A DESCRIPTION OF THE
C              CALCULATIONS PERFORMED IS CONTAINED IN SECTION B.4 OF
C              APPENDIX B OF THE ENGINEERING DOCUMENT. THE RELEVANT
C              EQUATIONS ARE (B.93) THROUGH (B.102).  THE RELEVANT
C              PROCEDURE IS PROCEDURE 6. THE CODE CLOSELY FOLLOWS THE
C              DEVELOPMENT AND NOTATION OF THIS PORTION OF APPENDIX B.
C******
      DIMENSION P(2,4),C(6,6)
      COMMON/SKAIC2/R1(2),R2(2),DR(2),E(7),GA(6,6),DUMS(241)
C        EQUATIONS AND PROCEDURES REFERENCED IN THIS ROUTINE
C        ARE CONTAINED IN APPENDIX B OF ENGINEERING DOCUMENT
C
C        SET ORDER OF MOMENTS DESIRED
      MXQ=6
      MXQP1=MXQ+1
      CALL ZERO(C,MXQ*MXQ)
C        CYCLE THROUGH SIDES OF QUADRILATERAL
      DO 500 IS=1,4
C        EXECUTE PROCEDURE 6
      ISP1=MOD(IS,4)+1
C        CALCULATE GEOMETRIC QUANTITIES ASSOCIATED WITH
C        SIDE OF QUADRILATERAL
      DO 50 I=1,2
      R1(I)=P(I,IS)
      R2(I)=P(I,ISP1)
 50   DR(I)=R2(I)-R1(I)
      DRMS=DR(1)*DR(1)+DR(2)*DR(2)
      IF(DRMS.EQ.0.) GO TO 500
      A=R1(1)*R2(2)-R1(2)*R2(1)
C        BRANCH TO PROCEDURE (6.A) OR (6.B)
      IF(ABS(DR(1))-ABS(DR(2))) 100,100,200
 100  A1=DR(1)/DR(2)
```

```
      A2=A/DP(2)
C         PROCEDURE (6.A.I)
      CALL ECAL(R1(2),R2(2),1.,1.,E,MXQP1)
      DO 130 N=1,MXQ
 130  GA(1,N)=A2*E(N+1)/FLOAT(N)
      IF(MXQ.LT.2) GO TO 300
C         PROCEDURE (6.A.II)
      DO 170 M=2,MXQ
      MXN=MXQ-M+1
      DO 170 N=1,MXN
 170  GA(M,N)=A1*GA(M-1,N+1)+A2*GA(M-1,N)
      GO TO 300
 200  A1=DR(2)/DP(1)
      A2=A/DP(1)
C         PROCEDURE (6.B.I)
      CALL ECAL(R1(1),R2(1),1.,1.,E,MXQP1)
      DO 230 M=1,MXQ
 230  GA(M,1)=A2*E(M+1)/FLOAT(M)
      IF(MXQ.LT.2) GO TO 300
C         PROCEDURE (6.B.II)
      DO 270 N=2,MXQ
      MXM=MXQ-N+1
      DO 270 M=1,MXM
 270  GA(M,N)=A1*GA(M+1,N-1)-A2*GA(M,N-1)
 300  CONTINUE
C         PERFORM ACCUMULATION OF EQUATION (B.94)
      DO 400 M=1,MXQ
      MXN=MXQ-M+1
      DO 400 N=1,MXN
 400  C(M,N)=C(M,N)+GA(M,N)/FLOAT(M+N)
 500  CONTINUE
 900  RETURN
      END
```

```
      SUBROUTINE CONTRL(NT,NM,NN,NC,NPA,ZM,ZC,ZCC,ZCR,ZDC,IPC,ITC)
C******
C     SUBROUTINE CONTRL(NT,NM,NN,NC,NPA,ZM,ZC,ZCC,ZCR,ZDC,IPC,ITC)
C
C     PURPOSE  TO COMPUTE CONTROL POINT DEFINING QUANTITIES FOR EACH
C              NETWORK
C
C     INPUT    CALLING SEQUENCE
C              NT - NETWORK TYPE
C              NM - NUMBER OF SPANWISE CUTS IN THE NETWORK
C              NN - NUMBER OF TRANSVERSE CUTS IN THE NETWORK
C              NPA - TOTAL NUMBER OF PANELS IN ALL PREVIOUS NETWORKS
C              ZM - COORDINATES OF CORNER POINTS OF THE NETWORK
C
C              COMMON BLOCK
C              /IPRINT/ - IPCNTR
C              /FSVEL/ - FSV
C              /PANDO/ - PC
C
C     OUTPUT   CALLING SEQUENCE
C              NC - NUMBER OF CONTROL POINTS ON THE NETWORK
C              ZC - COORDINATES OF CONTROL POINTS ON NETWORK
C              ZCC - SURFACE NORMAL VECTOR AT CONTROL POINTS
C              ZCR - NORMAL COMPONENTS OF FREESTREAM VELOCITY
C              ZDC - RELOCATION DISTANCE OF CONTROL POINT
C              IPC - SEQUENCE NUMBER OF PANEL TO WHICH CONTROL POINT
C                    BELONGS
C              ITC - NETWORK EDGE CONTROL POINT INDICATOR
C
C     SUBROUTINES
C     CALLED   GCPCAL,GRDIND,PTPNS,SURPRO,MMULT
C
C     DISCUSSION  THE ROUTINE CALCULATES QUANTITIES ASSOCIATED WITH THE
C                 CONTROL POINTS AND BOUNDARY CONDITIONS OF THE PROBLEM.
C                 SEPARATE COMPUTATIONS ARE PERFORMED FOR EACH NETWORK
C                 TYPE.
C                    FIRST THE CONTROL POINTS (POINTS AT WHICH THE
C                 BOUNDARY CONDITIONS ARE APPLIED) ARE LOCATED. THIS IS
C                 DONE BY AVERAGING CERTAIN COMBINATIONS OF CORNER POINTS
C                 AND THEN PROJECTING THE RESULTANT POINTS ONTO THE PANEL
C                 SURFACES. THOSE CONTROL POINTS LOCATED ON A NETWORK EDG
C                 ARE WITHDRAWN SLIGHTLY FROM THE EDGE AND NOT PROJECTED
C                 ONTO THEIR PANEL SURFACES TO AVOID NUMERICAL DIFFICULTY
C                 LATER. THE CONTROL POINTS ARE ORDERED AND INDEXED ALONG
C                 WITH AUXILIARY QUANTITIES WHICH ARE COMPUTED AS WELL.
C                    SUCH QUANTITIES INCLUDE THE PANEL NORMAL AT THE CONT
C                 POINT, THE COMPONENT OF FREESTREAM VELOCITY IN THIS
C                 DIRECTION (FOR USE IN APPLYING THE BOUNDARY CONDITIONS)
C                 AND THE DISTANCE THE EDGE CONTROL POINTS ARE WITHDRAWN.
C******
      COMMON/FSVEL/FSV(3),FSVM
      COMMON/PANDO/CP(3,4),PC(3),PO(3),AF(3,3),APT(3,3),P(2,4),A,B,DIAM,
```

```
     CC(6,6),AST(6,16),IIS(16),INS,ITS,NPDQ
      COMMON /SKFCH1/ZA(3,175),IA(175)
      COMMON /IPRINT/IPNPUT,IPGEOM,IPSING,IPCNTR,IPEIVC,IPOUTP
      DIMENSION ZC(3,100),ZCC(3,100),ZCR(100),ZDC(100),IPC(100),ITC(100)
      DIMENSION ZM(3,NM,NN)
      DATA  DELTA /1.0E-5/
      IF(NT.EQ.5.OR.NT.EQ.7)  DELTA = 1.0E-7
      IF(IPCNTR.NE.0)  PRINT 1001
 1001 FORMAT(1H1)
      NN1=NN+1
      NM1=NM+1
C        CALCULATE LOCATION OF CONTROL POINTS FROM CORNER POINT DATA
      CALL GCPCAL(NM,NN,NM1,NN1,ZM,ZA)
C        ORDER NON-IDENTICAL CONTROL POINTS
      CALL GRDIND(NM1,NN1,ZA,IA,NIA)
C        TRANSFER TO CODE FOR APPROPRIATE NETWORK TYPE
      GO TO(100,200,300,400,500,600,600)  NT
 100  CONTINUE
C        SOURCE/ANALYSIS NETWORK CALCULATIONS
C        (NOT AN OPTION IN PRESENT PROGRAM)
      JC=0
      DO 199 N=2,NN
      DO 198 M=2,NM
      JC=JC+1
      IP=M-1+(NM-1)*(N-2)+NPA
      IPC(JC)=IP
      CALL PTRNS(IP)
      CALL SURPRO(PC,ZC(1,JC),ZCC(1,JC))
      CALL MMULT(ZCC(1,JC),FSV,ZCR(JC),1,3,1)
      ZCR(JC)=-ZCR(JC)
 198  CONTINUE
 199  CONTINUE
      NC=JC
      GO TO 800
 200  CONTINUE
C        DOUBLET/ANALYSIS (WING) NETWORK CALCULATIONS
      JC=0
C        CYCLE THROUGH ALL CONTROL POINTS ON THE NETWORK
      DO 299 N=1,NN1
      DO 298 M=1,NM1
      LMN=M+NM1*(N-1)
C        COMPUTE INDICES ASSOCIATED WITH CONTROL POINT
      IF(IA(LMN).LE.JC) GO TO 298
      JC=JC+1
      IP=MIN0(MAX0(M,2),NM)-1+(NM-1)*(MIN0(MAX0(N,2),NN)-2)+NPA
      IPC(JC)=IP
      ITC(JC) = 0
      IF(M.EQ.1.OR.M.EQ.NM1.OR.N.EQ.1.OR.N.EQ.NN1)  ITC(JC) = 1
C        RETRIEVE PANEL INFORMATION
      CALL PTRNS(IP)
      SDC = 0.
C        CALCULATE CONTROL POINT COORDINATES
```

```
      DO 220 L=1,3
      DPZ = DELTA*(PC(L)-ZA(L,LMN))
      ZC(L,JC) = ZA(L,LMN) + DPZ
      SDC = SDC + DPZ**2
  220 CONTINUE
      ZDC(JC) = SQRT(SDC)
      IF(ITC(JC).NE.1)  ZDC(JC) = 0.
C        PROJECT CONTROL POINT ONTO PANEL SURFACE UNLESS
C        CONTROL POINT IS ON NETWORK EDGE
C        COMPUTE SURFACE NORMAL AT CONTROL POINT
      IF(ZDC(JC).EQ.0.)  CALL SURPRO(ZC(1,JC),ZC(1,JC),ZCC(1,JC))
      IF(ZDC(JC).NE.0.)  CALL SURPRO(ZC(1,JC),ZCC(1,JC),ZCC(1,JC))
C        COMPUTE NORMAL COMPONENT OF FREESTREAM
C        VELOCITY AT CONTROL POINT
      CALL MMULT(ZCC(1,JC),FSV,ZCP(JC),1,3,1)
      ZCP(JC)=-ZCP(JC)
      IF(IPCNTR.NE.0)
     $WRITE(6,1000) JC,IP,ZC(1,JC),ZC(2,JC),ZC(3,JC),ZCP(JC),ZDC(JC)
  298 CONTINUE
  299 CONTINUE
      NC=JC
      GO TO 800
  300 CONTINUE
C        SOURCE/DESIGN NETWORK CALCULATIONS
C        (NOT AN OPTION IN PRESENT PROGRAM)
      GO TO 800
  400 CONTINUE
C        DOUBLET/DESIGN (FREE SHEET) NETWORK CALCULATIONS
      JC=0
      DO 429 N=1,NN
      DO 428 M=1,NM
C        RE-ORDER CONTROL POINTS ELIMINATING CONTROL POINTS
C        ON TWO OF THE NETWORK EDGES
      LMN = M + NM*(N-1)
      LMNP=M+NM1*(N-1)
      DO 420 L=1,3
      ZA(L,LMN)=ZA(L,LMNP)
  420 CONTINUE
  428 CONTINUE
  429 CONTINUE
C        ORDER NON-IDENTICAL CONTROL POINTS
      CALL GRDIND(NM,NN,ZA,IA,NIA)
C        CYCLE THROUGH ALL CONTROL POINTS ON THE NETWORK
      DO 499 N=1,NN
      DO 498 M=1,NM
C        COMPUTE INDICES ASSOCIATED WITH CONTROL POINTS
      LMN=M+NM*(N-1)
      IF(IA(LMN).LE.JC) GO TO 498
      JC=JC+1
      IP=MIN0(MAX0(M,2),NM)-1+(NM-1)*(MIN0(MAX0(N,2),NN)-2)+NPA
      IPC(JC)=IP
      ITC(JC) = 0
```

```
      IF(M.EQ.1.OR.N.EQ.1)  ITC(JC) = 1
C        RETRIEVE PANEL INFORMATION
      CALL PTRNS(IP)
      SDC = 0.
C        CALCULATE CONTROL POINT COORDINATES
      DO 450 L=1,3
      DPZ = DELTA*(PC(L)-ZA(L,LMN))
      ZC(L,JC) = ZA(L,LMN) + DPZ
      SDC = SDC + DPZ**2
 450  CONTINUE
      ZDC(JC) = SQRT(SDC)
      IF(ITC(JC).NE.1)  ZDC(JC) = 0.
C        PROJECT CONTROL POINT ONTO PANEL SURFACE
C        UNLESS CONTROL POINT IS ON NETWORK EDGE
C        COMPUTE SURFACE NORMAL AT CONTROL POINT
      IF(ZDC(JC).EQ.0.)  CALL SURPRO(ZC(1,JC),ZC(1,JC),ZCC(1,JC))
      IF(ZDC(JC).NE.0.)  CALL SURPRO(ZC(1,JC),ZCC(1,JC),ZCC(1,JC))
C        COMPUTE NORMAL COMPONENT OF FREESTREAM
C        VELOCITY AT CONTROL POINT
      CALL MMULT(ZCC(1,JC),FSV,ZCR(JC),1,3,1)
      ZCR(JC)=-ZCR(JC)
      IF(IPCNTR.NE.0)
     $WRITE(6,1000) JC,IP,ZC(1,JC),ZC(2,JC),ZC(3,JC),ZCR(JC),ZDC(JC)
 498  CONTINUE
 499  CONTINUE
      NC = JC
      GO TO 800
 500  CONTINUE
C        DOUBLET/DESIGN (WAKE) NETWORK CALCULATIONS
      JC=0
C        CYCLE THROUGH ALL CONTROL POINTS ON THE NETWORK
      DO 599 N=1,1
      DO 598 M=1,NM1
C        COMPUTE INDICES ASSOCIATED WITH NETWORK
      LMN=M+NM1*(N-1)
      IF(IA(LMN).LE.JC) GO TO 598
      JC=JC+1
      IP=MINO(MAXO(M,2),NM)-1+(NM-1)*(MINO(MAXO(N,2),NN)-2)+NPA
      IPC(JC)=IP
      ITC(JC) = 1
C        RETRIEVE PANEL INFORMATION
      CALL PTRNS(IP)
      SDC = 0.
C        CALCULATE CONTROL POINT COORDINATES
      DO 520 L=1,3
      DPZ = DELTA*(PC(L)-ZA(L,LMN))
      ZC(L,JC) = ZA(L,LMN) + DPZ
      SDC = SDC + DPZ**2
 520  CONTINUE
      ZDC(JC) = SQRT(SDC)
C        PROJECT CONTROL POINT ONTO PANEL SURFACE
C        COMPUTE SURFACE NORMAL AT CONTROL POINT
```

```
      IF(ZDC(JC).EQ.0.)  CALL SURPRO(ZC(1,JC),ZC(1,JC),ZCC(1,JC))
      IF(ZDC(JC).NE.0.)  CALL SURPRO(ZC(1,JC),ZCC(1,JC),ZCC(1,JC))
C        COMPUTE NORMAL COMPONENT OF FREESTREAM
C        VELOCITY AT CONTROL POINT
      CALL MMULT(ZCC(1,JC),FSV,ZCR(JC),1,3,1)
      ZCR(JC)=-ZCR(JC)
      IF(IPCNTR.NE.0)
     $WRITE(6,1000) JC,IP,ZC(1,JC),ZC(2,JC),ZC(3,JC),ZCR(JC),ZDC(JC)
 598  CONTINUE
 599  CONTINUE
      NC=JC
      GO TO 800
 600  CONTINUE
C        DOUBLET/DESIGN (FED SHEET) CALCULATIONS
      JC=0
      ND = NN1
      IF(NT.EQ.7)  ND = 1
C        CYCLE THROUGH ALL CONTROL POINTS ON THE NETWORK
      DO 699 N=1,ND
      DO 698 M=1,1
C        COMPUTE INDICES ASSOCIATED WITH THE NETWORK
      LMN=M+NM1*(N-1)
      JC=JC+1
      IP=MIN0(MAX0(M,2),NM)-1+(NM-1)*(MIN0(MAX0(N,2),NN)-2)+NPA
      IPC(JC)=IP
      ITC(JC) = 1
C        RETRIEVE PANEL INFORMATION
      CALL PTRNS(IP)
      SDC = 0.
C        CALCULATE CONTROL POINT COORDINATES
      DO 620 L=1,3
      DPZ = DELTA*(PC(L)-ZA(L,LMN))
      ZC(L,JC) = ZA(L,LMN) + DPZ
      SDC = SDC + DPZ**2
 620  CONTINUE
      ZDC(JC) = SQRT(SDC)
C        PROJECT CONTROL POINT ONTO PANEL SURFACE
C        COMPUTE SURFACE NORMAL AT CONTROL POINT
      IF(ZDC(JC).EQ.0.)  CALL SURPRO(ZC(1,JC),ZC(1,JC),ZCC(1,JC))
      IF(ZDC(JC).NE.0.)  CALL SURPRO(ZC(1,JC),ZCC(1,JC),ZCC(1,JC))
C        COMPUTE NORMAL COMPONENT OF FREESTREAM
C        VELOCITY AT CONTROL POINT
      CALL MMULT(ZCC(1,JC),FSV,ZCR(JC),1,3,1)
      ZCR(JC)=-ZCR(JC)
      IF(IPCNTR.NE.0)
     $WRITE(6,1000) JC,IP,ZC(1,JC),ZC(2,JC),ZC(3,JC),ZCR(JC),ZDC(JC)
1000  FORMAT(2I5,5E15.6)
 698  CONTINUE
 699  CONTINUE
      NC=JC
 800  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DPIV
C******
C     SUBROUTINE DPIV
C
C     PURPOSE  TO CALCULATE THE VELOCITY INFLUENCE COEFFICIENTS INDUCED
C              AT A FIELD POINT BY A DOUBLET PANEL
C
C     INPUT    COMMON BLOCK
C              /ICONST/ - PI2,PI4I
C              /PIVINT/ - X,P,A,B,DIAM,C,NTST
C
C     OUTPUT   COMMON BLOCK
C              /PIVINT/ - DV
C
C     SUBROUTINES
C     CALLED   INTCAL,ZERO
C
C     DISCUSSION  THE ROUTINE COMPUTES THE DOUBLET PANEL VELOCITY
C              INFLUENCE COEFFICIENTS AT A SPECIFIED FIELD POINT. A
C              DESCRIPTION OF THE METHOD AND CALCULATIONS PERFORMED IS
C              CONTAINNED IN APPENDIX B OF THE ENGINEERING DOCUMENT. IF
C              THE FIELD POINT IS SUFFICIENTLY DISTANT FROM THE PANEL
C              A FAR FIELD APPROXIMATION IS EMPLOYED. THE APPROXIMATION
C              AND COMPUTATIONAL METHOD IS PRESENTED IN SECTION B.4 OF
C              APPENDIX B AND THE RELATED CODE COMPRISES THE PART OF
C              DPIV BETWEEN STATEMENT 120 AND STATEMENT 500. THE
C              LOOP 450 CONTAINS THE BULK OF THE CALCULATIONS AND ITS
C              PURPOSE IS TO COMPUTE THE J VECTORS OF EQUATION (B.91).
C              FOR THIS CALCULATION THE TERMS ON THE RIGHT SIDE OF
C              EQUATION (B.91) HAVE BEEN EXPANDED, HENCE THE CODE DOES
C              NOT DIRECTLY CORRELATE WITH THIS FORMULA. ANOTHER
C              EVALUATION PROCEDURE IS EMPLOYED WHEN THE FIELD POINT
C              IS NEAR THE PANEL. A DESCRIPTION OF THIS PROCEDURE IS
C              PRESENTED IN SECTIONS B.2 AND B.3 OF APPENDIX B. THE
C              RELATED CODE COMPRISES THE PART OF DPIV BETWEEN
C              STATEMENTS 500 AND 900. THE LOOP 750 CALCULATES THE
C              VECTOR J DEFINED BY EQUATION (B.34) WITH THE H INTEGRALS
C              COMPUTED BY THE ROUTINE INTCAL. THE LOOP 800 TRANSFORMS
C              THE INFLUENCE COEFFICIENTS RELATIVE TO THE EXPANSION OF
C              DOUBLET STRENGTH ABOUT THE PROJECTION OF THE FIELD
C              POINT TO COEFFICIENTS RELATIVE TO THE EXPANSION OF
C              DOUBLET STRENGTH ABOUT THE ORIGIN.
C******
      LOGICAL LOGAB
      COMMON/ICONST/PI,PI2,PI4I
      COMMON/INTG/H(6,6,7),HZ,IH,MXQ,MXK
      COMMON/PIVINT/X(3),P(2,4),A,B,DIAM,C(6,6),DV(3,6),NTST,NCF
      DIMENSION Q(3),QEX(3,6),MJ(6),NJ(6)
      DATA MJ,NJ /1,2,1,3,2,1,1,1,2,1,2,3/
      DATA DELMD,DELMQ /4.,2.45/
C        EQUATIONS AND QUANTITIES REFFRENCED IN THIS ROUTINE
C        ARE CONTAINED IN APPENDIX B OF THE ENGINEERING DOCUMENT
```

```
      A2=2.*A
      B2=2.*B
C        TEST FOR POSSIBILITY OF EMPLOYING FAR FIELD APPROXIMATION
      XM=SQRT(X(1)*X(1)+X(2)*X(2)+X(3)*X(3))
      RATIO=XM/DIAM
C        BASIC TEST
      IF(RATIO.GT.DELMD) GO TO 120
C        AUXILIARY TEST BASED ON MORE REFINED ESTIMATE
      RATIO=0.
      DO 100 IC=1,4
      QIP=ABS(P(1,IC)*(P(1,IC)-2.*X(1))+P(2,IC)*(P(2,IC)-2.*X(2)))
  100 RATIO=AMAX1(RATIO,QIP)
      RATIO=XM*XM/RATIO
C        BRANCH TO NEAR FIELD CALCULATION IF FIELD POINT TOO
C        CLOSE TO PANEL
      IF(RATIO.LT.DELMQ) GO TO 500
  120 CONTINUE
C        FAR FIELD CALCULATIONS
      U=1./XM
      U2=U*U
      U3=U*U2
      X(1)=U*X(1)
      X(2)=U*X(2)
      X(3)=U*X(3)
      A2X=A2*X(1)
      B2Y=B2*X(2)
      UD4PI=U3*PI4I
      UD8PI=.5*UD4PI
C        CALCULATE VECTOR J OF EQUATION (B.91)
      DO 450 J=1,NTST
      M=MJ(J)
      N=NJ(J)
      QDPI=UD4PI
      IF((J.EQ.4).OR.(J.EQ.6)) QDPI=UD8PI
      E1=C(M,N)
      E1M=C(M+1,N)
      E1N=C(M,N+1)
      E7=-A2X*E1M-B2Y*E1N+X(3)*E1
      XF=-3.*E7
      Q(1)=XF*X(1)-A2*E1M
      Q(2)=XF*X(2)-B2*E1N
      Q(3)=XF*X(3)+E1
C        USE ONLY MONOPOLE APPROXIMATION IF FIELD POINT SUFFICIENTLY
C        DISTANT
      IF(RATIO.GT.DELMD) GO TO 400
      E1MM=C(M+2,N)
      E1MN=C(M+1,N+1)
      E1NN=C(M,N+2)
      E2=X(1)*E1M+X(2)*E1N
      E2M=X(1)*E1MM+X(2)*E1MN
      E2N=X(1)*E1MN+X(2)*E1NN
      E5=A*E1MM+B*E1NN
```

```
      E7M=-A2X*E1MM-B2Y*E1MN+X(3)*E1M
      E7N=-A2X*E1MN-B2Y*E1NN+X(3)*E1N
      XF=-3.*E5-15.*(X(1)*E7M+X(2)*E7N+X(3)*X(3)*E5)
      Q(1)=Q(1)+U*(XF*X(1)+3.*(E7M-A2*E2M))
      Q(2)=Q(2)+U*(XF*X(2)+3.*(E7N-B2*E2N))
      Q(3)=Q(3)+U*(XF*X(3)+3.*(E2+2.*X(3)*E5))
  400 CONTINUE
      DO 425 I=1,3
      DV(I,J)=QDPI*Q(I)
  425 CONTINUE
  450 CONTINUE
      GO TO 900
  500 CONTINUE
C        NEAR FIELD CALCULATIONS
C        DETERMINE ORDER OF H INTEGRALS REQUIRED
      MXQ=6
      MXK=7
C        CHECK IF PANEL IS FLAT
      LOGAB=(A.EQ.0.).AND.(B.EQ.C.)
      IF(LOGAB) MXQ=4
      IF(LOGAB) MXK=5
      IF(NTST.LT.6) MXQ=MXQ-1
      IF(NTST.LT.3) MXQ=MXQ-1
C        CALCULATE H INTEGRALS
      CALL INTCAL
C        CALCULATE QUANTITIES FREQUENTLY USED IN SUBSEQUENT
C        COMPUTATIONS
      CAB=A*X(1)*X(1)+B*X(2)*X(2)-X(3)+HZ
      PIF=PI4I
      X2=2.*X(1)
      Y2=2.*X(2)
      A2X=A2*X(1)
      B2Y=B2*X(2)
      H3=3.*HZ
      H6=6.*HZ
      H9=9.*HZ
      HH3=3.*HZ*HZ
      HH15=15.*HZ*HZ
      HHH15=HZ*HH15
      CALL ZERO(QEX,3*NTST)
      IF(IH.EQ.0) GO TO 675
C        CALCULATE AUXILIARY TERMS OF H INTEGRALS DEFINED BY
C        EQUATION (B.51) IF PROCEDURE 3 HAS BEEN EMPLOYED
      SPI2=PI2
      IF(HZ.LT.(0.)) SPI2=-PI2
      IF(NTST.LT.3) GO TO 675
      QEX(1,2)=SPI2*(1.+A2*HZ)
      QEX(3,2)=SPI2*(A2X)
      QEX(2,3)=SPI2*(1.+B2*HZ)
      QEX(3,3)=SPI2*(B2Y)
      IF(NTST.LT.6) GO TO 675
      QEX(1,4)=SPI2*(4.*A2X*HZ)
```

```
      QEX(2,4)=SPI2*(2.*B2Y*HZ)
      QEX(3,4)=SPI2*(2.*(-HZ*(1.+HZ*(3.*A+B))+CAB))
      QEX(1,5)=SPI2*(B2Y*HZ)
      QEX(2,5)=SPI2*(A2X*HZ)
      QEX(1,6)=SPI2*(2.*A2X*HZ)
      QEX(2,6)=SPI2*(4.*B2Y*HZ)
      QEX(3,6)=SPI2*(2.*(-HZ*(1.+HZ*(3.*B+A))+CAB))
  675 CONTINUE
C        CALCULATE VECTOR J DEFINED BY EQUATION (B.34)
      DO 750 J=1,NTST
      M=MJ(J)
      N=NJ(J)
      Q(1)=QEX(1,J)+H3*H(M+1,N,5)
      Q(2)=QEX(2,J)+H3*H(M,N+1,5)
      Q(3)=QEX(3,J)+H(M,N,3)-HH3*H(M,N,5)
C        IGNORE TERMS DEPENDING ON CURVATURE IF PANEL IS FLAT
      IF(LOGAB) GO TO 699
      Q(1)=Q(1)
     C+A*((3.*H(M+3,N,5)-2.*H(M+1,N,3)+HH15*H(M+3,N,7))
     C+X2*(-H(M,N,3)+HH15*H(M+2,N,7)))
     C+B*((3.*H(M+1,N+2,5)+HH15*H(M+1,N+2,7))+Y2*(HH15*H(M+1,N+1,7)))
     C+CAB*(-3.*H(M+1,N,5)+HH15*H(M+1,N,7))
      Q(2)=Q(2)
     C+A*((3.*H(M+2,N+1,5)+HH15*H(M+2,N+1,7))+X2*(HH15*H(M+1,N+1,7)))
     C+B*((3.*H(M,N+3,5)-2.*H(M,N+1,3)+HH15*H(M,N+3,7))
     C+Y2*(-H(M,N,3)+HH15*H(M,N+2,7)))
     C+CAB*(-3.*H(M,N+1,5)+HH15*H(M,N+1,7))
      Q(3)=Q(3)
     C+A*((H3*H(M+2,N,5)-HHH15*H(M+2,N,7))
     C+X2*(H6*H(M+1,N,5)-HHH15*H(M+1,N,7)))
     C+B*((H3*H(M,N+2,5)-HHH15*H(M,N+2,7))
     C+Y2*(H6*H(M,N+1,5)-HHH15*H(M,N+1,7)))
     C+CAB*(H9*H(M,N,5)-HHH15*H(M,N,7))
  699 CONTINUE
      DO 700 I=1,3
      DV(I,J)=FIF*Q(I)
  700 CONTINUE
  750 CONTINUE
      IF(NTST.LT.3) GO TO 900
C        REARRANGE COEFFICIENTS AS REQUIRED BY EQUATION (B.31)
      DO 800 I=1,3
      DV(I,2)=DV(I,2)+X(1)*DV(I,1)
      DV(I,3)=DV(I,3)+X(2)*DV(I,1)
      IF(NTST.LT.6) GO TO 800
      DVX=DV(I,2)-.5*X(1)*DV(I,1)
      DVY=DV(I,3)-.5*X(2)*DV(I,1)
      DV(I,4)=.5*DV(I,4)+X(1)*DVX
      DV(I,5)=DV(I,5)+X(1)*DVY+X(2)*DVX
      DV(I,6)=.5*DV(I,6)+X(2)*DVY
  800 CONTINUE
  900 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ECAL(X1,X2,A1,A2,E,N)
C******
C     SUBROUTINE ECAL (X1,X2,A1,A2,E,N)
C
C     PURPOSE   TO EVALUATE E(I)=A2*X2**(I-1)-A1*X1**(I-1)    I=1,N.
C               (SEE EQUATION (B.59), APPENDIX B OF ENGINEERING DOCUMENT)
C
C     INPUT     CALLING SEQUENCE
C               X1 - (SEE PURPOSE)
C               X2 - (SEE PURPOSE)
C               A1 - (SEE PURPOSE)
C               A2 - (SEE PURPOSE)
C               N - (SEE PURPOSE)
C
C     OUTPUT    CALLING SEQUENCE
C               E - (SEE PURPOSE)
C
C     SUBROUTINES
C     CALLED    NONE
C
C     DISCUSSION  THE ROUTINE CALCULATES THE QUANTITIES
C               E(I)=A2*X2**(I-1)-A1*X1**(I-1)  FOR I=1,N  USING THE
C               RECURSION FORMULA  E(I)=(X1+X2)*E(I-1)-X1*X2*E(I-2)
C               AND THE INITIAL CONDITIONS  E(1)=A2-A1  AND
C               E(2)=A2*X2-A1*X1.
C******
      DIMENSION E(N)
      E(1)=A2-A1
      IF(N.LT.2) GO TO 900
      E(2)=A2*X2-A1*X1
      IF(N.LT.3) GO TO 900
      X2PX1=X2+X1
      X2TX1=X2*X1
      DO 10 I=3,N
 10   E(I)=X2PX1*E(I-1)-X2TX1*E(I-2)
 900  RETURN
      END
```

```
      SUBROUTINE EDGEIN
C******
C     SUBROUTINE EDGEIN
C
C     PURPOSE  TO PROVIDE NEW INDICES FOR THE CONTROL POINTS AND DOUBLET
C              -S SO THAT THE CORRESPONDING EQUATIONS(DOWNWASH CONDITION
C              ) AND DOUBLETS AT EDGES OF NETWORK WILL PRECEDE ALL THE
C              OTHERS
C
C     INPUT    COMMON BLOCK
C              /BDYCS/ - ITC
C              /INDEX/ - NCTRT
C
C     OUTPUT   COMMON BLOCK
C              /NINDX/ - NEQ,NJC,IJC
C
C     SUBROUTINES
C     CALLED   NONE
C
C     DISCUSSION  THE ROUTINE OBTAINS THE NUMBER OF EQUATIONS CORRESPOND
C              -ING TO CONTROL POINTS AT EDGES. THEN IT ASSIGNS INDICES
C              ACCORDING TO WHETHER CONTROL POINTS ARE AT EDGE OR INTER-
C              IOR.
C******
      COMMON/BDYCS/ZC(3,125),ZCC(3,125),ZCR(125),ZDC(125),IPC(125),
     1              ITC(125)
      COMMON/INDEX/NT(9),NM(9),NN(9),NP(9),NS(9),NC(9),NZ(9),
     CNPA(10),NSA(10),NCA(10),NZA(10),NNETT,NPANT,NSNGT,NCTRT,NZMPT
      COMMON /NINDX/NEQ,NJC(125),IJC(125)
C                           TO OBTAIN THE NUMBER OF EDGE DOWNWASH EQS.
      NEQ = 0
      DO 10 JC=1,NCTRT
   10 NEQ = NEQ + ITC(JC)
C                           TO ASSIGN THE INDICES
      LE = 0
      LC = NEQ
      DO 30 JC=1,NCTRT
      IF(ITC(JC).NE.1)  GO TO 20
      LE = LE + 1
      NJC(JC) = LE
      IJC(LE) = JC
      GO TO  30
   20 LC = LC + 1
      NJC(JC) = LC
      IJC(LC) = JC
   30 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE EIVC(ZC,ZNC,ZDC,IPINF)
C******
C     SUBROUTINE EIVC (ZC,ZNC,ZDC,IPINF)
C
C     PURPOSE   TO CALCULATE THE VELOCITY INDUCED BY A DOUBLET PANEL ON A
C               NETWORK EDGE CONTROL POINT
C
C     INPUT     CALLING SEQUENCE
C               ZC - COORDINATES OF CONTROL POINT
C               ZNC - UNIT NORMAL TO SURFACE AT CONTROL POINT
C               ZDC - DISTANCE FROM CONTROL POINT TO PANEL EDGE
C
C               COMMON BLOCK
C               /IPPRINT/ - IPEIVC
C               /ZIP/ - IPZ,IP,JCZ
C               /PANDQ/ - CP,PC,PQ,AR,P,DIAM
C               /SYMM/ - NSYMM
C
C     OUTPUT    CALLING SEQUENCE
C               IPINF - INDICATES WHETHER PANEL IS CLOSE ENOUGH TO
C                       CONTROL POINT TO INDUCE A SUBSTANTIAL DOWNWASH
C
C               COMMON BLOCK
C               /PIVM/ - DVDS
C
C     SUBROUTINES
C     CALLED    ZERO,CROSS,UNIPAN
C
C     DISCUSSION  THE ROUTINE CALCULATES THE VELOCITY INDUCED BY A
C               DOUBLET PANEL (AND ITS IMAGE IF CONFIGURATION IS
C               SYMMETRICAL) ON A NETWORK EDGE CONTROL POINT. THE
C               INFLUENCE IS COMPUTED BY ACCUMULATING THE INFLUENCE OF
C               EACH PANEL EDGE. THE INFLUENCE OF A PANEL EDGE IS IGNORED
C               UNLESS A POINT ON THE EDGE IS WITHIN A SMALL SPHERE
C               AROUND THE CONTROL POINT. IN THIS CASE THE INFLUENCE
C               DUE TO BOTH THE DOUBLET STRENGTH AND ITS DERIVATIVE
C               PERPENDICULAR TO THE EDGE (EVALUATED AT THAT EDGE
C               POINT) IS COMPUTED. THE RESULTANT VELOCITY IS THEN
C               DISTRIBUTED AMONG THE COEFFICIENTS OF THE DOUBLET
C               DISTRIBUTION ON THE PANEL.
C******
      COMMON/PIVM/DVDS(3,6)
      COMMON/PANDQ/CP(3,4),PC(3),PQ(3),AR(3,3),ART(3,3),P(2,4),A,B,DIAM,
     1C(6,6),AST(6,16),IIS(16),INS,ITS,NPDQ
      COMMON/SYMM/NSYMM
      COMMON /ZIP/IPZ,IP,ITZ,JCZ
      COMMON /IPRINT/IPNPUT,IPGEOM,IPSING,IPCNTR,IPEIVC,IPOUTP
      DIMENSION P1(3),P2(3),DF(3),Q(6),W(3),Z(3)
      DIMENSION  ZC(1),ZNC(1)
      DATA EX /1.1/
      IPINF=0
      EXZDC=EX*ZDC
```

```
      CALL ZERO(DVDS,18)
      W(1)=ZC(1)
      W(3)=ZC(3)
C        COMPUTE INFLUENCE OF PANEL AND IMAGE IF NSYMM=1
      DO 200 ISYMM=1,2
      SIGN=3-2*ISYMM
      W(2)=SIGN*ZC(2)
      DIST=SQRT((W(1)-PC(1))**2+(W(2)-PC(2))**2+(W(3)-PC(3))**2)
C        IGNORE INFLUENCE IF PANEL IS NOT IN
C        PROXIMITY OF CONTROL POINT
      IF(DIST.GT.DIAM) GO TO 500
C        CALCULATE POINT ON PANEL EDGE CLOSEST TO CONTROL POINT
      DO 100 IS=1,4
      ISP1=IS+1
      IF(ISP1.GT.4) ISP1=1
      DO 50 I=1,3
      R1(I)=CP(I,IS)-W(I)
      R2(I)=CP(I,ISP1)-W(I)
      DR(I)=CP(I,ISP1)-CP(I,IS)
   50 CONTINUE
      DRM=SQRT(DR(1)**2+DR(2)**2+DR(3)**2)
      IF(DRM.LT.ZDC) GO TO 100
      R1M=SQRT(R1(1)**2+R1(2)**2+R1(3)**2)
      R2M=SQRT(R2(1)**2+R2(2)**2+R2(3)**2)
      T=-(R1(1)*DR(1)+R1(2)*DR(2)+R1(3)*DR(3))/(DRM**2)
      IF((R1M.GT.EXZDC).AND.(T.LT.0.)) GO TO 100
      IF((R2M.GT.EXZDC).AND.(T.GT.1.)) GO TO 100
      DO 55 I=1,3
      Z(I)=CP(I,IS)+T*DR(I)
   55 CONTINUE
      CALL CROSS(R1,R2,Q)
      AM=SQRT(Q(1)**2+Q(2)**2+Q(3)**2)/DRM
C        IGNORE INFLUENCE OF PANEL EDGE IF EDGE IS
C        NOT IN PROXIMITY OF CONTROL POINT
      IF(AM.GT.EXZDC) GO TO 100
      SP1=(R1(1)*DR(1)+R1(2)*DR(2)+R1(3)*DR(3))/(R1M*DRM)
      SP2=(R2(1)*DR(1)+R2(2)*DR(2)+R2(3)*DR(3))/(R2M*DRM)
C        CALCULATE MAGNITUDE OF DOWNWASH INDUCED BY DOUBLET
C        STRENGTH AT PANEL EDGE POINT CLOSEST TO CONTROL POINT
      IF(SP1*SP2.GE.0.) F1=AM*(1./R2M**2-1./R1M**2)/(SP1+SP2)
      IF(SP1*SP2.LT.0.) F1=(SP1-SP2)/AM
      T=Q(1)*ZNC(1)+Q(2)*ZNC(2)+Q(3)*ZNC(3)
      IF(T.LE.0.) F1=-F1
      F1 = ZDC*F1
C        CALCULATE MAGNITUDE OF DOWNWASH INDUCED BY DERIVATIVE
C        OF DOUBLET STRENGTH PERPENDICULAR TO EDGE AT PANEL
C        EDGE POINT CLOSEST TO CONTROL POOINT
      IF(SP1.GE.0.) F=ALOG((R2M*(1.+SP2))/(R1M*(1.+SP1)))
      IF(SP2.LE.0.) F=-ALOG((R2M*(1.-SP2))/(R1M*(1.-SP1)))
      IF((SP1.LT.0.).AND.(SP2.GT.0.))
     CF=ALOG(R2M*(1.+SP2)*R1M*(1.-SP1)/(AM**2))
      F = ZDC*F
```

```
      DP=SQRT((P(1,ISP1)-P(1,IS))**2+(P(2,ISP1)-P(2,IS))**2)
      F2=-F*(P(2,ISP1)-P(2,IS))/DP
      F3=F*(P(1,ISP1)-P(1,IS))/DP
      CALL UNIPAN(AP,RO,Z,Z)
C        CALCULATE DOWNWASH INDUCED BY EACH COEFFICIENT OF
C        QUADRATIC DOUBLET DISTRIBUTION ON PANEL
      Q(1)=F1
      Q(2)=F2+F1*Z(1)
      Q(3)=F3+F1*Z(2)
      F4=F2+.5*F1*Z(1)
      F5=F3+.5*F1*Z(2)
      Q(4)=Z(1)*F4
      Q(5)=Z(1)*F5+Z(2)*F4
      Q(6)=Z(2)*F5
      DO 75 J=1,6
      DVDS(1,J)=DVDS(1,J)+ZNC(1)*Q(J)
      DVDS(2,J)=DVDS(2,J)+SIGN*ZNC(2)*Q(J)
      DVDS(3,J)=DVDS(3,J)+ZNC(3)*Q(J)
 75   CONTINUE
      IF(IPEIVC.NE.0)
     $WRITE(6,1000)  JCZ,IPZ,IP,IS,W,F1,F2,F3
 1000 FORMAT(4I5,6E15.6)
      IPINF=1
 100  CONTINUE
      IF(NSYMM.EQ.0.) GO TO 500
 200  CONTINUE
 500  RETURN
      END
```

```
      SUBROUTINE FMNCAL
C******
C     SUBROUTINE FMNCAL
C
C     PURPOSE   TO CALCULATE CERTAIN F INTEGRALS USED TO COMPUTE THE H
C               INTEGRALS INVOLVED IN THE FORMULAS FOR THE SOURCE AND
C               DOUBLET PANEL INDUCED VELOCITY INFLUENCE COEFFICIENTS.
C               (SEE SECTION B.3 OF APPENDIX B OF THE ENGINEERING
C               DOCUMENT.)
C
C     INPUT     COMMON BLOCK
C               /INTQ/ - MXQ
C               /SKAICL/ - LMXQ2,LMXQ3
C               /SKAIC1/ - AKS1,AET1,AKS2,AET2,ANK,ANE,A,AA,S1,S2,HH
C
C     OUTPUT    COMMON BLOCK
C               /SKAIC2/ - GKMN,GEMN,GAMN
C
C     SUBROUTINES
C     CALLED    ECAL
C
C     DISCUSSION  THE ROUTINE COMPUTES THE INTEGRALS F(M,N,1) FOR
C               N=1,MXQ AND M=1,MXQ-N+1  WHERE
C               F(M,N,1)=I(L,KSE**(M-1)*ETA**(N-1)/RHO,DL). A DESCRIPTION
C               OF THE CALCULATIONS PERFORMED IS CONTAINED IN SECTION B.3
C               OF APPENDIX B OF THE ENGINEERING DOCUMENT. THE RELEVANT
C               EQUATIONS ARE (B.62), (B.63), (B.64) AND (B.65). THE
C               RELEVANT PROCEDURES ARE 4 AND 5. THE CODE CLOSELY
C               FOLLOWS THE DEVELOPMENT AND NOTATION OF SECTION B.3.
C               NOTE THAT FMN(M,N)=F(M,N,1).
C******
      LOGICAL         LMXQ2,LMXQ3,LMXQ4,LMXK3,LMXK5,LMKEX
      COMMON/INTQ/H(6,6,7),HZ,IH,MXQ,MXK
      COMMON/SKAICL/LMXQ2,LMXQ3,LMXQ4,LMXK3,LMXK5,LMKEX
      COMMON/SKAIC1/AKS1,AET1,AKS2,AET2,DPM,EL1,EL2,ELM,ANK,ANE,A,AA,
     CGG,S1,S2,S1I,S2I,HM,HH
      COMMON/SKAIC2/GAK(21),GKNK(5,5),GENK(5,5),GKMN(6,6),GEMN(6,6),
     CGAMN(6,6),H111,FK(37),FNK(6,5),FMN(6,6),F(37)
C        INITIALIZE F(M,N,1) INTEGRALS
      FMN(1,1)=FK(1)
      IF(LMXQ2) GO TO 500
C        BRANCH TO STEP (3.A) OR (3.B) OF PROCEDURE 4 RESPECTIVELY
      IF(ABS(ANE)-ABS(ANK)) 100,100,200
  100 CONTINUE
      C1=A*ANE
      C3=ANK
C        EXECUTE STEP (3.A.I) (EQUATION (B.62)) OF PROCEDURE 4
      CALL ECAL(AET1,AET2,S1,S2,E,MXQ-1)
      FMN(1,2)=C1*FMN(1,1)+C3*E(1)
      IF(LMXQ3) GO TO 150
      C2=-(AA+HH*ANK*ANK)
      DO 130 N=3,MXQ
```

```
 130  FMN(1,N)=(FLOAT(2*N-3)*C1*FMN(1,N-1)+FLOAT(N-2)*C2*FMN(1,N-2)
     C+C3*E(N-1))/FLOAT(N-1)
 150  A1=-ANE/ANK
      A2=A/ANK
C        EXECUTE STEP (3.A.II) (EQUATION (B.63)) OF PROCEDURE 4
      DO 170 M=2,MXQ
      MXN=MXQ-M+1
      DO 170 N=1,MXN
 170  FMN(M,N)=A1*FMN(M-1,N+1)+A2*FMN(M-1,N)
      GO TO 500
 200  CONTINUE
      C1=A*ANK
      C3=-ANE
C        EXECUTE STEP (3.B.I) (EQUATION (B.64)) OF PROCEDURE 4
      CALL ECAL(AKS1,AKS2,S1,S2,E,MXQ-1)
      FMN(2,1)=C1*FMN(1,1)+C3*E(1)
      IF(LMXQ3) GO TO 250
      C2=-(AA+HH*ANE*ANE)
      DO 230 M=3,MXQ
 230  FMN(M,1)=(FLOAT(2*M-3)*C1*FMN(M-1,1)+FLOAT(M-2)*C2*FMN(M-2,1)
     C+C3*E(M-1))/FLOAT(M-1)
 250  A1=-ANK/ANE
      A2=A/ANE
C        EXECUTE STEP (3.B.II) (EQUATION (B.65)) OF PROCEDURE 4
      DO 270 N=2,MXQ
      MXM=MXQ-N+1
      DO 270 M=1,MXM
 270  FMN(M,N)=A1*FMN(M+1,N-1)+A2*FMN(M,N-1)
 500  CONTINUE
C        ACCUMULATE CONTRIBUTION OF QUADRILATERAL SIDE TO F
C        INTEGRALS FOR USE IN STEP 3 (EQUATION (B.43)), STEP 4
C        (EQUATION (B.44)) AND STEP 5 (EQUATION (B.45))
C        OF PROCEDURE 1
      DO 690 N=1,MXQ
      GEMN(N)=GEMN(N)+ANE*FMN(1,N)
      MXM=MXQ-N+1
      DO 680 M=1,MXM
      GKMN(M,N)=GKMN(M,N)+ANK*FMN(M,N)
 680  GAMN(M,N)=GAMN(M,N)+A*FMN(M,N)
 690  CONTINUE
 999  RETURN
      END
```

```
      SUBROUTINE FNKCAL
C******
C     SUBROUTINE FNKCAL
C
C     PURPOSE   TO CALCULATE CERTAIN F INTEGRALS USED TO COMPUTE THE H
C               INTEGRALS INVOLVED IN THE FORMULAS FOR THE SOURCE AND
C               DOUBLET PANEL INDUCED VELOCITY INFLUENCE COEFFICIENTS.
C               (SEE SECTION B.3 OF APPENDIX B OF THE ENGINEERING
C               DOCUMENT.)
C
C     INPUT     COMMON BLOCK
C               /SKAICL/ - LMXQ3,LMXQ4,LMXK5
C               /SKAIC1/ - ANK,ANE,AA,S1I,S2I,HH
C               /SKAICI/ - MXKM2,MXQM1
C
C     OUTPUT    COMMON BLOCK
C               /SKAIC2/ - GKNK,GENK
C
C     SUBROUTINES
C     CALLED    FCAL
C
C     DISCUSSION  THE ROUTINE COMPUTES THE INTEGRALS F(1,N,K) FOR
C               N=2,MXQ AND K=3,MXK-2,2   WHERE
C               F(1,N,K)=I(L,ETA**(N-1)/RHO**K,DL). A DESCRIPTION OF THE
C               CALCULATIONS PERFORMED IS CONTAINED IN SECTION B.3 OF
C               APPENDIX B OF THE ENGINEERING DOCUMENT. THE RELEVANT
C               EQUATIONS ARE (B.66) AND (B.67). THE RELEVANT PROCEDURES
C               ARE PROCEDURES 4 AND 5. THE CODE CLOSELY FOLLOWS THE
C               DEVELOPMENT AND NOTATION OF SECTION B.3. NOTE THAT
C               FNK(N,K)=F(1,N,K).
C******
      LOGICAL       LMXQ2,LMXQ3,LMXQ4,LMXK3,LMXK5,LMKEX
      COMMON/SKAICL/LMXQ2,LMXQ3,LMXQ4,LMXK3,LMXK5,LMKEX
      COMMON/SKAIC1/AKS1,AET1,AKS2,AET2,DRM,EL1,EL2,ELM,ANK,ANE,A,AA,
     CGG,S1,S2,S1I,S2I,HM,HH
      COMMON/SKAIC2/GAK(21),GKNK(5,5),GENK(5,5),GKMN(6,6),GEMN(6),
     CGAMN(6,6),H111,FK(37),FNK(6,5),FMN(6,6),E(37)
      COMMON/SKAICI/MXFK,MXFKN,MXFNK,MXKM2,MXKM4,MXQM1
C        EQUATIONS AND PROCEDURED REFERENCED IN THIS ROUTINE
C        ARE CONTAINED IN APPENDIX B IN ENGINEERING DOCUMENT
C
C        INITIALIZE THE ARRAY FNK USING PREVIOUSLY COMPUTED INTEGRALS
      DO 100 K=1,MXKM2,2
  100 FNK(1,K)=FK(K)
      IF(LMXQ3) GO TO 500
      DO 150 N=2,MXQM1
  150 FNK(N,1)=FMN(1,N)
      IF(LMXK5) GO TO 500
C        EXECUTE STEPS 4 AND 5 OF PROCEDURE 4 OR 5
      CALL FCAL(S1I,S2I,1.,1.,E,MXKM2-1)
      DO 250 K=3,MXKM2,2
      C1=A*ANE
```

```
C         STEP 4 (EQUATION B.66)
      FNK(2,K)=C1*FNK(1,K)-ANK*E(K-1)/FLOAT(K-2)
      IF(LMXO4) GO TO 250
      C1=2.*C1
      C3=ANK*ANK
      C2=-(AA+C3*HH)
C         STEP 5 (EQUATION B.67)
      DO 200 N=3,MXQM1
  200 FNK(N,K)=C1*FNK(N-1,K)+C2*FNK(N-2,K)+C3*FNK(N-2,K-2)
  250 CONTINUE
  500 CONTINUE
C         ACCUMULATE F INTEGRALS OVER ALL FOUR SIDES OF QUADRILATERAL
C         FOR USE IN STEPS 6 (EQUATION B.46) AND 7 (EQUATION B.47)
C         OF PROCEDURES 1, 2 AND 3
      DO 690 K=1,MXKM2,2
      DO 680 N=1,MXQM1
      GKNK(N,K)=GKNK(N,K)+ANK*FNK(N,K)
  680 GENK(N,K)=GENK(N,K)+ANE*FNK(N,K)
  690 CONTINUE
  999 RETURN
      END
```

```
      SUBROUTINE FKCAL
C******
C     SUBROUTINE FKCAL
C
C     PURPOSE  TO CALCULATE CERTAIN F INTEGRALS USED TO COMPUTE THE H
C              INTEGRALS INVOLVED IN THE FORMULAS FOR THE SOURCE AND
C              DOUBLET PANEL INDUCED VELOCITY INFLUENCE COEFFICIENTS.
C              (SEE SECTION B.3 OF APPENDIX B OF THE ENGINEERING
C              DOCUMENT.)
C
C     INPUT    COMMON BLOCK
C              /SKAICL/ - LMKEX
C              /SKAIC1/ - EL1,EL2,ELM,A,AA,GG,S1,S2,S1I,S2I,HM
C              /SKAICI/ - MXFK
C
C     OUTPUT   COMMON BLOCK
C              /SKAICI/ - MXFKN
C              /SKAIC2/ - GAK,H111
C
C     SUBROUTINES
C     CALLED   ECAL
C
C     DISCUSSION  THE ROUTINE COMPUTES THE INTEGRALS F(1,1,K) FOR
C              K=1,MXFK  WHERE F(1,1,K)=I(L,1./RHO**K,DL). A DESCRIPTION
C              OF THE CALCULATIONS PERFORMED IS CONTAINED IN SECTION B.3
C              OF APPENDIX B OF THE ENGINEERING DOCUMENT. THE RELEVANT
C              EQUATIONS ARE (B.60), (B.61), (B.68) AND (B.69). THE
C              RELEVANT PROCEDURES ARE 4 AND 5. THE ROUTINE ALSO
C              COMPUTES THE ARCTANGENT TERMS OF STEP 1 (EQUATION (B.41))
C              OF PROCEDURE 1. THE CODE CLOSELY FOLLOWS THE DEVELOPMENT
C              AND NOTATION OF SECTION B.3. NOTE THAT FNK(N,K)=F(1,N,K).
C******
      LOGICAL        LMXQ2,LMXQ3,LMXQ4,LMXK3,LMXK5,LMKEX
      COMMON/SKAICL/LMXQ2,LMXQ3,LMXQ4,LMXK3,LMXK5,LMKEX
      COMMON/SKAIC1/AKS1,AET1,AKS2,AET2,DRM,EL1,EL2,ELM,ANK,ANE,A,AA,
     CGG,S1,S2,S1I,S2I,HM,HH
      COMMON/SKAIC2/GAK(21),GKNK(5,5),GENK(5,5),GKMN(6,6),GEMN(6),
     CGAMN(6,6),H111,FK(37),FNK(6,5),FMN(6,6),E(37)
      COMMON/SKAICI/MXFK,MXFKN,MXFNK,MXKM2,MXKM4,MXQM1
      DATA DELFKS,NFK /.01,16/
C        EQUATIONS AND PROCEDURES REFERENCED IN THIS ROUTINE
C        ARE CONTAINED IN APPENDIX B OF THE ENGINEERING DOCUMENT
C
C        EXECUTE STEP 1 OF PROCEDURE 4 (EQUATION (B.60))
      IF(EL2) 10,10,20
 10   RATIO=(S1-EL1)/(S2-EL2)
      GO TO 50
 20   IF(EL1) 30,30,40
 30   RATIO=(S1-EL1)*(S2+EL2)/GG
      GO TO 50
 40   RATIO=(S2+EL2)/(S1+EL1)
 50   FK(1)=ALOG(RATIO)
```

```
      IF(LMKEX) 60,55
 55   C1=GG+HM*S1
      C2=GG+HM*S2
C        CALCULATE AND ACCUMULATE CONTRIBUTION OF QUADRILATERAL
C        SIDE TO ARCTANGENT TERM OF H(1,1,1) AS DEFINED BY
C        STEP 1 (EQUATION (B.41)) OF PROCEDURE 1
      H111=H111+A*FK(1)-HM*ATAN2(A*(C1*EL2-C2*EL1),C1*C2+AA*EL1*EL2)
 60   IF(MXFK.LT.3) GO TO 500
C        BRANCH TO PROCEDURE 5 OR 4 RESPECTIVELY
      IF(GG.LT.DELFKS*(ELM*ELM+GG)) 400,200
 200  CONTINUE
      MXFKN=MXFK
C        EXECUTE STEP 2 (EQUATION (B.61)) OF PROCEDURE 4
      CALL ECAL(S1I,S2I,EL1,EL2,E,MXFK-1)
      DO 250 K=3,MXFK,2
 250  FK(K)=(FLOAT(K-3)*FK(K-2)+E(K-1))/(FLOAT(K-2)*GG)
      GO TO 500
 400  CONTINUE
      MXFKN=MXFK+NFK
C        EXECUTE STEPS 1 AND 2 (EQUATION (B.68) AND (B.69))
C        OF PROCEDURE 5
      CALL ECAL(S1I,S2I,EL1,EL2,E,MXFKN-1)
      FK(MXFKN)=0.
      DO 450 KP=5,MXFKN,2
      K=MXFKN+5-KP
 450  FK(K-2)=(FLOAT(K-2)*GG*FK(K)-E(K-1))/FLOAT(K-3)
 500  CONTINUE
C        ACCUMULATE CONTRIBUTION OF QUADRILATERAL SIDE TO F
C        INTEGRALS FOR USE IN STEP 1 (EQUATION (B.41)) AND 2
C        (EQUATION (B.42)) OF PROCEDURE 1 AND STEP 2
C        (EQUATION (B.50)) OF PROCEDURE 2
      DO 600 K=1,MXFK,2
 600  GAK(K)=GAK(K)+A*FK(K)
 999  RETURN
      END
```

```
      SUBROUTINE GCPCAL(NM,NN,NM1,NN1,ZM,ZA)
C******
C     SUBROUTINE GCPCAL (NM,NN,NM1,NN1,ZM,ZA)
C
C     PURPOSE  TO CONSTRUCT AN NM+1 BY NN+1 GRID OF POINTS FROM CORNER
C              POINT DATA
C
C     INPUT    CALLING SEQUENCE
C              NM - NUMBER OF CORNER POINTS IN A ROW
C              NN - NUMBER OF CORNER POINTS IN A COLUMN
C              NM1 - NUMBER OF GRID POINTS IN A ROW (NM+1)
C              NN1 - NUMBER OF GRID POINTS IN A COLUMN (NN+1)
C              ZM - COORDINATES OF CORNER POINTS
C
C     OUTPUT   CALLING SEQUENCE
C              ZA - COORDINATES OF GRID POINTS
C
C     SUBROUTINES
C     CALLED   NONE
C
C     DISCUSSION  THE ROUTINE COMPUTES AN NM+1 BY NN+1 GRID OF POINTS
C              DERIVED FROM CORNER POINT DATA. THE POINTS IN THE GRID
C              CONSIST OF THE AVERAGE OF EACH SET OF FOUR ADJACENT
C              CORNER POINTS, THE AVERAGE OF EACH SET OF TWO ADJACENT
C              EDGE CORNER POINTS AND THE FOUR EXTREME CORNER POINTS.
C              THESE POINTS ARE OBTAINED BY COMPUTING APPROXIMATE
C              AVERAGES OF THE CORNER POINTS.
C******
      DIMENSION ZM(3,NM,NN),ZA(3,NM1,NN1)
      DO 99 N=1,NN1
      N1=MAX0(N-1,1)
      N2=MIN0(N,NN)
      DO 98 M=1,NM1
      M1=MAX0(M-1,1)
      M2=MIN0(M,NM)
      DO 90 L=1,3
      ZA(L,M,N)=.25*(ZM(L,M1,N1)+ZM(L,M2,N1)+ZM(L,M1,N2)+ZM(L,M2,N2))
   90 CONTINUE
   98 CONTINUE
   99 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE GEOMC(NT,NM,NN,NPA,ZM)
C******
C     SUBROUTINE GEOMC (NT,NM,NN,NPA,ZM)
C
C     PURPOSE  TO CALCULATE GEOMETRIC DEFINING QUANTITIES FOR EACH PANEL
C              IN A NETWORK
C
C     INPUT    CALLING SEQUENCE
C              NT - NETWORK TYPE
C              NM - NUMBER OF SPANWISE CUTS IN NETWORK
C              NN - NUMBER OF TRANSVERSE CUTS IN NETWORK
C              NPA - TOTAL NUMBER OF PANELS IN ALL PREVIOUS NETWORKS
C              ZM - COORDINATES OF CORNER POINTS IN THE NETWORK
C
C              COMMON BLOCK
C              /IPRINT/ - IPGEOM
C
C     OUTPUT   COMMON BLOCK
C              /PANDQ/ - CP,PC,RO,AR,ART,P,A,B,DIAM,C
C
C     SUBROUTINES
C     CALLED   SURFIT,CCAL,IPTRNS
C
C     DISCUSSION  THE ROUTINE CALCULATES AND STORES GEOMETRIC DEFINING
C              QUANTITIES FOR EACH PANEL OF A NETWORK. FIRST THE FOUR
C              GRID POINTS DEFINING THE PANEL CORNER POINTS ARE FOUND.
C              TOGETHER WITH ADJACENT GRID POINTS THESE CORNER POINTS
C              ARE FED INTO SURFIT WHICH DEFINES THE ACTUAL PANEL
C              SURFACE AND THE LOCAL PANEL COORDINATE SYSTEM. THEN CCAL
C              IS CALLED TO CALCULATE PANEL MOMENTS. FINALLY, ALL THE
C              PANEL DEFINING QUANTITIES ARE STORED ON A FILE.
C******
      COMMON/LSQSFC/ZK(3,16),WTK(16),AK(6,16),NO,NPK
      COMMON/PANDQ/CP(3,4),PC(3),RO(3),AR(3,3),ART(3,3),P(2,4),A,B,DIAM,
     CC(6,6),AST(6,16),IIS(16),INS,ITS,NPDQ
      COMMON /IPRINT/IPNPUT,IPGEOM,IPSING,IPCNTP,IPOUTP
      DIMENSION ZM(3,NM,NN)
      DATA WT /1.E5/
      IF(IPGEOM.NE.0)  PRINT 1001
 1001 FORMAT(1H1)
C        CYCLE THROUGH ALL PANELS IN THE NETWORK
      DO 199 N=2,NN
      DO 198 M=2,NM
      IP=M-1+(NM-1)*(N-2)+NPA
C        ASSEMBLE FOUR GRID POINTS DEFINING THE PANEL
      DO 110 L=1,3
      CP(L,1)=ZM(L,M-1,N-1)
      CP(L,2)=ZM(L,M-1,N)
      CP(L,3)=ZM(L,M,N)
      CP(L,4)=ZM(L,M,N-1)
  110 CONTINUE
C        ASSEMBLE ADJACENT GRID POINTS AND RESPECTIVE
```

```
C          WEIGHTS FOR DEFINING CURVED PANEL SURFACE
      NPK=0
      DO 129 J=1,4
      NJ=MINO(MAXO(N+J-3,1),NN)
      DO 128 I=1,4
      MI=MINO(MAXO(M+I-3,1),NM)
      NPK=NPK+1
      WTK(NPK)=1.
      IF(((I.EQ.2).OR.(I.EQ.3)).AND.((J.EQ.2).OR.(J.EQ.3))) WTK(NPK)=WT
      DO 120 L=1,3
      ZK(L,NPK)=ZM(L,MI,NJ)
      IF((NN.EQ.2).AND.((J.EQ.1).OR.(J.EQ.4)))
     CZK(L,NPK)=.5*(ZM(L,MI,1)+ZM(L,MI,2))
      IF((NM.EQ.2).AND.((I.EQ.1).OR.(I.EQ.4)))
     CZK(L,NPK)=.5*(ZM(L,1,NJ)+ZM(L,2,NJ))
 120  CONTINUE
 128  CONTINUE
 129  CONTINUE
      IF(IPGEOM.NE.0)  PRINT 2002, IP
 2002 FORMAT(//* PANEL*,I4)
C          DEFINE PANEL SURFACE
      CALL SURFIT
      CALL TRANS(AR,ART,3,3)
C          CALCULATE PANEL CHARACTERISTIC LENGTH
      D13=SQRT((P(1,3)-P(1,1))**2+(P(2,3)-P(2,1))**2)
      D24=SQRT((P(1,4)-P(1,2))**2+(P(2,4)-P(2,2))**2)
      DIAM=AMAX1(D13,D24)
C          CALCULATE PANEL MOMENTS FOR LATER USE IN
C          INFLUENCE COEFFICIENT CALCULATIONS
      CALL CCAL(P,C)
C          STORE PANEL DEFINING QUANTITIES ON A FILE
      CALL IPTRNS(IP)
      IF(IPGEOM.NE.0)  PRINT 1002, CP,PC,RO,AR,P,A,B,DIAM
 1002 FORMAT(6E15.6)
 198  CONTINUE
 199  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE GRDIND(NM,NN,Z,I,IS)
C******
C     SUBROUTINE GRDIND (NM,NN,Z,I,IS)
C
C     PURPOSE   TO ORDER NON-IDENTICAL POINTS OF AN NM BY NN GRID OF
C               POINTS VIA AN INDEX ARRAY
C
C     INPUT     CALLING SEQUENCE
C               NM - NUMBER OF GRID POINTS IN A ROW
C               NN - NUMBER OF GRID POINTS IN A COLUMN
C               Z - COORDINATES OF GRID POINTS
C
C     OUTPUT    CALLING SEQUENCE
C               I - INDEX ARRAY CONTAINING SEQUENCE NUMBER OF EACH GRID
C                   POINT
C               IS - TOTAL NUMBER OF NON-IDENTICAL POINTS IN A GRID
C
C     SUBROUTINES
C     CALLED    PIDENT
C
C     DISCUSSION  THE ROUTINE SEQUENCES AN NM BY NN GRID OF POINTS.
C               THE SEQUENCING PROCEEDS IN THE ORDER ((M=1,NM),N=1,NN)
C               WHERE (M,N) IS THE POINT IN ROW M AND COLUMN N. ANY POINT
C               IDENTICAL WITH THE POINT IN THE SAME ROW AND PREVIOUS
C               COLUMN OR WITH THE POINT IN THE SAME COLUMN AND PREVIOUS
C               ROW IS ASSIGNED THE SAME SEQUENCE NUMBER AS THAT POINT.
C               THE SEQUENCE NUMBERS OF THE GRID POINTS ARE STORED IN AN
C               NM X NN INDEX ARRAY AND RETURNED AS OUTPUT ALONG WITH THE
C               TOTAL NUMBER OF NON-IDENTIFIED POINTS.
C******
      LOGICAL IDENT
      DIMENSION Z(3,NM,NN),I(NM,NN)
C        INITIALIZE SEQUENCE NUMBER
      IS=0
C        CYCLE THROUGH GRID POINTS COLUMN-WISE
      DO 99 N=1,NN
C        CYCLE THROUGH COLUMN ROW-WISE
      DO 98 M=1,NM
      IDENT=.FALSE.
C        CHECK IDENTITY WITH POINT IN SAME COLUMN AS PREVIOUS ROW
      IF(M.GT.1) CALL PIDENT(Z(1,M,N),Z(1,M-1,N),IDENT)
      IF(IDENT) I(M,N)=I(M-1,N)
      IF(IDENT) GO TO 98
C        CHECK IDENTITY WITH POINT IN SAME ROW AND PREVIOUS COLUMN
      IF(N.GT.1) CALL PIDENT(Z(1,M,N),Z(1,M,N-1),IDENT)
      IF(IDENT) I(M,N)=I(M,N-1)
      IF(IDENT) GO TO 98
C        BUMP SEQUENCE NUMBER IF POINT IS NEW
      IS=IS+1
      I(M,N)=IS
   98 CONTINUE
   99 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE INTCAL
C******
C     SUBROUTINE INTCAL
C
C     PURPOSE  TO COMPUTE THE H INTEGRALS INVOLVED IN THE FORMULAS FOR
C              THE SOURCE AND DOUBLET PANEL INDUCED VELOCITY INFLUENCE
C              COEFFICIENTS. (SEE SECTION B.3 OF APPENDIX B OF THE
C              ENGINEERING DOCUMENT.)
C
C     INPUT    COMMON BLOCK
C              /INTQ/ - MXQ,MXK
C              /PIVINT/ - X,P,AC,BC,DIAM
C
C     OUTPUT   COMMON BLOCK
C              /INTQ/ - H,HZ,IH
C
C     SUBROUTINES
C     CALLED   SIDECL,ZERO,TRNSFR,FKCAL,FMNCAL,FNKCAL
C
C     DISCUSSION  THE ROUTINE CALCULATES THE INTEGRALS
C              H(M,N,K)=I(SIGMA,KSE**(M-1)*ETA**(N-1)/RHO**K,DKSE*DETA)
C              FOR M=1,MXQ AND N=1,MXQ-M+1 AND K=1,MXK,2.
C              A DESCRIPTION OF THE CALCULATIONS PERFORMED IS CONTAINED
C              IN SECTION B.3 OF APPENDIX B OF THE ENGINEERING
C              DOCUMENT. THE ROUTINE CAN BE DIVIDED INTO THREE PARTS.
C              IN THE FIRST PART PRELIMINARY QUANTITIES CONCERNING THE
C              GEOMETRIC RELATIONSHIP OF THE FIELD POINT TO THE
C              QUADRILATERAL ARE CALCULATED. IN THE SECOND PART THE
C              F INTEGRALS ARE CALCULATED FOR EACH SIDE OF THE
C              QUADRILATERAL AND ACCUMULATED. IN THE THIRD PART
C              PROCEDURE 1,2 OR 3 IS EXECUTED.
C******
      LOGICAL        LMXQ2,LMXQ3,LMXQ4,LMXK3,LMXK5,LMKEX
      COMMON/INTQ/H(6,6,7),HZ,IH,MXQ,MXK
      COMMON/PIVINT/X(3),P(2,4),AC,BC,DIAM,C(6,6),DV(3,6),NTST,NCF
      COMMON/SIDEQ/QSIDE(12,4)
      COMMON/SKAICL/LMXQ2,LMXQ3,LMXQ4,LMXK3,LMXK5,LMKEX
      COMMON/SKAIC1/AKS1,AET1,AKS2,AET2,DRM,EL1,EL2,ELM,ANK,ANE,A,AA,
     CGG,S1,S2,S1I,S2I,HM,HH
      COMMON/SKAIC2/GAK(21),GKNK(5,5),GENK(5,5),GKMN(6,6),GEMN(6),
     CGAMN(6,6),H111,FK(37),FNK(6,5),FMN(6,6),E(37)
      COMMON/SKAICI/MXFK,MXFKN,MXFNK,MXKM2,MXKM4,MXQM1
      DIMENSION Q(2)
      DATA DELTH /.01/
      DATA DELTHZ /1.E-8/
      DATA NHKEX /16/
C        EQUATIONS AND PROCEDURES REFERENCED IN THIS ROUTINE ARE
C        CONTAINED IN APPENDIX B OF THE ENGINEERING DOCUMENT
C
C        CALCULATE QUANTITIES ASSOCIATED WITH GEOMETRICAL RELATIONSHIP
C        OF FIELD POINT TO QUADRILATERAL
      CALL SIDECL(Q,DSMIN,D)
```

```
C          CALCULATE H OF EQUATION (B.14)
      HZ=X(3)-AC*Q(1)*Q(1)-BC*Q(2)*Q(2)
      IF(ABS(HZ).LT.DELTHZ*DIAM) HZ=0.
      HM=ABS(HZ)
      HH=HZ*HZ
      LMKEX=HM.LT.(DELTH*DSMIN)
      IH=0
      IF(LMKEX.AND.(D.EQ.0.)) IH=1
C          SET INDICES AND LOGICAL VARIABLES FOR FUTURE BRANCHES
      MXKM2=MXK-2
      MXKM4=MXK-4
      MXQM1=MXQ-1
      LMXQ2=MXQ.LT.2
      LMXQ3=MXQ.LT.3
      LMXQ4=MXQ.LT.4
      LMXK3=MXK.LT.3
      LMXK5=MXK.LT.5
      MXFK=MXK-2
      IF(LMKEX) MXFK=MXFK+NHKEX
      CALL ZERO(GAK,150)
C          CALCULATE AND ACCUMULATE F INTEGRALS OVER FOUR SIDES
C          OF QUADRILATERAL
      DO 500 IS=1,4
      CALL TENSFR(QSIDE(1,IS),AKS1,12)
C           IGNORE SIDE IF LENGTH IS ZERO
      IF(DPM.EQ.0.) GO TO 500
C          CALCULATE FURTHER QUANTITIES ASSOCIATED WITH
C          RELATIONSHIP OF FIELD POINT TO QUADRILATERAL
      GG=AA+HH
      S1S=EL1*EL1+GG
      S2S=EL2*EL2+GG
      S1=SQRT(S1S)
      S2=SQRT(S2S)
      S1I=1./S1
      S2I=1./S2
C          CALCULATE F(1,1,K) INTEGRALS
      CALL FKCAL
      IF(LMXQ2) GO TO 500
C          CALCULATE F(M,N,1) INTEGRALS
      CALL FMNCAL
      IF(LMXK3) GO TO 500
C          CALCULATE F(1,N,K) INTEGRALS
      CALL FNKCAL
  500 CONTINUE
C          BRANCH TO PROCEDURE 1 OR PROCEDURE 2
      IF(LMKEX) 675,625
C          EXECUTE STEP 1 OF PROCEDURE 1
  625 H(1,1,1)=H111
      IF(LMXK3) GO TO 700
C          EXECUTE STEP 2 (EQUATION (B.42)) OF PROCEDURE 1
      DO 650 K=3,MXK,2
  650 H(1,1,K)=(FLOAT(K-4)*H(1,1,K-2)+GAK(K-2))/(FLOAT(K-2)*HH)
```

```
      GO TO 700
C         EXECUTE STEP 1 (EQUATION (B.49)) OF PROCEDURE 2
  675 Z=0.
C         EXECUTE STEP 2 (EQUATION (B.50)) OF PROCEDURE 2
      DO 680 KP=2,NHKEX,2
      K=NHKEX+MXK-KP+2
  680 Z=(HH*FLOAT(K-2)*Z-GAK(K-2))/FLOAT(K-4)
      H(1,1,MXK)=Z
      IF(LMXK3) GO TO 700
      DO 690 KP=3,MXK,2
      K=MXK-KP+3
  690 H(1,1,K-2)=(HH*FLOAT(K-2)*H(1,1,K)-GAK(K-2))/FLOAT(K-4)
  700 IF(LMXQ2) GO TO 999
C         EXECUTE STEPS 3 AND 4 (EQUATION (B.43) AND (B.44))
C         OF PROCEDURE 1
      H(1,2,1)=.5*(HH*GEMN(1)+GAMN(1,2))
      MXN=MXQ-1
      DO 760 N=1,MXN
  760 H(2,N,1)=(HH*GKMN(1,N)+GAMN(2,N))/FLOAT(N+1)
      IF(LMXQ3) GO TO 800
      DO 770 N=3,MXQ
  770 H(1,N,1)=(HH*(-FLOAT(N-2)*H(1,N-2,1)+GEMN(N-1))+GAMN(1,N))/
     CFLOAT(N)
C         EXECUTE STEP 5 (EQUATION (B.45)) OF PROCEDURE 1
      DO 790 M=3,MXQ
      MXN=MXQ-M+1
      DO 780 N=1,MXN
  780 H(M,N,1)=(HH*(-FLOAT(M-2)*H(M-2,N,1)+GKMN(M-1,N))+GAMN(M,N))/
     CFLOAT(M+N-1)
  790 CONTINUE
  800 IF(LMXK3) GO TO 999
C         EXECUTE STEPS 5 AND 6 (EQUATIONS (B.46) AND (B.47))
C         OF PROCEDURE 1
      DO 890 K=3,MXK,2
      FACTK=1./FLOAT(K-2)
      MXN=MXQ-1
      DO 870 N=1,MXN
  870 H(2,N,K)=-FACTK*GKNK(N,K-2)
      H(1,2,K)=-FACTK*GENK(1,K-2)
      IF(LMXQ3) GO TO 890
      DO 880 N=3,MXQ
  880 H(1,N,K)= FACTK*(FLOAT(N-2)*H(1,N-2,K-2)-GENK(N-1,K-2))
  890 CONTINUE
  900 IF(LMXQ3) GO TO 999
C         EXECUTE STEP 8 (EQUATION (B.48) OF PROCEDURE 1
      DO 990 K=3,MXK,2
      DO 990 M=3,MXQ
      MXN=MXQ-M+1
      DO 990 N=1,MXN
  990 H(M,N,K)=-H(M-2,N+2,K)-HH*H(M-2,N,K)+H(M-2,N,K-2)
  999 RETURN
      END
```

```
      SUBROUTINE KSORT(A,M,N,KEY,W)
C******
C     SUBROUTINE KSORT
C
C     PURPOSE   TO SORT THE COLUMN OF A TWO-DIMENSIONAL ARRAY USING THE
C               GIVEN KEY INDEX ARRAY
C
C     INPUT     CALLING SEQUENCE
C               A - ARRAY OF WHICH THE COLUMN IS TO BE SORTED
C               M - NUMBER OF ROWS OF A
C               N - NUMBER OF COLUMNS OF A
C               KEY - ARRAY CONSISTS OF GIVEN KEY INDICES
C               W - WORKING ARRAY OF SAME DIMENSION AS A
C
C     OUTPUT    CALLING SEQUENCE
C               A - THE SORTED ARRAY
C
C     SUBROUTINES
C     CALLED    NONE
C
C     DISCUSSION  THE CONTENTS OF ARRAY A ARE STORED IN A WORKING ARRAY
C               USING THE INDICES GIVEN IN KEY ARRAY. WORKING ARRAY IS
C               THEN TRANSFERED BACK TO ARRAY A.
C******
      DIMENSION A(M,1),KEY(1),W(M,1)
      DO 10 J=1,N
      K = KEY(J)
      DO 10 I=1,M
   10 W(I,K) = A(I,J)
      DO 20 J=1,N
      DO 20 I=1,M
   20 A(I,J) = W(I,J)
      RETURN
      END
```

```
      SUBROUTINE LSQSF
C******
C     SUBROUTINE LSQSF
C
C     PURPOSE  TO FIND THE GENERALIZED INVERSE FROM A LEAST SQUARES FIT
C
C     INPUT    COMMON BLOCK
C              /LSQSFC/ - ZK,WTK,NO,NPK
C
C     OUTPUT   COMMON BLOCK
C              /LSQSFC/ - AK
C
C     SUBROUTINES
C     CALLED   TRANS,MMULT,PDSEQS
C
C     DISCUSSION  THE ROUTINE FIRST FORMS THE WEIGHTED NORMAL EQUATIONS.
C              IT THEN CALLS ROUTINE USING CHOLESKY SCHEME TO SOLVE THE
C              SYSTEM OF EQUATIONS AND FINDS THE GENERALIZED INVERSE. IF
C              THE SYSTEM OF EQUATIONS IS NOT POSITIVE DEFINITE, AN ERR-
C              OR MESSAGE WILL BE PRINTED AND EXECUTION OF THE COMPUTER
C              PROGRAM WILL BE TERMINATED.
C******
      COMMON/LSQSFC/ZK(3,16),WTK(16),AK(6,16),NO,NPK
      DIMENSION V(96),C(96),B(36)
      NI=6
      IF((NO.LT.2).OR.(NPK.LT.6)) NI=3
      IF((NO.LT.1).OR.(NPK.LT.3)) NI=1
C                                FORMS A RECTANGULAR SYSTEM OF EQUATIONS V
C                                FROM LEAST SQUARES FIT
      DO 250 K=1,NPK
      L=NI*(K-1)
      V(L+1)=1.
      IF(NI.LT.2) GO TO 200
      V(L+2)=ZK(1,K)
      V(L+3)=ZK(2,K)
      IF(NI.LT.4) GO TO 200
      V(L+4)=.5*ZK(1,K)*ZK(1,K)
      V(L+5)=ZK(1,K)*ZK(2,K)
      V(L+6)=.5*ZK(2,K)*ZK(2,K)
  200 CONTINUE
C                                MULTIPLIES V BY A DIAGONAL MATRIX WTK
C                                CONSISTS OF GIVEN WEIGHTS
      DO 225 I=1,NI
      L=I+NI*(K-1)
      C(L)=WTK(K)*V(L)
  225 CONTINUE
  250 CONTINUE
C                                FORMS THE WEIGHTED NORMAL EQUATIONS
      CALL TRANS(V,AK,NI,NPK)
      CALL MMULT(C,AK,B,NI,NPK,NI)
C                                CALLS ROUTINE USING CHOLESKY SCHEME TO
C                                SOLVE THE NORMAL EQUATIONS AND OBTAINS
```

```
C                                   THE GENERALIZED INVERSE
      CALL PDSEQS(B,NI,NI,V,C,NPK,D1)
      IF(D1.NE.0.0)  GO TO 350
      PRINT 300
  300 FORMAT(//* NORMAL EQUATIONS APPEARS SINGULAR*)
      RETURN
  350 CONTINUE
C                                   STORES THE GENERALIZED INVERSE IN ARRAY AK
      DO 499 K=1,NPK
      DO 475 I=1,6
 475  AK(I,K)=0.
      DO 450 I=1,NI
      L=I+NI*(K-1)
      AK(I,K) = C(L)
 450  CONTINUE
 499  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PIDENT(P,Q,IDENT)
C******
C     SUBROUTINE PIDENT (P,Q,IDENT)
C
C     PURPOSE  TO DETERMINE WHETHER THE POINTS P AND Q ARE TO BE
C              CONSIDERED NUMERICALLY IDENTICAL
C
C     INPUT    CALLING SEQUENCE
C              P - COORDINATES OF FIRST POINT
C              Q - COORDINATES OF SECOND POINT
C
C     OUTPUT   CALLING SEQUENCE
C              IDENT - LOGICAL VARIABLE EQUAL TO TRUE IF P AND Q ARE
C                      CONSIDERED IDENTICAL AND FALSE OTHERWISE
C
C     SUBROUTINES
C     CALLED   NONE
C
C     DISCUSSION  THE ROUTINE DETERMINES WHETHER THE POINTS P AND Q ARE
C              CONSIDERED NUMERICALLY IDENTICAL. THE CRITERIA FOR
C              IDENTITY IS THAT THE DISTANCE FROM P TO Q MUST BE SMALLER
C              THAN OR EQUAL TO 1.E-12 TIMES THE SUM OF THE LENGTHS OF
C              P AND Q.
C******
      LOGICAL IDENT
      DIMENSION P(3),Q(3)
      DATA DELTA /1.E-12/
      SCALE=SQRT(P(1)**2+P(2)**2+P(3)**2)+SQRT(Q(1)**2+Q(2)**2+Q(3)**2)
      IDENT=((P(1)-Q(1))**2+(P(2)-Q(2))**2+(P(3)-Q(3))**2).LE.
     C((SCALE*DELTA)**2)
      RETURN
      END
```

```
      SUBROUTINE PIVC(Z)
C******
C     SUBROUTINE PIVC
C
C     PURPOSE  TO OBTAIN DOUBLET PANEL INFLUENCE COEFFICIENTS FOR A GIV-
C              EN CONTROL POINT
C
C     INPUT    CALLING SEQUENCE
C              Z - X,Y,Z COORDINATES OF A GIVEN CONTROL POINT
C              COMMON BLOCK
C              /PANDQ/ - RO,AR,ART,P,A,B,DIAM,C
C              /SYMM/ - NSYMM
C              /ZIP/ - IPZ,IP
C
C     OUTPUT   COMMON BLOCK
C              /PIVM/ - DVDS
C
C     SUBROUTINES
C     CALLED   UNIPAN,DIPV,MMULT
C
C     DISCUSSION  THE ROUTINE FIRST TRANSFERS SOME OF PANEL INFORMATION
C              TO BE USED BY THE INTEGRATION ROUTINE. IT THEN CALLS THE
C              INTEGRATION ROUTINE TO PROVIDE INFLUENCE COEFFICIENTS FOR
C              A GIVEN CONTROL POINT INDUCED BY DOUBLET DISTRIBUTION OF
C              THE SPECIFIED PANEL AND ITS IMAGE (WHEN NSYMM IS SET TO 1
C              ). THE INFLUENCE COEFFICIENTS ARE MODIFIED TO ACCOUNT FOR
C              THE CASE WHEN THE GIVEN CONTROL POINT IS LOCATED ON THE
C              INFLUENCING PANEL ITSELF (SEE ENGINEERING DOCUMENT - AERO
C              -DYNAMIC INFLUENCE COEFFICIENTS).
C******
      COMMON/PANDQ/CP(3,4),PC(3),RO(3),AR(3,3),ART(3,3),P(2,4),A,B,DIAM,
     CC(6,6),AST(6,16),IIS(16),INS,ITS,NPDQ
      COMMON /PIVINT/XX(3),PP(8),AA,BB,DDIAM,CC(36),DVDV(3,6),NTST,NCF
      COMMON/PIVM/DVDS(3,6)
      COMMON/SYMM/NSYMM
      COMMON /ZIP/IPZ,IP,ITZ,JCZ
      DIMENSION Z(3),W(3),DVS(3,6),GDVDV(3,6)
C                             SETS NUMBER OF TERMS OF LEAST SQUARES FIT
C                             FOR DOUBLET DISTRIBUTION
      NTST = 6
C                             TRANSFERS SOME OF PANEL INFORMATION TO BE
C                             USED BY THE INTEGRATION ROUTINE
      AA=A
      BB=B
      DDIAM = DIAM
      DO 20 I=1,36
 20   CC(I)=C(I)
      DO 100 I=1,8
 100  PP(I)=P(I)
C                             SETS ARRAYS DVS AND DVDS TO ZERO
      CALL ZERO(DVS,18)
      CALL ZERO(DVDS,18)
```

```
C                                  OBTIANS INFLUENCE COEFFICIENTS FOR A
C                                  GIVEN CONTROL POINT INDUCED BY DOUBLET
C                                  DISTRIBUTION OF THE SPECIFIED PANEL AND
C                                  ITS IMAGE (WHEN NSYMM=1)
      W(1)=Z(1)
      W(3)=Z(3)
      DO 200 ISYMM=1,2
      SIGN=3-2*ISYMM
      W(2)=SIGN*Z(2)
      CALL UNIPAN(AR,RO,W,XX)
      IF(ITS.EQ.2) CALL DPIV
      CALL MMULT(ART,DVDV,GDVDV,3,3,6)
      IF(ISYMM.NE.1.OR.IP.NE.IPZ)  GO TO 160
C                                  TO ACCOUNT FOR THE CASE WHEN THE GIVEN
C                                  CONTROL POINT IS LOCATED ON THE INFLUENC-
C                                  ING PANEL ITSELF
      HALF = -0.5
      HXI  = HALF*XX(1)
      HETA = HALF*XX(2)
      DVS(1,2) = HALF
      DVS(1,4) = HXI
      DVS(1,5) = HETA
      DVS(2,3) = HALF
      DVS(2,5) = HXI
      DVS(2,6) = HETA
      CALL MMULT(ART,DVS,DVDS,3,3,6)
  160 CONTINUE
      DO 175 I=1,6
      DVDS(1,I) = DVDS(1,I) + GDVDV(1,I)
      DVDS(2,I) = DVDS(2,I) + SIGN*GDVDV(2,I)
      DVDS(3,I) = DVDS(3,I) + GDVDV(3,I)
 175  CONTINUE
      IF(NSYMM.EQ.0) GO TO 400
 200  CONTINUE
 400  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SIDECL(W,DSMIN,D)
C******
C     SUBROUTINE SIDECL (W,DSMIN,D)
C
C     PURPOSE   TO COMPUTE GEOMETRIC QUANTITIES ASSOCIATED WITH THE
C               RELATIONSHIP OF THE FIELD POINT TO THE QUADRILATERAL
C               SIGMA FOR USE IN COMPUTING THE H INTEGRALS.(SEE FIGURE 30
C               AND SECTION B.3 OF APPENDIX B OF THE ENGINEERING
C               DOCUMENT.)
C
C     INPUT     COMMON BLOCK
C               /PIVINT/ - X,P
C
C     OUTPUT    CALLING SEQUENCE
C               W - POINT ON QUADRILATERAL CLOSEST TO PROJECTION OF FIELD
C                   POINT ONTO QUADRILATERAL PLANE
C               DSMIN - MINIMUM DISTANCE OF PROJECTION OF FIELD POINT
C                       ONTO QUADRILATERAL PLANE TO PERIMETER OF
C                       QUADRILATERAL
C               D - DISTANCE FROM W TO PROJECTION OF FIELD POINT ONTO
C                   QUADRILATERAL PLANE
C
C               COMMON BLOCK
C               /SIDEQ/ - QSIDE
C               /SKAIC1/ - AKS1,AET1,AKS2,AET2,DRM,EL1,EL2,ELM,ANK,ANE,
C                          A,AA
C
C     SUBROUTINES
C     CALLED    TRNSFR
C
C     DISCUSSION  THE ROUTINE COMPUTES GEOMETRIC QUANTITIES ASSOCIATED
C               WITH THE RELATIONSHIP OF THE QUADRILATERAL SIGMA TO THE
C               PROJECTION OF THE FIELD POINT ONTO THE QUADRILATERAL
C               PLANE. IN PARTICULAR THE ROUTINE DETERMINES WHETHER THE
C               PROJECTION LIES INSIDE OR OUTSIDE OF THE QUADRILATERAL
C               AS WELL AS CALCULATES THE MINIMUM DISTANCE FROM THE
C               PROJECTION TO THE PERIMETER OF THE QUADRILATERAL.
C               OTHER QUANTITIES COMPUTED INCLUDE THOSE QUANTITIES
C               DISPLAYED IN FIGURE 31 AND DISCUSSED IN SECTION
C               B.3 OF APPENDIX B OF THE ENGINEERING DOCUMENT. THE
C               QUANTITIES ASSOCIATED WITH THE QUADRILATERAL IN GENERAL
C               ARE RETURNED VIA THE CALL LIST WHEREAS THE QUANTITIES
C               ASSOCIATED WITH EACH SIDE OF THE QUADRILATERAL ARE
C               STORED IN A COMMON BLOCK ARRAY A SIDE AT A TIME.
C******
      COMMON/PIVINT/X(3),P(2,4),AC,BC,DIAM,C(6,6),DV(3,6),NTST,NCF
      COMMON/SIDEQ/QSIDE(12,4)
      COMMON/SKAIC1/AKS1,AET1,AKS2,AET2,DRM,EL1,EL2,ELM,ANK,ANE,A,AA,
     CGG,S1,S2,S1I,S2I,HM,HH
      DIMENSION W(2)
C        QUANTITIES AND PROCEDURES REFERENCED IN THIS ROUTINE ARE
C        DISCUSSED IN APPENDIX B OF THE ENGINEERING DOCUMENT
```

```
      D=0.
      NNCT=0
      NPST=0
      ISA=0
C        CYCLE THROUGH SIDES OF QUADRILATERAL
      DO 500 IS=1,4
      ISP1=MOD(IS,4)+1
C        CALCULATE QUANTITIES DISPLAYED IN FIGURE 31
      AKS1=P(1,IS)-X(1)
      AET1=P(2,IS)-X(2)
      AKS2=P(1,ISP1)-X(1)
      AET2=P(2,ISP1)-X(2)
      DKS=AKS2-AKS1
      DET=AET2-AET1
      DRM=SQRT(DKS*DKS+DET*DET)
C        IGNORE SIDE IF LENGTH IS ZERO
      IF(DRM.EQ.0.) GO TO 500
      ISA=ISA+1
      DRMI=1./DRM
      A=DRMI*(AKS1*AET2-AKS2*AET1)
      AA=A*A
      ANK=DRMI*DET
      ANE=-DRMI*DKS
      EL1=DRMI*(AKS1*DKS+AET1*DET)
      EL2=DRMI*(AKS2*DKS+AET2*DET)
C        COMPUTE INCREMENT OF INTEGERS WHICH WILL EVENTUALLY
C        DETERMINE WHETHER THE FIELD POINT PROJECTION ONTO THE
C        QUADRILATERAL PLANE LIES INSIDE OF OUTSIDE THE QUADRILATERAL
      IF(A.GT.0.) NPST=NPST+1
      B=AKS1*AKS2+AET1*AET2
      IF((A.GT.0.).AND.(B.LT.0.)) NNCT=NNCT+1
C        CALCULATE MINIMUM DISTANCE FROM FIELD POINT TO QUADRILATERAL
C        SIDE
      IF(EL1*EL2) 75,75,85
 75   ELM=0.
      GO TO 90
 85   ELM=SIGN(AMIN1(ABS(EL1),ABS(EL2)),EL1)
 90   DIS=ELM*ELM+AA
      IF((ISA.GT.1).AND.(DIS.GT.D)) GO TO 500
      D=DIS
      ISS=IS
C        STORE CALCULATED QUANTITIES FOR EACH SIDE
 500  CALL TENSFR(AKS1,QSIDE(1,IS),12)
      D=SQRT(D)
      DSMIN=D
C        BRANCH TO 700 OF 800 DEPENDING UPON WHETHER FIELD POINT
C        PROJECTION LIES INSIDE OR OUTSIDE QUADRILATERAL
      IF((NPST.EQ.4).OR.((NPST.EQ.3).AND.(NNCT.GE.2))) 700,800
 700  D=0.
      W(1)=X(1)
      W(2)=X(2)
      RETURN
C        RETRIEVE CALCULATED QUANTITIES FOR SIDE CONTAINING POINT
```

```
C           CLOSEST TO FIELD POINT PROJECTION AND CALCULATE
C           COORDINATES OF THAT POINT
 800  CALL TRNSFR(OSIDE(1,ISS),AKS1,12)
      W(1)=X(1)+A*ANK-ELM*ANE
      W(2)=X(2)+A*ANE+ELM*ANK
      RETURN
      END
```

```
      SUBRCUTINE SING(NT,NM,NN,NS,NSA,NPA,ZM)
C******
C     SUBRCUTINE SING (NT,NM,NN,NS,NSA,NPA,ZM)
C
C     PURPOSE  TO CALCULATE THE SINGULARITY DISTRIBUTION DEFINING
C              QUANTITIES FOR A GIVEN NETWORK
C
C     INPUT    CALLING SEQUENCE
C              NT - NETWORK TYPE
C              NM - NUMBER OF SPANWISE CUTS IN THE NETWORK
C              NN - NUMBER OF TRANSVERSE CUTS IN THE NETWORK
C              NSA - TOTAL NUMBER OF SINGULARITY PARAMETERS IN ALL
C                    PREVIOUS NETWORKS
C              NPA - TOTAL NUMBER OF PANELS IN ALL PREVIOUS NETWORKS
C              ZM - COORDINATES OF CORNER POINTS IN THE NETWORK
C
C              COMMON BLOCK
C              /IPRINT/ - IPSING
C              /PANDQ/ - FC,AR
C
C     OUTPUT   CALLING SEQUENCE
C              NS - NUMBER OF SINGULARITY PARAMETERS IN THE NETWORK
C
C              COMMON BLOCK
C              /PANDQ/ - AST,IIS,INS,ITS
C
C     SUBROUTINES
C     CALLED   GCPCAL,GRDIND,PTRNS,UNIPAN,LSQSF,IPTRNS
C
C     DISCUSSICN  THE ROUTINE CALCULATES THE DEPENDENCE OF EACH PANEL
C              SINGULARITY STRENGTH DISTRIBUTION ON THE FREE SINGULARITY
C              PARAMETERS OF THE NETWORK. SEPARATE COMPUTATIONS ARE
C              PERFORMED FOR EACH NETWORK TYPE. FIRST THE LOCATIONS OF
C              THE FREE SINGULARITY PARAMETERS ON THE NETWORK ARE
C              COMPUTED AND INDEXED. FOR EACH PANEL THE SINGULARITY
C              PARAMETERS AFFECTING THE DISTRIBUTION OF SINGULARITY
C              STRENGTH ON THAT PANEL ARE ISOLATED. EACH SUCH
C              PARAMETER IS ASSIGNED A WEIGHT (LARGE IF THE PARAMETER
C              ACTUALLY LIES ON THE PANEL). THE PANEL SINGULARITY
C              DISTRIBUTION IS THEN OBTAINED BY FITTING A QUADRATIC
C              FORM (IF THE SINGULARITY IS OF DOUBLET TYPE) TO THE
C              PARAMETERS BY THE METHOD OF LEAST SQUARES. THE MATRIX
C              WHICH RELATES THE COEFFICIENTS OF THE DISTRIBUTION TO
C              THE SINGULARITY PARAMETERS IS THEN STORED ON A FILE ALONG
C              WITH INDICES IDENTIFYING THE PARAMETERS.
C******
      COMMCN/LSQSFC/ZK(3,16),WTK(16),AK(6,16),NO,NPK
      COMMON/PANDQ/CP(3,4),PC(3),FC(3),AF(3,3),ART(3,3),P(2,4),A,B,DIAM,
     CC(6,6),AST(6,16),IIS(16),INS,ITS,NPDQ
      COMMCN /SKRCH1/ZA(3,175),IA(175)
      COMMCN /IPRINT/IPNPUT,IPGEOM,IPSING,IPCNTR,IPFIVC,IPOUTP
      DIMENSION ZM(3,NM,NN)
```

```
      DIMENSION ZPK(3)
      DATA WT /1.E5/
      IF(IPSING.NE.0)  PRINT 1001
 1001 FORMAT(1H1)
      NN1=NN+1
      NM1=NM+1
C        CALCULATE LOCATION OF SINGULARITY PARAMETERS
      CALL GCPCAL(NM,NN,NM1,NN1,ZM,ZA)
C        ORDER NON-IDENTICAL SINGULARITY PARAMETERS
      CALL GRDIND(NM1,NN1,ZA,IA,NIA)
C        TRANSFER TO CODE FOR APPROPRIATE NETWORK TYPE
      GO TO (100,200,300,400,500,600,600) NT
 100  CONTINUE
C        SOURCE/ANALYSIS NETWORK CALCULATIONS
C        (NOT AN OPTION IN PRESENT PROGRAM)
      DO 199 N=2,NN
      DO 198 M=2,NM
      IP=M-1+(NM-1)*(N-2)+NPA
      CALL PTRNS(IP)
      ITS=1
      NPK=0
      DO 129 J=1,3
      NJ=N+J-2
      IF((NJ.LT.2).OR.(NJ.GT.NN)) GO TO 129
      DO 128 I=1,3
      MI=M+I-2
      IF((MI.LT.2).OR.(MI.GT.NM)) GO TO 128
      NPK=NPK+1
      ITS(NPK)=MI-1+(NM-1)*(NJ-2)+NSA
      LMN=MI+NM1*(NJ-1)
      DO 125 L=1,3
      ZPK(L)=ZA(L,LMN)
 125  CONTINUE
      CALL UNIPAN(AR,RO,ZPK,ZPK)
      ZK(1,NPK)=ZPK(1)
      ZK(2,NPK)=ZPK(2)
      WTK(NPK)=1.
      IF((I.EQ.2).AND.(J.EQ.2)) WTK(NPK)=WT
 128  CONTINUE
 129  CONTINUE
      INS=NPK
      NO=1
      CALL LSQSF
      DO 149 K=1,NPK
      DO 148 I=1,3
      AST(I,K)=AK(I,K)
 148  CONTINUE
 149  CONTINUE
      CALL IPTRNS(IP)
 198  CONTINUE
 199  CONTINUE
      NS=(NM-1)*(NN-1)
```

```
      GO TO 800
  200 CONTINUE
C        DOUBLET/ANALYSIS (WING) NETWORK CALCULATIONS
C        CYCLE THROUGH ALL PANELS IN THE NETWORK
      DO 299 N=2,NN
      DO 298 M=2,NM
      IP=M-1+(NM-1)*(N-2)+NPA
C        RETRIEVE PANEL GEOMETRY DEFINING QUANTITIES
      CALL PTRNS(IP)
      ITS=2
      NPK=0
C        CALCULATE LOCATIONS OF SINGULARITY PARAMETERS
C        AFFECTING PANEL SINGULARITY DISTRIBUTION
      DO 229 J=1,3
      NJ=N+J-2
      DO 228 I=1,3
      MI=M+I-2
      LMN=MI+NM1*(NJ-1)
      NPK=NPK+1
      IIS(NPK)=IA(LMN)+NSA
      CALL UNIPAN(AR,RO,ZA(1,LMN),ZPK)
      ZK(1,NPK)=ZPK(1)
      ZK(2,NPK)=ZPK(2)
C        WEIGHT CONTRIBUTION OF SINGULARITY PARAMETER
      WTK(NPK)=1.
      IF(((MI.EQ.1).OR.(MI.EQ.NM1).OR.(I.EQ.2)).AND.
     C((NJ.EQ.1).OR.(NJ.EQ.NN1).OR.(J.EQ.2))) WTK(NPK)=WT
  228 CONTINUE
  229 CONTINUE
      INS=NPK
      NO=2
C        LEAST SQUARE PANEL SINGULARITY DISTRIBUTION
C        TO SINGULARITY PARAMETERS
      CALL LSQSF
      DO 249 K=1,NPK
      DO 248 I=1,6
      AST(I,K)=AK(I,K)
  248 CONTINUE
  249 CONTINUE
C        STORE SINGULARITY DEFINING QUANTITIES ON A FILE
      CALL IPTRNS(IP)
      IF(IPSING.NE.0)
     $WRITE(6,1000) IP,INS,IIS,NO,AST,ZK
  298 CONTINUE
  299 CONTINUE
      NS=NIA
      GO TO 800
C        SOURCE/DESIGN NETWORK CALCULATIONS
C        (NOT AN OPTION IN PRESENT PROGRAM)
  300 CONTINUE
      GO TO 800
  400 CONTINUE
```

```
C          DOUBLET/DESIGN (FREE SHEET) NETWORK CALCULATIONS
C          ORDER NON-IDENTICAL SINGULARITY PARAMETERS
      CALL GRDIND(NM,NN,ZM,IA,NIA)
C          CYCLE THROUGH ALL PANELS IN THE NETWORK
      DO 499 N=2,NN
      DO 498 M=2,NM
      IP=M-1+(NM-1)*(N-2)+NPA
C          RETRIEVE PANEL GEOMETRY DEFINING QUANTITIES
      CALL PTRNS(IP)
      ITS=2
      NPK=0
C          CALCULATE LOCATIONS OF SINGULARITY PARAMETERS
C          AFFECTING PANEL SINGULARITY DISTRIBUTION
      DO 429 J=1,4
      NJ=N+J-3
      IF((NJ.LT.1).OR.(NJ.GT.NN)) GO TO 429
      DO 428 I=1,4
      MI=M+I-3
      IF((MI.LT.1).OR.(MI.GT.NM)) GO TO 428
      LMN = MI + NM*(NJ-1)
      NPK=NPK+1
      IIS(NPK)=IA(LMN)+NSA
      CALL UNIPAN(AR,RO,ZM(1,MI,NJ),ZPK)
      ZK(1,NPK)=ZPK(1)
      ZK(2,NPK)=ZPK(2)
C          WEIGHT CONTRIBUTION OF SINGULARITY PARAMETER
      WTK(NPK)=1.
      IF(((I.EQ.2).OR.(I.EQ.3)).AND.((J.EQ.2).OR.(J.EQ.3))) WTK(NPK)=WT
  428 CONTINUE
  429 CONTINUE
      INS=NPK
      NO=2
C          LEAST SQUARE PANEL SINGULARITY DISTRIBUTION
C          TO SINGULARITY PARAMETERS
      CALL LSQSF
      DO 449 K=1,NPK
      DO 448 I=1,6
      AST(I,K)=AK(I,K)
  448 CONTINUE
  449 CONTINUE
C          STORE SINGULARITY DEFINING QUANTITIES ON A FILE
      CALL IPTRNS(IP)
      IF(IPSING.NE.0)
     $WRITE(6,1000) IP,INS,IIS,NO,AST,ZK
  498 CONTINUE
  499 CONTINUE
      NS=NIA
      GO TO 800
  500 CONTINUE
C          DOUBLET/DESIGN (WAKE) NETWORK CALCULATTIONS
C          CYCLE THROUGH ALL PANELS IN THE NETWORK
      DO 599 N=2,NN
```

```
      DO 598 M=2,NM
      IP=M-1+(NM-1)*(N-2)+NPA
C        RETRIEVE PANEL GEOMETRY DEFINING QUANTITIES
      CALL PTRNS(IP)
      ITS=2
      NPK=0
C        CALCULATE LOCATIONS OF SINGULARITY PARAMETERS
C        AFFECTING PANEL SINGULARITY DISTRIBUTION
      DO 529 J=1,3
      NJ=N+J-2
      DO 528 I=1,3
      MI=M+I-2
      LMN=MI+NM1*(NJ-1)
      NPK=NPK+1
      IIS(NPK)=IA(MI)+NSA
      CALL UNIPAN(AR,RO,ZA(1,LMN),ZPK)
      ZK(1,NPK)=ZPK(1)
      ZK(2,NPK)=ZPK(2)
C        WEIGHT CONTRIBUTION OF SINGULARITY PARAMETER
      WTK(NPK)=1.
      IF(((MI.EQ.1).OR.(MI.EQ.NM1).OR.(I.EQ.2)).AND.
     C((NJ.EQ.1).OR.(NJ.EQ.NN1).OR.(J.EQ.2))) WTK(NPK)=WT
  528 CONTINUE
  529 CONTINUE
      INS=NPK
      NO=2
C        LEAST SQUARE PANEL SINGULARITY DISTRIBUTION
C        TO SINGULARITY PARAMETERS
      CALL LSQSF
      DO 549 K=1,NPK
      DO 548 I=1,6
      AST(I,K)=AK(I,K)
  548 CONTINUE
  549 CONTINUE
C        STORE SINGULARITY DEFINING QUANTITIES ON A FILE
      CALL IPTRNS(IP)
      IF(IPSING.NE.0)
     $WRITE(6,1000) IP,INS,IIS,NO,AST,ZK
  598 CONTINUE
  599 CONTINUE
      NS=IA(NM1)
      GO TO 800
  600 CONTINUE
C        DOUBLET/DESIGN (FED SHEET) NETWORK CALCULATIONS
C        CYCLE THROUGH ALL PANELS IN THE NETWORK
      DO 699 N=2,NN
      DO 698 M=2,NM
      IP=M-1+(NM-1)*(N-2)+NPA
C        RETRIEVE PANEL GEOMETRY DEFINING QUANTITIES
      CALL PTRNS(IP)
      ITS=2
      NPK=0
```

```
C          CALCULATE LOCATIONS OF SINGULARITY PARAMETERS
C          AFFECTING PANEL SINGULARITY DISTRIBUTION
      DO 629 J=1,3
      NJ=N+J-2
      DO 628 I=1,3
      MI=M+I-2
      LMN=MI+NM1*(NJ-1)
      NPK=NPK+1
      IIS(NPK)=NJ+NSA
      IF(NT.EQ.7)  IIS(NPK) = 1 + NSA
      CALL UNIPAN(AR,RO,ZA(1,LMN),ZPK)
      ZK(1,NPK)=ZPK(1)
      ZK(2,NPK)=ZPK(2)
C          WEIGHT CONTRIBUTION OF SINGULARITY PARAMETER
      WTK(NPK)=1.
      IF(((MI.EQ.1).OR.(MI.EQ.NM1).OR.(I.EQ.2)).AND.
     C((NJ.EQ.1).OR.(NJ.EQ.NN1).OR.(J.EQ.2))) WTK(NPK)=WT
  628 CONTINUE
  629 CONTINUE
      INS=NPK
      NO=2
C          LEAST SQUARE PANEL SINGULARITY DISTRIBUTION
C          TO SINGULARITY PARAMETERS
      CALL LSQSF
      DO 649 K=1,NPK
      DO 648 I=1,6
      AST(I,K)=AK(I,K)
  648 CONTINUE
  649 CONTINUE
C          STORE SINGULARITY DEFINING QUANTITIES ON A FILE
      CALL IPTRNS(IP)
      IF(IPSING.NE.0)
     $WRITE(6,1000) IP,INS,IIS,NO,AST,ZK
 1000 FORMAT(//,19I5,/(6E15.6)//)
  698 CONTINUE
  699 CONTINUE
      NS=NN1
      IF(NT.EQ.7)  NS = 1
  800 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SURFIT
C******
C     SUBROUTINE SURFIT
C
C     PURPOSE  TO DEFINE PANEL SURFACE AND LOCAL PANEL COORDINATE SYSTEM
C
C     INPUT    COMMON BLOCK
C              /FLATP/ - NFLTP
C              /LSQSFC/ - ZK,WTK,NO,NPK
C              /PANDQ/ - CP
C              /IPRINT/ - IPGEOM
C
C     OUTPUT   COMMON BLOCK
C              /PANDQ/ - PC,PO,AR,P,A,B
C
C     SUBROUTINES
C     CALLED   CROSS,UVECT,TRANS,UNIPAN,LSQSF,MMULT
C
C     DISCUSSION  THE ROUTINE DEFINES A PANEL SURFACE AND LOCAL PANEL
C              COORDINATE SYSTEM. AS A FIRST APPROXIMATION TO THE PANEL
C              SURFACE THE ROUTINE TAKES THE QUADRILATERAL FORMED BY
C              PROJECTING THE PANEL CORNER POINTS ONTO THE PLANE THROUGH
C              THE MIDPOINTS OF THE LINE SEGMENTS JOINING THESE CORNER
C              POINTS. A LOCAL COORDINATE SYSTEM IS CONSTRUCTED WITH
C              THE ORIGIN AT THE AVERAGE OF THE QUADRILATERAL CORNER
C              POINTS AND WITH ONE AXIS NORMAL TO THE QUADRILATERAL. TO
C              OBTAIN A SECOND ORDER APPROXIMATION TO THE PANEL SURFACE
C              THE ROUTINE CALCULATES A PARABOLOID PASSING THROUGH THE
C              CORNER POINTS WITH CURVATURE OBTAINED BY LEAST SQUARING
C              THE PARABOLOID TO ADJACENT CORNER POINTS. THE LOCAL
C              COORDINATE SYSTEM IS THEN ROTATED AND TRANSLATED IN SUCH
C              A MANNER THAT THE PARABOLOID CAN BE REPRESENTED IN
C              CANONICAL FORM. AN ITERATIVE PROCESS IS REQUIRED TO
C              ELIMINATE LINEAR TERMS WITHOUT TRANSLATING THE ORIGIN.
C******
      COMMON/FLATP/NFLTP
      COMMON/LSQSFC/ZK(3,16),WTK(16),AK(6,16),NO,NPK
      COMMON/PANDQ/CP(3,4),PC(3),PO(3),AR(3,3),ART(3,3),P(2,4),A,B,DIAM,
     CC(6,6),AST(6,16),IIS(16),INS,ITS,NPDQ
      COMMON /IPRINT/IPNPUT,IPGEOM,IPSING,IPCNTR,IPEIVC,IPUUTP
      DIMENSION H(3,3),HT(3,3),U(3),V(3),W(3),ZETA(16),COF(6),POP(3)
      DIMENSION WK(3,16)
      EQUIVALENCE (U(1),HT(1)), (V(1),HT(4)), (W(1),HT(7))
      DATA NIT,DELTA /10,1.E-8/
C        CALCULATE BISECTING DIRECTIONS AND CENTER POINT
      DO 50 I=1,3
      U(I)=CP(I,1)+CP(I,4)-CP(I,2)-CP(I,3)
      V(I)=CP(I,1)+CP(I,2)-CP(I,3)-CP(I,4)
      PC(I)=.25*(CP(I,1)+CP(I,2)+CP(I,3)+CP(I,4))
      DO 49 K=1,NPK
      WK(I,K)=ZK(I,K)
 49   CONTINUE
```

```
 50    CONTINUE
C         CONSTRUCT ORTHOGONAL U,V,W COORDINATE SYSTEM
C         WITH W PERPENDICULAR TO PANEL PLANE
       CALL CROSS(U,V,W)
       CALL CROSS(W,U,V)
       CALL UVECT(U)
       CALL UVECT(V)
       CALL UVECT(W)
C         CALCULATE ORTHOGONAL MATRIX WHICH TRANSFORMS
C         COORDINATES FROM GLOBAL TO LOCAL
       CALL TRANS(HT,AR,3,3)
C         ITERATE TO FIND PANEL CURVATURES
       DO 170 IT=1,NIT
       DO 100 K=1,NPK
C         TRANSFORM ADJACENT CORNER POINTS TO LOCAL COORDINATES
       CALL UNIPAN(AR,PC,WK(1,K),ZK(1,K))
       ZETA(K)=ZK(3,K)
       IF(NFLTP.EQ.1) ZETA(K)=0.
 100   CONTINUE
       NO=2
C         CONSTRUCT LEAST SQUARES PARABOLOID THROUGH CORNER POINTS
       CALL LSQSF
       CALL MMULT(AK,ZETA,COF,6,NPK,1)
       IF(IPGEOM.NE.0)
      $WRITE(6,1000) COF
 1000 FORMAT(/ 4X5HCOF   ,(6F20.12))
C         ROTATE COORDINATE SYSTEM ABOUT NORMAL TO PANEL
C         TO GET RID OF QUADRATIC CROSS TERM
       COF46=.5*ABS(COF(4)-COF(6))
       DP=SQRT(COF(5)**2+COF46**2)
       IF(DP.EQ.(0.)) SPSI=0.
       IF(DP.NE.(0.)) SPSI=SQRT(.5*ABS(1.-COF46/DP))
       IF((COF(5)*(COF(4)-COF(6))).LT.(0.)) SPSI=-SPSI
       CPSI=SQRT(ABS(1.-SPSI**2))
       ART(1,1)=CPSI
       ART(2,1)=-SPSI
       ART(3,1)=0.
       ART(1,2)=SPSI
       ART(2,2)=CPSI
       ART(3,2)=0.
       ART(1,3)=0.
       ART(2,3)=0.
       ART(3,3)=1.
C         CALCULATE PRINCIPAL  CURVATURES
       A=.5*(COF(4)*CPSI**2+COF(6)*SPSI**2)+COF(5)*SPSI*CPSI
       B=.5*(COF(4)*SPSI**2+COF(6)*CPSI**2)-COF(5)*SPSI*CPSI
       D=COF(2)*CPSI+COF(3)*SPSI
       IF(ABS(D).LT.DELTA) D=0.
       E=-COF(2)*SPSI+COF(3)*CPSI
       IF(ABS(E).LT.DELTA) E=0.
       F=COF(1)
C         CALCULATE ORIGIN OF NEW LOCAL COORDINATE SYSTEM
```

```
      IF(A.EQ.(0.)) ROP(1)=0.
      IF(A.NE.(0.)) ROP(1)=-.5*D/A
      IF(B.EQ.(0.)) ROP(2)=0.
      IF(B.NE.(0.)) ROP(2)=-.5*E/B
      ROP(3)=COF(1)-A*ROP(1)**2-B*ROP(2)**2
      IF(IPGEOM.NE.0)
     $WRITE(6,2000) A,B,D,E,F,ROP
 2000 FORMAT((8E15.6))
C        ROTATE COORDINATE SYSTEM ABOUT AXES IN PANEL PLANE
C        TO TRY TO ELIMINATE LINEAR TERMS
      CA=1./SQRT(1.+D*D)
      SA=D*CA
      CB=1./SQRT(1.+E*E)
      SB=E*CB
      HT(1,1)=CA
      HT(1,2)=0.
      HT(1,3)=SA
      HT(2,1)=-SB*SA
      HT(2,2)=CB
      HT(2,3)=SB*CA
      HT(3,1)=-CB*SA
      HT(3,2)=-SB
      HT(3,3)=CB*CA
C        CALCULATE ORTHOGONAL TRANSFORMATION MATRIX
C        FOR NEW LOCAL COORDINATE SYSTEM
      CALL MMULT(ART,AR,H,3,3,3)
      CALL MMULT(HT,H,AR,3,3,3)
      IF((D.EQ.0.).AND.(E.EQ.0.)) GO TO 175
 170  CONTINUE
 175  CONTINUE
      CALL MMULT(ROP,AR,W,1,3,3)
C        CALCULATE ORIGIN OF LOCAL PANEL COORDINATE SYSTEM
      DO 200 I=1,3
 200  RO(I)=PC(I)+W(I)
C        CALCULATE LOCAL COORDINATES OF PROJECTIONS OF
C        CORNER POINTS ONTO PANEL PLANE
      DO 300 I=1,4
      CALL UNIPAN(AR,RO,CP(1,I),W)
      P(1,I)=W(1)
      P(2,I)=W(2)
 300  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SURPRO(Z,ZP,UN)
C******
C     SUBROUTINE SURPRO (Z,ZP,UN)
C
C     PURPOSE   TO FIND THE LOCATION OF THE PROJECTION OF A POINT ONTO
C               A PANEL SURFACE AS WELL AS THE SURFACE NORMAL AT THIS
C               LOCATION.
C
C     INPUT     CALLING SEQUENCE
C               Z - GLOBAL COORDINATES OF POINT TO BE PROJECTED
C
C               COMMON BLOCK
C               /PANDQ/ - RO,AR,ART
C
C     OUTPUT    CALLING SEQUENCE
C               ZP - GLOBAL COORDINATES OF LOCATION OF PROJECTION
C               UN - GLOBAL COORDINATES OF UNIT NORMAL TO PANEL SURFACE
C                    AT THIS LOCATION
C
C     SUBROUTINES
C     CALLED    UNIPAN,UVECT,PANUNI,MMULT
C
C     DISCUSSION  THE ROUTINE CALCULATES THE PROJECTION OF A POINT ONTO
C               A PANEL SURFACE AS WELL AS THE SURFACE NORMAL VECTOR AT
C               THE PROJECTED POINT. ALL INPUT AND OUTPUT VECTORS ARE
C               ASSUMED TO BE GIVEN IN GLOBAL COORDINATES. THE ROUTINE
C               CONVERTS TO LOCAL COORDINATES, PROJECTS AND CONVERTS
C               BACK TO GLOBAL COORDINATES. IN THE EVENT THAT THE GIVEN
C               POINT DOES NOT LIE ABOVE OR BELOW THE PANEL THE
C               PROJECTION IS MADE ONTO THE PARABOLOID OF WHICH THE
C               PANEL IS A PART.
C******
      COMMON/PANDQ/CP(3,4),PC(3),RO(3),AR(3,3),ART(3,3),P(2,4),A,B,DIAM,
     CC(6,6),AST(6,16),IIS(16),INS,ITS,NPDQ
      DIMENSION Z(3),ZP(3),UN(3),ZN(3)
C        TRANSFORM REPRESENTATION OF POINT FROM GLOBAL TO LOCAL
C        PANEL COORDINATE SYSTEM
      CALL UNIPAN(AR,RO,Z,ZP)
C        CALCULATE VERTICAL COORDINATE OF POINT ON PANEL HAVING SAME
C        HORIZONNTAL LOCATION
      ZP(3)=A*ZP(1)*ZP(1)+B*ZP(2)*ZP(2)
C        CALCULATE SURFACE NORMAL VECTOR AT THIS POINT
      ZN(1)=-2.*A*ZP(1)
      ZN(2)=-2.*B*ZP(2)
      ZN(3)=1.
C        CONVERT NORMAL VECTOR TO UNIT NORMAL VECTOR
      CALL UVECT(ZN)
C        CONVERT PROJECTION TO GLOBAL COORDINATES
      CALL PANUNI(ART,RO,ZP,ZP)
C        CONVERT NORMAL VECTOR TO GLOBAL COORDINATES
      CALL MMULT(ART,ZN,UN,3,3,1)
      RETURN
      END
```

```
      SUBROUTINE TCNTRL
C******
C     SUBROUTINE TCNTRL
C
C     PURPOSE   TO DESIGNATE THE LOCATION OF CONTROL POINTS FOR ALL NET-
C               WORK PANELS AND TO COMPUTE THE UNIT NORMAL VECTOR AND THE
C               NORMAL COMPONENT OF FREE STREAM VELOCITY VECTOR AT EVERY
C               CONTROL POINT
C
C     INPUT     COMMON BLOCK
C               /INDEX/ - NT,NM,NN,NPA,NZA,NNETT
C               /MSPNTS/ - ZM
C
C     OUTPUT    COMMON BLOCK
C               /BDYCS/ - ZC,ZCC,ZCR,ZDC,IPC,ITC
C               /INDEX/ - NCA,NCTPT
C
C     SUBROUTINES
C     CALLED    CONTRL
C
C     DISCUSSION  THE ROUTINE CALLS CONTRL TO CALCULATE THE LOCATION OF
C               CONTROL POINTS FOR ALL PANELS AND TO COMPUTE THE UNIT NOR
C               -MAL VECTOR AND THE NORMAL COMPONENT OF FREE STREAM VELO-
C               CITY VECTOR AT EVERY CONTROL POINT ON ALL PANELS FOR EACH
C               NETWORK. IT ALSO FINDS THE CUMULATIVE NUMBER OF CONTROL
C               POINTS AND THE TOTAL NUMBER OF CONTROL POINTS.
C******
      COMMON/BDYCS/ZC(3,125),ZCC(3,125),ZCR(125),ZDC(125),IPC(125),
     1              ITC(125)
      COMMON/INDEX/NT(9),NM(9),NN(9),NP(9),NS(9),NC(9),NZ(9),
     CNPA(10),NSA(10),NCA(10),NZA(10),NNETT,NPANT,NSNGT,NCTPT,NZMPT
      COMMON /MSPNTS/ZM(3,175),ZL(75)
      NCA(1)=0
      DO 200 K=1,NNETT
      NZMPA1=NZA(K)+1
      NCTRA1=NCA(K)+1
      CALL CONTRL(NT(K),NM(K),NN(K),NC(K),NPA(K),ZM(1,NZMPA1),
     1ZC(1,NCTRA1),ZCC(1,NCTRA1),ZCR(NCTRA1),ZDC(NCTRA1),IPC(NCTRA1),
     2ITC(NCTRA1))
      NCA(K+1)=NCA(K)+NC(K)
 200  CONTINUE
C                               OBTAINS THE TOTAL NUMBER OF CONTROL POINTS
      NCTPT=NCA(NNETT+1)
      RETURN
      END
```

```
      SUBROUTINE TGEOMC
C******
C     SUBROUTINE TGEOMC
C
C     PURPOSE  TO GENERATE ESSENTIAL GEOMETRY INFORMATION FOR EACH PANEL
C              OF ALL THE NETWORKS
C
C     INPUT    COMMON BLOCK
C              /INDEX/ - NT,NM,NN,NPA,NZA,NNETT
C              /MSPNTS/ - ZM
C
C     OUTPUT   SEE OUTPUT OF SUBROUTINE GEOMC
C
C     SUBROUTINES
C     CALLED   GEOMC
C
C     DISCUSSION  THE ROUTINE CALLS GEOMC TO CALCULATE ESSENTIAL GEOME-
C              TRY FOR ALL PANELS OF EACH NETWORK.
C******
      COMMON/INDEX/NT(9),NM(9),NN(9),NP(9),NS(9),NC(9),NZ(9),
     CNPA(10),NSA(10),NCA(10),NZA(10),NNETT,NPANT,NSNGT,NCTPT,NZMPT
      COMMON /MSPNTS/ZM(3,175),ZL(75)
      DO 200 K=1,NNETT
      NZMPA1=NZA(K)+1
      NPANA1=NPA(K)+1
      CALL GEOMC(NT(K),NM(K),NN(K),NPA(K),ZM(1,NZMPA1))
  200 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TSING
C******
C     SUBROUTINE TSING
C
C     PURPOSE  TO DESIGNATE THE LOCATION OF DOUBLETS ON ALL NETWORK
C              PANELS AND TO COMPUTE THE MATRIX FOR COEFFICIENTS OF QUAD
C              -RATIC DOUBLETS DISTRIBUTION FOR EACH PANEL
C
C     INPUT    COMMON BLOCK
C              /INDEX/ - NT,NM,NN,NPA,NZA,NNETT
C              /MSPNTS/ - ZM
C
C     OUTPUT   COMMON BLOCK
C              /INDEX/ - NS,NSA,NSNGT
C
C     SUBROUTINES
C     CALLED   SING
C
C     DISCUSSION  THE ROUTINE CALLS SING TO CALCULATE THE LOCATION OF
C              DOUBLETS ON PANELS AND TO COMPUTE THE MATRIX FOR COEFFI-
C              CIENTS OF QUADRATIC DOUBLET DISTRIBUTION FOR EVERY PANEL
C              OF EACH NETWORK. IT ALSO FINDS CUMULATIVE NUMBER OF
C              DOUBLETS. FINALLY THE TOTAL NUMBER OF DOUBLETS IS OB-
C              TAINED.
C******
      COMMON/INDEX/NT(9),NM(9),NN(9),NP(9),NS(9),NC(9),NZ(9),
     CNPA(10),NSA(10),NCA(10),NZA(10),NNETT,NPANT,NSNGT,NCTRT,NZMPT
      COMMON /MSPNTS/ZM(3,175),ZL(75)
      NSA(1)=0
      DO 200 K=1,NNETT
      NZMPA1=NZA(K)+1
      CALL SING(NT(K),NM(K),NN(K),NS(K),NSA(K),NPA(K),ZM(1,NZMPA1))
      NSA(K+1)=NSA(K)+NS(K)
  200 CONTINUE
C                               OBTAINS THE TOTAL NUMBER OF DOUBLETS
      NSNGT=NSA(NNETT+1)
      RETURN
      END
```

```
      SUBROUTINE VINFCC(Z,ZN,ZD,JPC)
C******
C     SUBROUTINE VINFCC
C
C     PURPOSE  TO GENERATE THE THREE COMPONENTS OF AERODYNAMIC INFLUENCE
C              COEFFICIENTS FOR A GIVEN CONTROL POINT INDUCED BY ALL PAN
C              -ELS DOUBLET DISTRIBUTION
C
C     INPUT    CALLING SEQUENCE
C              Z - X,Y,Z COORDINATES OF A GIVEN CONTROL POINT
C              ZN - NORMAL VECTOR AT THE CONTROL POINT ON PANEL SURFACE
C              ZD - PERTURBATION DISTANCE FOR CONTROL POINT AT EDGES
C              JPC - INDEX OF PANEL OF WHICH COMPONENTS OF AIC ARE TO BE
C                    TRANSFORMED TO ITS LOCAL COORDINATES
C              COMMON BLOCK
C              /CMO3/ - NPIF
C              /INDEX/ - NPANT,NSNGT
C
C     OUTPUT   COMMON BLOCK
C              /PINC/ - DVDFS
C
C     SUBROUTINES
C     CALLED   PTRNS,EIVC,PIVC,MMULT
C
C     DISCUSSION  FOR EVERY PANEL, THE ROUTINE CALLS PTRNS TO TRANSFER
C              PANEL INFORMATION. DEPENDING ON THE GIVEN CONROL POINT IS
C              AT EDGE OR INTERIOR OF THE PANEL, IT CALLS EIVC OR PIVC
C              TO EVALUATE THE INTEGRALS. THE LATTER IS THEN MULTIPLIED
C              BY THE GENERALIZED INVERSE FROM LEAST SQUARES FIT OF QUAD
C              -DRATIC DOUBLET DISTRIBUTION OBTAINED IN SUBROUTINE SING
C              TO FORM THE THREE COMPONENTS OF AERODYNAMIC INFLUENCE
C              COEFFICIENTS. IF JPC IS SPECIFIED, THE COMPONENTS OF AIC
C              WILL BE TRANSFORMED TO LOCAL COORDINATES OF THAT PARTI-
C              CULAR PANEL.
C******
      COMMON  /CMO3/NTSIN,NTSOUT,NTGD,NPIF,NAIC3,NAIC,NJAC,NSCR
      COMMON/INDEX/NT(9),NM(9),NN(9),NP(9),NS(9),NC(9),NZ(9),
     CNPA(10),NSA(10),NCA(10),NZA(10),NNETT,NPANT,NSNGT,NCTRT,NZMPT
      COMMON/PANDQ/CP(3,4),PC(3),PO(3),AP(3,3),APT(3,3),P(2,4),A,B,DIAM,
     CC(6,6),AST(6,16),IIS(16),INS,ITS,NPQQ
      COMMON /PINC/DVDFS(3,125)
      COMMON /PINDX/KP,KQ,NPWR,NPRD
      COMMON/PIVM/DVDS(3,6)
      COMMON /ZIP/IPZ,IP,ITZ,JCZ
      DIMENSION VF(3),VS(3,16),Z(3),ZN(3)
C                             SETS ARRAY DVDFS TO ZERO
      CALL ZERO(DVDFS,3*NSNGT)
C                             OBTAINS THE 3 COMPONENTS OF AERODYNAMIC
C                             INFLUENCE COEFFICIENTS FOR A GIVEN CONTROL
C                             POINT INDUCED BY ALL DOUBLET PANELS OF HALF
C                             THE CONFIGURATION AND THEIR IMAGES
      KQ = 0
```

```
      REWIND NPIF  $  NPFD = NPIF
      DO 700 IP=1,NPANT
      CALL PTFNS(IP)
      IF(ZD.EQ.0)   GO TO 600
      CALL EIVC(Z,ZN,ZD,IPINF)
      IF(IPINF)  625,700,625
  600 CALL PIVC(Z)
  625 CALL MMULT(DVDS,AST,VS,3,6,INS)
      DO 650 IC=1,INS
      IS=IIS(IC)
      DVDFS(1,IS)=DVDFS(1,IS)+VS(1,IC)
      DVDFS(2,IS)=DVDFS(2,IS)+VS(2,IC)
      DVDFS(3,IS)=DVDFS(3,IS)+VS(3,IC)
  650 CONTINUE
  700 CONTINUE
      IF(JPC.EQ.0) GO TO 900
C                                   TRANSFORMS AIC TO LOCAL PANEL (JPC) COORD.
      CALL PTRNS(JPC)
      DO 850 IS=1,NSNGT
      CALL MMULT(AR,DVDFS(1,IS),VR,3,3,1)
      DO 800 I=1,3
  800 DVDFS(I,IS)=VR(I)
  850 CONTINUE
  900 RETURN
      END
```

```
      OVERLAY(VORTEX,3,0)
      PROGRAM SOLVER
C******
C     PROGRAM  SOLVER
C
C     PURPOSE  TO SOLVE A LINEAR SYSTEM OF EQUATIONS A*X=B
C
C     INPUT    COMMON BLOCK
C              /NEQS/ - NE,NR,NMAT,NRHS
C
C
C     OUTPUT   COMMON BLOCK
C              /NEQS/ - NRHS
C
C     SUBROUTINES
C     CALLED   LINEQS
C
C     DISCUSSION  SEE PROGRAM DOCUMENT 1.3 DESCRIPTION AND FLOW CHART OF
C              OVERLAY PROGRAMS.
C                 THE PROGRAM HAS BEEN SET UP WITH THE CONSIDERATION
C              THAT AN OUT-OF-CORE EQUATION SOLVER CAN BE REPLACE THE PR
C              SENT IN-CORE ONE WITHOUT CHANGING THE DATA STRUCTURE
C              SIGNIFICANTLY.
C******
      COMMON /NEQS/NE,NR,NMAT,NRHS
      DIMENSION A(130,130),B(130),IPR(130)
      NM = 130
C                                 READS OUT COEFFICIENT MATRIX AND RIGHT-
C                                 HAND SIDE AND STORES THEM IN ARRAYS A $ B
      REWIND NMAT
      REWIND NRHS
      DO 10 I=1,NE
      READ(NMAT)  (A(I,J),J=1,NE)
   10 READ(NRHS)  B(I)
      CALL LINEQS(A,NM,NE,IPR,B,NR,D1)
      IF(D1.NE.0.)  GO TO 20
      PRINT 15
   15 FORMAT(///* THE MATRIX APPEARS SINGULAR*)
      STOP
C                                 WRITES SOLUTION VECTOR ON THE FILE
   20 REWIND NRHS
      WRITE(NRHS)  B
      RETURN
      END
```

```
      OVERLAY(VORTEX,4,0)
      PROGRAM OUTPUT
C******
C     PROGRAM   OUTPUT
C
C     PURPOSE   TO CALCULATE AND PRINT THE FOLLOWING RESULTS
C               PANEL INDICES OF THE WING, FREE AND FED SHEETS AND WAKE
C               X,Y,Z COORDINATES, PANEL NUMBER AND CIRCULATION AT POINTS
C                  ALONG THE TERMINATED EDGE OF THE FED SHEET
C               X,Y,Z COORDINATES, PANEL NUMBER AND CIRCULATION AT POINTS
C                  ALONG THE WING TRAILING EDGE
C               PANEL NUMBER, X,Y,Z COORDINATES OF PANEL CENTER POINT,
C                  UPPER AND LOWER VELOCITY, DELTA CP, UPPER AND LOWER
C                  CP, AND PANEL AREA
C               NORMAL FORCE COEFFICIENT
C               PITCHING MOMENT COEFFICIENT
C               PITCH AXIS
C               ROOT CHORD
C               WING AREA
C               X,Y,Z COORDINATES OF PANEL CORNER POINTS IN THE FREE
C                  SHEET NETWORK
C               X,Y,Z COORDINATES OF PANEL CORNER POINTS IN THE FED
C                  SHEET NETWORK
C
C     INPUT     COMMON BLOCK
C               /BDYCS/ - ZC
C               /CM03/ - NPIF,NAIC3
C               /FSVEL/ - FSV,XPITCH
C               /INDEX/ - NM,NPA,NZA
C               /MSPNTS/ - ZM
C               /NFAJ/ - NEQ,NF,NG
C               /PANDQ/ - AR,ART,C
C               /PINC/ - DVDFS
C               /SOLN/ - S
C
C     OUTPUT    SEE PURPOSE
C
C     SUBROUTINES
C     CALLED    MMULT,PTRNS,SNGCAL,UVECT,VIP
C
C     DISCUSSION  SEE PROGRAM DOCUMENT 1.3 DESCRIPTION AND FLOW CHART OF
C               OVERLAY PROGRAMS.
C******
      COMMON  /CM03/NTSIN,NTSOUT,NTGD,NPIF,NAIC3,NAIC,NJAC,NSCR
      COMMON/BDYCS/ZC(3,125),ZCC(3,125),ZCR(125),ZDC(125),IPC(125),
     1             ITC(125)
      COMMON/INDEX/NT(9),NM(9),NN(9),NP(9),NS(9),NC(9),NZ(9),
     CNPA(10),NSA(10),NCA(10),NZA(10),NNETT,NPANT,NSNGT,NCTRT,NZMPT
      COMMON/MSPNTS/ZM(3,175),ZL(75)
      COMMON/PANDQ/CP(3,4),PC(3),RC(3),AR(3,3),ART(3,3),P(2,4),A,B,DIAM,
     CC(6,6),AST(6,16),IIS(16),INS,ITS,NPDQ
      COMMON /PINC/DVDFS(3,125)
```

```
      COMMON /PINDX/KP,KQ,NPWF,NPRD
      COMMON/FSVEL/FSV(3),FSVM,ALPHA,XPITCH,RCHORD
      COMMON /NFAJ/NEQ,NF,NG
      COMMON /NITF/NFUN,JT,ITMX,KIT,ITPRIN
      COMMON /SOLN/S(125),ZA(75)
      COMMON /IPRINT/IPNPUT,IPGEOM,IPSING,IPCNTR,IPEIVC,IPOUTP
      DIMENSION VEL(3),VELFS(3),Z(3),TSC(6),VU(3),VL(3)
      IF(IPOUTP.EQ.0.OR.NFUN.NE.100)  GO TO 50
      PRINT 2010, NNETT,NPANT,NSNGT,NCTRT,NZMPT
 2010 FORMAT(*1FROM OUTPUT*/5I5)
      PRINT 2020, (S(I),I=1,NSNGT)
 2020 FORMAT(//* SOLUTION S*/(5E14.6))
   50 CONTINUE
C
C                                 PRINTS PANEL NO. FOR DIFFERENT NETWORKS
      I1 = 1             $   I2 = NPA(2)
      I3 = NPA(2)+1      $   I4 = NPA(3)
      I5 = NPA(3)+1      $   I6 = NPA(4)
      I7 = NPA(4)+1      $   I8 = NPA(6)
      WRITE(NTSOUT,5010)  I1,I2,I3,I4,I5,I6,I7,I8
 5010 FORMAT( ///48X,*WING PANEL NUMBER*,8X,I4,*  TO*,I4/
     1             48X,*FREE SHEET PANEL NUMBER*,2X,I4,*  TO*,I4/
     2             48X,*FED SHEET PANEL NUMBER*,3X,I4,*  TO*,I4/
     3             48X,*WAKE PANEL NUMBER*,8X,I4,*  TO*,I4)
C
C                                 PRINTS CIRCULATION ALONG TERMINATED EDGE
C                                 OF FED SHEET
      WRITE(NTSOUT,5020)
 5020 FORMAT(///43X,*CIRCULATION ALONG TERMINATED EDGE OF FED SHEET*//
     142X,*X*,10X,*Y*,10X,*Z*,10X,*PANEL*,6X,*CIRCULATION*/)
      KQ = 0
      REWIND NPIF  $  NPRD = NPIF
      M3 = NM(3)
      L1 = NZA(3) + M3
      DO 200 IP=I5,I6
      L2 = L1 + M3
      CALL PTENS(IP)
      DO 100 L=1,3
      Z(L) = 0.5*(ZM(L,L1) + ZM(L,L2))
  100 CONTINUE
      CALL SNGCAL(Z,TSC)
      WRITE(NTSOUT,5030)  (Z(L),L=1,3),IP,TSC(1)
 5030 FORMAT(34X,3F11.4,8X,I4,4X,F11.4)
      L1 = L2
  200 CONTINUE
C
C                                 PRINTS CIRCULATION ALONG WING TRAILING EDGE
      WRITE(NTSOUT,5040)
 5040 FORMAT(//48X,*CIRCULATION ALONG WING TRAILING EDGE*//
     142X,*X*,10X,*Y*,10X,*Z*,10X,*PANEL*,6X,*CIRCULATION*/)
      INP = NPA(5) - (NM(2)-1)
      L1 = NZA(4) + 1
```

```
      DO 400 IP=I7,INP
      L2 = L1 + 1
      CALL PTRNS(IP)
      DO 300 L=1,3
      Z(L) = 0.5*(ZM(L,L1) + ZM(L,L2))
  300 CONTINUE
      CALL SNGCAL(Z,TSC)
      WRITE(NTSOUT,5030)  (Z(L),L=1,3),IP,TSC(1)
      L1 = L2
  400 CONTINUE
C
C                                PRINTS V(UPPER), V(LOWER), CP(UPPER),
C                                CP(LOWER), DELTA CP, ETC. EVALUATED AT THE
C                                CONTROL POINTS CORRESPONDING TO CENTRAL
C                                LOCATION OF PANELS
      WRITE(NTSOUT,5050)
 5050 FORMAT(///4X,*PANEL*,5X,*ZCX*,6X,*ZCY*,6X,*ZCZ*,6X,*VUX*,6X,*VUY*,
     16X,*VUZ*,6X,*VLX*,6X,*VLY*,6X,*VLZ*,6X,*DCP*,6X,*CPU*,6X,*CPL*,6X,
     2*AREA*/)
      REWIND NAIC3
      KO = 0
      REWIND NPIF  $  NPFD = NPIF
      NEW = NE - NG
      DO 500 I=1,NEQ
      READ(NAIC3)  DVDFS(1)
  500 CONTINUE
      SW = 0.
      CN = 0.
      CM = 0.
      DO 900 IJ=1,NE
      READ(NAIC3)  DVDFS
      IP = IJ  $  JC = NEQ + IJ
      CALL PTRNS(IP)
C                          CALCULATES  V(TOTAL) AND GRAD(MU)
      CALL MMULT(DVDFS,S,VEL,3,NSNGT,1)
      DO 600 I=1,3
  600 VEL(I) = VEL(I) + FSV(I)
      CALL SNGCAL(ZC(1,JC),TSC)
      Z(1)=TSC(2)  $  Z(2) = TSC(3)  $  Z(3) = 0.
      CALL MMULT(ART,Z,VELFS,3,3,1)
C                                CALCULATES V(UPPER) AND V(LOWER)
      DO 700 I=1,3
      VU(I) = VEL(I) + 0.5*VELFS(I)
      VL(I) = VEL(I) - 0.5*VELFS(I)
  700 CONTINUE
C                          CALCULATES  DELTA CP
      CALL VIP(VEL,1,VELFS,1,3,HDCP)
      DCP = 2.*HDCP
      IF(IJ.GT.NEW)  GO TO 800
C                                CALCULATES CP(UPPER) AND CP(LOWER) FOR WING
      CALL VIP(VEL,1,VEL,1,3,VSQ)
      CALL VIP(VELFS,1,VELFS,1,3,GMUSQ)
```

```
      CPU = 1. - (VSQ + HDCP + 0.25*GMUSQ)
      CPL = 1. - (VSQ - HDCP + 0.25*GMUSQ)
C                               CALCULATES WING AREA, NORMAL FORCE COEFF.,
C                               PITCHING MOMENT COEFF.
      SP = C(1,1)
      SW = SW + SP
      CNF = AF(3,3)*DCP*SP
      CN = CN + CNF
      CM = CM + CNF*(ZC(1,JC) - XPITCH)
      WRITE(NTSOUT,5060)  IP,(ZC(I,JC),I=1,3),VU,VL,DCP,CPU,CPL.SP
 5060 FORMAT(3X,I4,2X,13F9.4)
      GO TO 900
  800 WRITE(NTSOUT,5070)  IP,(ZC(I,JC),I=1,3),VU,VL,DCP
 5070 FORMAT(3X,I4,2X,10F9.4)
  900 CONTINUE
C                               CALCULATES NORMAL FORCE COEFF.,
C                               PITCHING MOMENT COEFF.
      SW = 2.*SW
      CN = 2.*CN/SW
      CM = 2.*CM/(RCHORD*SW)
      WRITE(NTSOUT,5080)  CN,CM,XPITCH,RCHORD,SW
 5080 FORMAT(///47X,*NORMAL FORCE COEFFICIENT =*,3X,F9.4/
     1             47X,*PITCHING MOMENT COEFFICIENT =*,F9.4/
     2             47X,*PITCH AXIS =*,17X,F9.4/
     3             47X,*ROOT CHORD =*,17X,F9.4/
     4             47X,*WING AREA =*,18X,F9.4)
C
C                               PRINTS CORNER POINTS OF FREE SHEET NETWORK
C                               AND FED SHEET NETWORK
      WRITE(NTSOUT,5090)
 5090 FORMAT(///49X,*X Y Z COORDINATES OF CORNER POINTS*)
      J1 = NZA(2)+1  $  J2 = NZA(3)
      WRITE(NTSOUT,5100)  (ZM(1,J),ZM(2,J),ZM(3,J),J=J1,J2)
 5100 FORMAT(/57X,*FREE SHEET NETWORK*//(15F8.3))
      J1 = NZA(3)+1  $  J2 = NZA(4)
      WRITE(NTSOUT,5110)  (ZM(1,J),ZM(2,J),ZM(3,J),J=J1,J2)
 5110 FORMAT(/57X,*FED SHEET NETWORK*//(15F8.3))
      RETURN
      END
```

```
      SUBROUTINE SINFCC(Z)
C******
C     SUBROUTINE SINFCC (Z)
C
C     PURPOSE  GIVEN THE X,Y,Z COORDINATES OF A POINT SINFCC DEFINES A
C              MATRIX (DSDFS), WHICH WHEN MULTIPLIED BY A VECTOR CON-
C              SISTING OF VALUES OF ALL DOUBLET PARAMETERS, GIVES THE
C              VALUE AND 1ST,2ND DERIVATIVES OF DOUBLET STRENGTH AT THE
C              GIVEN POINT
C
C     INPUT    CALLING SEQUENCE
C              Z - X,Y,Z COORDINATES OF THE GIVEN POINT
C              COMMON BLOCK
C              /INDEX/ - NSNGT
C              /PANDQ/ - RO,AR,AST,IIS,INS
C
C     OUTPUT   COMMON BLOCK
C              /SNGC/ - DSDFS
C
C     SUBROUTINES
C     CALLED   UNIPAN
C
C     DISCUSSION  SUBROUTINE UNIPAN CONVERTS THE INPUT POINT FROM THE
C              UNIVERSAL TO LOCAL PANEL COORDINATE SYSTEM.
C                 A SIX BY SIX MATRIX IS FORMED BY THE GENERAL EQUATION
C              REPRESENTING THE DOUBLET STRENGTH DISTRIBUTION AT THE GIV
C              -EN POINT ON A PANEL AND ITS DERIVATIVES.
C                 A SIX BY SIXTEEN MATRIX (AST) FOR COEFFICIENTS OF QUAD
C              -RATIC DOUBLET DISTRIBUTION ON THE PANEL ALSO EXISTS. THE
C              MATRIX IS COMPUTED IN SUBROUTINE SING.
C                 THE MATRIX DSDFS IS FORMED BY MULTIPLYING THESE TWO
C              MATRICES.
C******
      COMMON/INDEX/NT(9),NM(9),NN(9),NP(9),NS(9),NC(9),NZ(9),
     CNPA(10),NSA(10),NCA(10),NZA(10),NNETT,NPANT,NSNGT,NCTRT,NZMPT
      COMMON/PANDQ/CP(3,4),PC(3),RO(3),AR(3,3),ART(3,3),P(2,4),A,B,DIAM,
     CC(6,6),AST(6,16),IIS(16),INS,ITS,NPDQ
      COMMON /SNGC/ DSDFS(6,125)
      DIMENSION Z(3),W(3)
      EQUIVALENCE (X,W(1)), (Y,W(2))
C                               TRANSFORMS THE INPUT POINT FROM GLOBAL TO
C                               LOCAL PANEL COORDINATE SYSTEM
      CALL UNIPAN(AR,RO,Z,W)
C                               SETS ARRAY DSDFS TO ZERO
      CALL ZERO(DSDFS,6*NSNGT)
C                               MULTIPLIES TWO MATRICES TO FORM THE MATRIX
C                               DSDFS
      DO 200 IC=1,INS
      IS=IIS(IC)
      DX=AST(4,IC)*X+AST(5,IC)*Y
      DY=AST(5,IC)*X+AST(6,IC)*Y
      DSDFS(1,IS)=DSDFS(1,IS)+AST(1,IC)+(AST(2,IC)+.5*DX)*X
```

```
     C+(AST(3,IC)+.5*DY)*Y
      DSDFS(2,IS)=DSDFS(2,IS)+AST(2,IC)+DX
      DSDFS(3,IS)=DSDFS(3,IS)+AST(3,IC)+DY
      DSDFS(4,IS)=DSDFS(4,IS)+AST(4,IC)
      DSDFS(5,IS)=DSDFS(5,IS)+AST(5,IC)
      DSDFS(6,IS)=DSDFS(6,IS)+AST(6,IC)
200   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SNGCAL(Z,TSC)
C******
C     SUBROUTINE SNGCAL (Z,TSC)
C
C     PURPOSE  TO CALCULATE THE VALUE AND 1ST,2ND DERIVATIVES OF DOUB-
C              LET STRENGTH AT THE SPECIFIED POINT
C
C     INPUT    CALLING SEQUENCE
C              Z - X,Y,Z COORDINATES OF THE GIVEN POINT
C              COMMON BLOCK
C              /SOLN/ - S
C
C     OUTPUT   CALLING SEQUENCE
C              TSC - ARRAY CONSISTS OF THE VALUE AND 1ST,2ND DERIVATIVES
C                    OF DOUBLET STRENGTH
C
C     SUBROUTINES
C     CALLED   SINFCC,MMULT
C
C     DISCUSSION  SNGCAL CALLS SUBROUTINE SINFCC TO PRODUCE THE MATRIX
C              DSDFS. MMULT MULTIPLIES THIS MATRIX BY THE VECTOR CONSIST
C              -INT OF VALUES OF ALL DOUBLET PARAMETERS PREVIOUSLY OB-
C              TAINED TO PRODUCE THE VALUE AND 1ST,2ND DERIVATIVES OF
C              DOUBLET STRENGTH AT THE GIVEN POINT
C******
      COMMON/INDEX/NT(9),NM(9),NN(9),NP(9),NS(9),NC(9),NZ(9),
     CNPA(10),NSA(10),NCA(10),NZA(10),NNETT,NPANT,NSNGT,NCTRT,NZMPT
      COMMON /SNGC/DSDFS(6,125)
      COMMON /SOLN/S(125),ZA(75)
      DIMENSION Z(3),TSC(6)
      CALL SINFCC(Z)
      CALL MMULT(DSDFS,S,TSC,6,NSNGT,1)
      RETURN
      END
```

```
      SUBROUTINE BSUBSM(A,NR,N,IPR,B,M)
C******
C     SUBROUTINE BSUBSM (A,NR,N,IPR,B,M)
C
C     PURPOSE   TO PERFORM BACK SUBSTITUTIONS USING THE FACTORIZATION OB-
C               TAINED FROM A DECOMPOSITION ROUTINE AND FIND THE SOLUTION
C               FOR A SYSTEM OF EQUATIONS
C
C     INPUT     CALLING SEQUENCE
C               A - THE LOWER TRIANGLE OF THE ARRAY CONSISTS OF A LOWER
C                   TRIANGULAR MATRIX L AND THE UPPER TRIANGLE CONSISTS
C                   OF AN UPPER TRIANGULAR MATRIX U. THEY ARE OBTAINED
C                   FROM A DECOMPOSITION ROUTINE SUCH AS TDECOM
C               NR - MAXIMUM ROW DIMENSION OF ARRAYS A AND B
C               N - ORDER OF THE COEFFICIENT MATRIX
C               IPR - ARRAY CONSISTS OF NUMBERS OF PIVOTAL ROW, AS DERIV-
C                     ED FROM THE SUBROUTINE TDECOM
C               B - ARRAY CONSISTS OF M RIGHT-HAND SIDES OF THE LINEAR
C                   SYSTEM
C               M - NUMBER OF RIGHT-HAND SIDES
C
C     OUTPUT    CALLING SEQUENCE
C               B - SOLUTION VECTORS
C
C     SUBROUTINES
C     CALLED    VIPS
C
C     DISCUSSION  THE ROUTINE FIRST USES PIVOTAL INFORMATION GIVEN IN
C               THE ARRAY IPR TO EXCHANGE ELEMENTS OF RIGHT-HAND SIDES.
C               IT THEN PERFORMS FORWARD SUBSTITUTION BY SOLVING THE LOW-
C               ER TRIANGULAR SYSTEM OF EQUATIONS LY=B AND BACKWARD SUB-
C               STITUTION BY SOLVING THE UPPER TRIANGULAR SYSTEM OF EQUA-
C               TIONS UX=Y. X IS THE DESIRED SOLUTION OF THE GIVEN SYSTEM
C               OF EQUATIONS.
C                  THE ROUTINE IS A MODIFIED VERSION OF A ROUTINE IN THE
C               SUBROUTINE LIBRARY OF THE BOEING COMPUTER SERVICES CO.
C******
      DIMENSION A(NR,1),IPR(1),B(NR,1)
C                             USES PIVOTAL INFORMATION TO EXCHANGE
C                             ELEMENTS OF RIGHT-HAND SIDES
      DO 10 I=1,N
      IF(IPR(I).EQ.I)  GO TO 10
      DO 5 K=1,M
      X=B(I,K)
      J=IPR(I)
      B(I,K)=B(J,K)
    5 B(J,K)=X
   10 CONTINUE
C                             PERFORMS FORWARD SUBSTITUTION
      NM1 = N - 1
      DO 50 K=1,M
      B(1,K) = B(1,K)/A(1,1)
```

```
      IF(N.EQ.1) GO TO 30
      DO 20 I=2,N
      X=B(I,K)
      CALL VIPS(A(I,1),NR,B(1,K),1,I-1,X)
   20 B(I,K) = X/A(I,I)
C                                  PERFORMS BACKWARD SUBSTITUTION
   30 B(N,K) = B(N,K)
      IF(N.EQ.1)  GO TO 50
      DO 40 IN=1,NM1
      I = N-IN
      X = B(I,K)
      I1 = I+1
      CALL VIPS(A(I,I1),NR,B(I1,K),1,IN,X)
      B(I,K) = X
   40 CONTINUE
   50 CONTINUE
      RETURN
      END
```

```
          IDENT   CMAB          (A,B,R,NRA,NCA,NCB)
*******
*      SUBROUTINE CMAB (A,B,R,NRA,NCA,NCB)
*
*      PURPOSE  TO MULTIPLY TWO MATRICES WHOSE ELEMENTS ARE STORED
*               COMPACTLY BY ROWS (COMPASS)
*
*      INPUT    CALLING SEQUENCE
*               A - LOCATION OF FIRST MATRIX
*               B - LOCATION OF SECOND MATRIX
*               R - LOCATION OF RESULTANT MATRIX
*               NRA - NUMBER OF ROWS IN FIRST MATRIX
*               NCA - NUMBER OF COLUMNS IN FIRST MATRIX
*               NCB - NUMBER OF COLUMNS IN SECOND MATRIX
*
*      OUTPUT   CALLING SEQUENCE
*               R - RESULTANT MATRIX
*
*      SUBROUTINES
*      CALLED   NONE
*
*      DISCUSSION  PERFORMS THE MATRIX OPERATION   (R) = (B) (A)
*
*          WARNING -
*             THIS ROUTINE USES RUN CONVENTION CALLING SEQUENCE
*             DO NOT CALL FROM FTN COMPILED PROGRAMS
*******
          ENTRY CMAB
 CMAB     BSSZ   1
*
*         INITIALIZATION PORTION
*
          SA4     B4                 LOAD NUMBER OF ROWS A MATRIX
          SA5     B5                 LOAD NUMBER OF ROWS B MATRIX
          DX7     X4*X5               GET NUMBER ELEMENTS IN MATRIX
          SB4     B1+X5              LAST ADDRESS PLUS ONE FIRST ROW A
          SA0     B1                 SAVE ADDRESS OF A IN A0
          SB5     X5                 NUMBER COLUMNS A
          SA3     B6                 NUMBER COLUMNS B
          SX0     1                  ONE TO X0
          SB7     B1+X7              LWA+ONE OF A MATRIX TO B7
          SB6     X3                 NUMBER COLUMNS B MATRIX
          BX1     X0*X3              EVEN/ODD FLAG FOR NCB TO X1
          SA4     B3+X3              STORE FWA SECOND ROW OF B IN A4
          ZR      X1,CLOOP           IF B HAS AN EVEN NUMBER OF COLUMNS
*                                    NEED NOT DO THE ODD LOOP
*
*         PROCESS FIRST COLUMN OF R IF NCB IS ODD
*
          SX0     B3                 SAVE ADDRESS OF R IN X0
*
*         THE FOLLOWING CODE IS EXECUTED IFF NCB IS ODD
```

```
*
*        LOOPBACK IS FOR EACH ROW IN A
*
 FL      MX6    0                    ZERO TO X6 AS ACCUMULATOR
         SA1    B1                   FIRST ELEMENT OF ROW  OF A TO X1
         SA2    B2                   FIRST ELEMENT OF COLUMN OF B TO X1
*
*        LOOPBACK IS FOR THE INNER PRODUCT (NCA TIMES)
*
 IL      FX5    X1*X2                MULTIPLY ELEMENT A * ELEMENT B
         SB1    B1+1                 BUMP ADDRESS IN A UP AS LOOP COUNTER
         SA2    A2+B6                LOAD NEXT ELEMENT COLUMN OF B
         SA1    B1                   LOAD NEXT ELEMENT ROW OF A
         FX6    X5+X6                ADD ON CURRENT CONT. TO INNER PRODUCT
         LT     B1,B4,IL             DONE IF B1 IS POINTS TO NEXT ROW OF A
         NX6     X6
         SA6    B3                   STORE ELEMENT IN THE R MATRIX
         SB4    B4+B5                BUMP INNER LOOP DONE COUNTER BY NCA
         SB3    B3+B6                BUMP R STORE BY NCB
         LE     B4,B7,FL             TEST FOR ALL ROWS FIRST COL OF R DONE
         SB1    A0                   RESTORE ADDRESS OF A
         SB3    X0+1                 RESTORE R ADD. TO SECOND ELEMENT OF R
         SB2    B2+1                 RESTORE B ADD. TO SECOND ELEMENT OF B
         SB4    B1+B5                RESTORE B4 TO LWA+1 FIRST ROW OF A
         SX1    B6-1                 TEST FOR DONE AT THIS POINT
         ZR     X1,CMAB              DONE IF ONLY ONE COLUMN IN B
*
*        PRIMARY PORTION OF CODE TO PROCESS MULTIPLY
*
*
*        LOOPBACK IS FOR A PAIR OF COLUMNS IN B
*
 CLOOP   SA1    B1                   LOAD FIRST ELEMENT OF A FOR INNER LOOP
         SX0    B3                   SAVE ADDRESS OF R MATRIX COL IN X0
*
*        LOOPBACK IS FOR ROWS OF A
*
 RLOOP   MX6    0                    ZERO TO X6 FOR ODD ACCUMULATOR
         SA2    B2                   LOAD FIRST  ELEMENT ODD COLUMN OF B
         SA3    B2+1                 LOAD FIRST  ELEMENT EVEN COLUMN OF B
         MX7    0                    ZERO TO X7 AS EVEN ACCUMULATOR
*
*        LOOPBACK IS FOR INNER PRODUCT (NCA TIMES)
*
 ILOOP   FX4    X1*X2                START ODD MULTIPLY GOING
         SB1    B1+1                 BUMP B1 AS LOOP COUNTER
         FX5    X1*X3                START EVEN MULTIPLY GOING
         SA2    A2+B6                LOAD NEXT ELEMENT ODD COLUMN OF B
         SA1    B1                   LOAD NEXT ELEMENT ROW OF A
         FX6    X6+X4                ADD ON INNER PROD. ODD COL
         SA3    A2+1                 LOAD NEXT ELEMENT EVEN COLUMN OF B
         FX7    X7+X5                ADD ON IINNER PROD. EVEN COL
```

```
         LT      B1,B4,ILOOP        DONE IF B1 POINTS TO NEXT ROW OF A
         NX6      X6
         SB4     B4+B5              ADVANCE B4 TO NEXT ROW OF A
         SA6     B3                 STORE ELEMENT R IN ODD COL
         NX7     X7
         SB3     B3+B6              BUMP THE R STORE BY NCB
         SA7     A6+1               STORE ELEMENT R IN EVEN COL
         LT      B1,B7,RLOOP        DONE IF B1 IS PAST THE A MATRIX
         SB3     X0+2               ADVANCE  INITIAL VALUE OF B3 BY TWO
         SB1     A0                 RESTORE B1 TO FIRST ELEMENT OF A
         SB2     B2+2               ADVANCE COL B POINTER BY TWO
         SB4     B1+B5              RESTORE B4 TO SECOND ROW OF A
         SX2     A4-B3              LWA+1 OF R - NEXT COL OF R ADDRESS
         NZ      X2,CLOOP           DONE IF NEXT COL OF R IS SECOND ROW
         EQ      CMAB               GET OUT
         END
```

```
      SUBROUTINE CROSS(A,B,C)
C******
C     SUBROUTINE CROSS (A,B,C)
C
C     PURPOSE  TO CALCULATE THE CROSS PRODUCT OF TWO VECTORS
C
C     INPUT    CALLING SEQUENCE
C              A - FIRST VECTOR
C              B - SECOND VECTOR
C
C     OUTPUT   CALLING SEQUENCE
C              C - RESULTANT VECTOR
C
C     SUBROUTINES
C     CALLED   NONE
C
C     DISCUSSION  CROSS PERFORMS THE FOLLOWING CALCULATIONS-
C                       C(1) = (A(2)*B(3)) - (A(3)*B(2))
C                       C(2) = (A(3)*B(1)) - (A(1)*B(3))
C                       C(3) = (A(1)*B(2)) - (A(2)*B(1))
C******
      DIMENSION A(3),B(3),C(3)
      C(1)=A(2)*B(3)-A(3)*B(2)
      C(2)=A(3)*B(1)-A(1)*B(3)
      C(3)=A(1)*B(2)-A(2)*B(1)
      RETURN
      END
```

```
      SUBROUTINE IPTRNS (IP)
C******
C     SUBROUTINE IPTRNS (IP)
C
C     PURPOSE  TO WRITE PANEL INFORMATION ON DISK
C
C     INPUT     CALLING SEQUENCE
C               IP - PANEL NUMBER OF INFORMATION TO BE WRITTEN
C               COMMON BLOCK
C               /PANDQ/ - CP,PC,RO,AP,ART,P,A,B,DIAM,C,AST,IIS,INS,ITS
C               /PINDX/ - KP,NPWR
C
C     OUTPUT    COMMON BLOCK
C               /PINDX/ - KP
C
C     SUBROUTINES
C     CALLED    NONE
C
C     DISCUSSION  WRITES 197 WORDS OF PANEL INFORMATION FROM COMMON
C               BLOCK PANDQ ONTO DISK FILE SPECIFIED BY NPWR
C******
      COMMON /PANDQ/ PDQ(197), NPDQ
      COMMON /PINDX/ KP,KDUM,NPAN,NDUM
C
      ID = IP - KP
      IF (ID) 200,300,100
  100 IBRANCH = ID
      GO TO 250
  200 IBRANCH = IP
      REWIND NPAN
  250 IF (IBRANCH .EQ. 1) GO TO 290
      MAX = IBRANCH - 1
      DO 275 I=1,MAX
  275 WRITE(NPAN)   PDQ(1)
  290 WRITE(NPAN)   PDQ
  300 KP = IP
      RETURN
      END
```

```
      SUBROUTINE LINEQS(A,NR,N,IPR,B,M,D1)
C******
C     SUBROUTINE LINEQS (A,NR,N,IPR,B,M,D1)
C
C     PURPOSE  TO SOLVE A SYSTEM OF LINEAR EQUATIONS A*X = B
C
C     INPUT    CALLING SEQUENCE
C              A - ARRAY CONSISTS OF ELEMENTS OF THE COEFFICIENT MATRIX
C              NR - MAXIMUM ROW DIMENSION OF ARRAYS A AND B
C              N - ORDER OF THE COEFFICIENT MATRIX
C              B - ARRAY CONSISTS OF M RIGHT-HAND SIDES OF THE LINEAR
C                  SYSTEM
C              M - NUMBER OF RIGHT-HAND SIDES
C
C     OUTPUT   CALLING SEQUENCE
C              A - THE LOWER TRIANGLE OF THE ARRAY CONSISTS OF A LOWER
C                  TRIANGULAR MATRIX L AND THE UPPER TRIANGLE CONSISTS
C                  OF AN UPPER TRIANGULAR MATRIX U (SINCE U IS UNIT UP-
C                  PER TRIANGULAR, ITS DIAGONAL ELEMENTS ARE NOT STORED)
C              IPR - ARRAY GIVES NUMBERS OF PIVOTAL ROW (A RECORD OF IN-
C                    TERCHANGES)
C              B - SOLUTION VECTORS
C              D1 - = +1 OR -1 ACCORDING AS THE NUMBER OF INTERCHANGES
C                   IS EVEN OR ODD. IT ALSO INDICATES SUCCESSFUL RETURN
C                   = 0 INDICATES THAT THE COEFFICIENT MATRIX APPEARS
C                   SINGULAR
C
C     SUBROUTINES
C     CALLED   TDECOM,BSUBSM
C
C     DISCUSSION  ROUTINE TDECOM IS FIRST CALLED BY LINEQS TO PERFORM
C              THE DECOMPOSITION OF THE COEFFICIENT MATRIX A INTO A LOW-
C              ER TRIANGULAR MATRIX L AND AN UPPER TRIANGULAR MATRIX U.
C              THE RESULT IS THEN USED IN BSUBSM FOR CARRYING OUT BACK
C              SUBSTITUTIONS AND OBTAINING THE SOLUTION TO THE SYSTEM OF
C              EQUATIONS.
C                 THIS ROUTINE IS A MODIFIED VERSION OF A ROUTINE IN THE
C              SUBROUTINE LIBRARY OF THE BOEING COMPUTER SERVICES CO.
C******
      DIMENSION A(NR,1),IPR(1),B(NR,1)
C                             CALLS ROUTINE TO DECOMPOSE THE GIVEN
C                             COEFFICIENT MATRIX
      CALL TDECOM(A,NR,N,IPR,IPR,D1)
      IF(D1.EQ.0.) GO TO 10
C                             CALLS ROUTINE TO PERFORM BACK SUBSTITUTIONS
C                             AND OBTAIN THE SOLUTION FOR THE SYSTEM OF
C                             EQUATIONS
      CALL BSUBSM(A,NR,N,IPR,B,M)
   10 RETURN
      END
```

```
      SUPRCUTINE MMULT(A,B,C,L,M,N)
C******
C     SUBRCUTINE MMULT (A,B,C,L,M,N)
C
C     PURPCSE  TO MULTIPLY TWO MATRICES
C
C     INPUT    CALLING SEQUENCE
C              A - ARRAY CONTAINING ELEMENTS OF MATRIX A
C              B - ARRAY CONTAINING ELEMENTS OF MATRIX B
C              L - NUMBER OF ROWS IN A AND C
C              M - NUMBER OF COLUMNS IN A AND ROWS IN B
C              N - NUMBER OF COLUMNS IN B AND C
C
C     OUTPUT   CALLING SEQUENCE
C              C - RESULTANT MATRIX
C
C     SUBRCUTINES
C     CALLED   CMAB
C
C     DISCUSSICN  MMULT CALLS CMAB TO CALCULATE (C) = (A) (B)
C******
      DIMENSION A(L,M),B(M,N),C(L,N)
      CALL CMAB(B,A,C,N,M,L)
      RETURN
      END
```

```
      SUBROUTINE PANUNI(ART,RO,Y,X)
C******
C     SUBROUTINE PANUNI (ART,RO,Y,X)
C
C     PURPOSE  TO TRANSFORM POINT COORDINATES FROM THE LOCAL PANEL
C              SYSTEM TO THE UNIVERSAL SYSTEM
C
C     INPUT    CALLING SEQUENCE
C              ART - LOCAL TO GLOBAL PANEL SYSTEM TRANSFORMATION MATRIX
C              RO  - X,Y,Z COORDINATES OF PANEL CENTER (UNIVERSAL)
C              Y   - X,Y,Z COORDINATES OF POINT TO BE TRANSFORMED(LOCAL)
C
C     OUTPUT   CALLING SEQUENCE
C              X   - X,Y,Z COORDINATES OF TRANSFORMED POINT (UNIVERSAL)
C
C     SUBROUTINES
C     CALLED   MMULT
C
C     DISCUSSION  THE LOCAL PANEL COORDINATES ARE MULTIPLIED BY THE
C              MATRIX ART IN SUBROUTINE MMULT TO PRODUCE THE GLOBAL
C              PANEL COORDINATES WHICH, WHEN ADDED TO THE UNIVERSAL
C              PANEL CENTER, PRODUCE THE UNIVERSAL COORDINATES.
C******
      DIMENSION ART(3,3),RO(3),X(3),Y(3),W(3)
      CALL MMULT(ART,Y,W,3,3,1)
      DO 10 I=1,3
   10 X(I)=W(I)+RO(I)
      RETURN
      END
```

```
      SUBROUTINE PDSEQS(A,NR,N,DN,B,M,D1)
C******
C     SUBROUTINE PDSEQS (A,NR,N,DN,B,M,D1)
C
C     PURPOSE   TO SOLVE A SYSTEM OF EQUATIONS A*X = B, WHERE A IS A POSI
C               -TIVE DEFINITE SYMMETRIC MATRIX, USING CHOLESKY DECOMPOSI
C               -TION
C
C     INPUT     CALLING SEQUENCE
C               A - ARRAY OF WHICH THE UPPER TRIANGLE IS THE UPPER TRIAN-
C                   GLE OF A GIVEN POSITIVE DEFINITE SYMMETRIC MATRIX
C               NR - MAXIMUM ROW DIMENSION OF ARRAYS A AND B
C               N - ORDER OF THE POSITIVE DEFINITE COEFFICIENT MATRIX
C               B - ARRAY CONSISTS OF M RIGHT-HAND SIDES OF THE LINEAR
C                   SYSTEM
C               M - NUMBER OF RIGHT-HAND SIDES
C
C     OUTPUT    CALLING SEQUENCE
C               B - SOLUTION VECTORS
C               A - ARRAY OF WHICH THE UPPER TRIANGLE IS SAME AS INPUT,
C                   THE LOWER TRIANGLE CONTAINS THE LOWER TRIANGULAR MAT-
C                   RIX L FROM CHOLESKY DECOMPOSITION WITH DIAGONAL ELE-
C                   MENTS EXCLUDED
C               DN - THE RECIPROCALS OF DIAGONAL ELEMENTS OF L
C               D1 - = 1  FOR SUCCESSFUL RETURN
C                    = 0  INDICATES THAT THE GIVEN COEFFICIENT MATRIX AP-
C                         PEARS NOT POSITIVE DEFINITE
C
C     SUBROUTINES
C     CALLED    NONE
C
C     DISCUSSION  THE ROUTINE FIRST PERFORMS THE CHOLESKY DECOMPOSITION
C               OF THE GIVEN MATRIX A INTO A LOWER TRIANGULAR MATRIX L
C               AND ITS TRANSPOSE. IT THEN SOLVES THE GIVEN SYSTEM OF EQU
C               -ATIONS BY BACK SUBSTITUTIONS.
C******
      DIMENSION A(NR,1),DN(1),B(NR,1)
C                                   PERFORMS CHOLESKY DECOMPOSITION
      DO 20 I=1,N
      KI = I-1
      DO 20 J=I,N
      X = A(I,J)
      IF(KI.GT.0)  CALL VIPS(A(I,1),NR,A(J,1),NR,KI,X)
      IF(J.NE.I)  GO TO 10
      IF(X.LE.0.)  GO TO 80
      DN(I) = 1./SQRT(X)
      GO TO 20
   10 A(J,I) = X*DN(I)
   20 CONTINUE
      D1 = 1.
C                                   BACK SUBSTITUTIONS
      NM1 = N - 1
```

```
      DO 60 J=1,M
      B(1,J) = B(1,J)*DN(1)
      IF(N.EQ.1)  GO TO 40
      DO 30 I=2,N
      Y = B(I,J)
      CALL VIPS(A(I,1),NR,B(1,J),1,I-1,Y)
      B(I,J) = Y*DN(I)
      IF(N.EQ.1)  GO TO 60
   30 CONTINUE
   40 B(N,J) = B(N,J)*DN(N)
      DO 50 IN=1,NM1
      I = N - IN
      Y = B(I,J)
      I1 = I+1
      CALL VIPS(A(I1,I),1,B(I1,J),1,IN,Y)
      B(I,J) = Y*DN(I)
   50 CONTINUE
   60 CONTINUE
   70 RETURN
   80 D1 = 0.
      GO TO 70
      END
```

```
      SUBROUTINE PTRNS (IP)
C******
C     SUBROUTINE PTRNS (IP)
C
C     PURPOSE  TO READ PANEL INFORMATION FROM DISK
C
C     INPUT    CALLING SEQUENCE
C              IP - PANEL NUMBER OF INFORMATION TO BE READ
C              COMMON BLOCK
C              /PINDX/ - KQ,NPRD
C
C     OUTPUT   COMMON BLOCK
C              /PANDQ/ - CP,PC,RC,AR,AFT,P,A,B,DIAM,C,AST,IIS,INS,ITS.
C              /PINDX/ - KQ
C
C     SUBROUTINES
C     CALLED   NONE
C
C     DISCUSSION  READS 197 WORDS OF PANEL INFORMATION FROM DISK FILE
C              SPECIFIED BY NPRD INTO COMMON BLOCK PANDQ.
C******
C
C         READS PANEL INFORMATION FROM DISK
C
      COMMON /PANDQ/ PDQ(197), NPDQ
      COMMON /PINDX/ KDUM,KP,NDUM,NPAN
C
      ID = IP - KP
      IF (ID) 200,300,100
  100 IBRANCH = ID
      GO TO 250
  200 IBRANCH = IP
      REWIND NPAN
  250 IF (IBRANCH .EQ. 1) GO TO 290
      MAX = IBRANCH - 1
      DO 275 I=1,MAX
  275 READ(NPAN)   PDQ(1)
  290 READ(NPAN)   PDQ
  300 KP = IP
      RETURN
      END
```

```
      SUBRCUTINE TDECOM(A,NR,N,V,IPR,D1)
C******
C     SUBRCUTINE TDECOM (A,NR,N,V,IPR,D1)
C
C     PURPOSE   TO DECOMPOSE A SQUARE MATRIX INTO LOWER AND UPPER TRIAN-
C               GULAR MATRICES WITH PARTIAL PIVOTING AND ROW EQUILIBRA-
C               TION
C
C     INPUT     CALLING SEQUENCE
C               A - ARRAY CONSISTS OF ELEMENTS OF A GIVEN MATRIX
C               NR - MAXIMUM ROW DIMENSION OF ARRAY A
C               N - ORDER OF THE GIVEN MATRIX
C               V - SCRATCH ARRAY, MAY BE SAME ARRAY AS IPR TO SAVE STOR
C                   -AGE
C
C     OUTPUT    CALLING SEQUENCE
C               A - THE LOWER TRIANGLE OF THE ARRAY CONSISTS OF A LOWER
C                   TRIANGULAR MATRIX L AND THE UPPER TRIANGLE CONSISTS
C                   OF AN UPPER TRIANGULAR MATRIX U (SINCE U IS UNIT UP-
C                   PER TRIANGULAR, ITS DIAGONAL ELEMENTS ARE NOT STORED)
C               IPR - ARRAY GIVES NUMBERS OF PIVOTAL ROW (A RECORD OF IN-
C                     TERCHANGES)
C               D1 - = +1 OR -1 ACCORDING AS THE NUMBER OF INTERCHANGES
C                      IS EVEN OR ODD. IT ALSO INDICATES SUCCESSFUL DECOM-
C                      POSITION
C                    = 0 INDICATES THAT THE GIVEN MATRIX APPEARS SINGULAR
C
C     SUBRCUTINES
C     CALLED    VIP,VIPS
C
C     DISCUSSION  THE ROUTINE PERFORMS THE CROUT FACTORIZATION OF A GIV-
C               EN MATRIX WITH PARTIAL PIVOTING AND ROW EQUILIBRATION.
C               THE UPPER AND LOWER TRIANGULAR MATRICES RESULTED FROM THE
C               DECOMPOSITION ARE STORED IN THE ARRAY A WHICH ORIGINALLY
C               CONSISTS ELEMENTS OF THE GIVEN MATRIX. IF ONE OF THE PI-
C               VOTS APPEARS TO BE TOO SMALL, D1 IS SET TO ZERO AND AN
C               ERROR EXIT IS TAKEN.
C                  THIS ROUTINE IS A MODIFIED VERSION OF A ROUTINE IN THE
C               SUBROUTINE LIBRARY OF THE BOEING COMPUTER SERVICES CO.
C******
      DIMENSION A(NR,1),V(1),IPR(1)
      DATA EPS/16407777777777777776B/
      E8=8.*EPS
C                                  CALCULATES LENGTH OF ROW VECTORS
      DO 10 I=1,N
      CALL VIP(A(I,1),NR,A(I,1),NR,N,Y)
      IF(Y.LE.0.)  GO TO 70
   10 V(I)=1./SQRT(Y)
C                                  PERFORMS THE DECOMPOSITION
      D1=1.
      DO 50 K=1,N
      L=K
```

```
      X=0.
      K1=K-1
      DO 20 I=K,N
      Y = A(I,K)
      IF(K1.GT.0)  CALL VIPS(A(I,1),NF,A(1,K),1,K1,Y)
      A(I,K) = Y
      Y=ABS(Y*V(I))
      IF(Y.LE.X) GO TO 20
      X=Y
      L=I
   20 CONTINUE
      IF(L.EQ.K) GO TO  35
      D1=-D1
      DO 30 J=1,N
      Y=A(K,J)
      A(K,J)=A(L,J)
   30 A(L,J)=Y
      V(L)=V(K)
   35 IPR(K)=L
C                                CHECKS IF THE PIVOT IS TOO SMALL
      IF(X.LT.EB)  GO TO 70
      X = 1./A(K,K)
      J=K+1
   40 IF(J.GT.N)  GO TO 50
      Y = A(K,J)
      IF(K1.GT.0)  CALL VIPS(A(K,1),NF,A(1,J),1,K1,Y)
      A(K,J) = X*Y
      J=J+1
      GO TO 40
   50 CONTINUE
   60 RETURN
   70 D1 = 0.
      GO TO 60
      END
```

```
      SUBROUTINE TRANS(A,AT,M,N)
C******
C     SUBROUTINE TRANS (A,AT,M,N)
C
C     PURPOSE  TO FORM THE TRANSPOSE OF A MATRIX A AND STORE
C              THE RESULT IN A MATRIX B
C
C     INPUT    CALLING SEQUENCE
C              A - ARRAY CONTAINING MATRIX ELEMENTS TO BE TRANSPOSED
C              M - NUMBER OF ROWS IN A AND COLUMNS IN B
C              N - NUMBER OF COLUMNS IN A AND ROWS IN B
C
C     OUTPUT   CALLING SEQUENCE
C              AT - ARRAY CONTAINING ELEMENTS OF THE TRANSPOSE OF
C                   THE GIVEN MATRIX
C
C     SUBROUTINES
C     CALLED   NONE
C
C     DISCUSSION  AT(J,I) IS SET TO A(I,J) AS I VARIES FROM 1 TO M AND
C                 J VARIES FROM 1 TO N.
C******
      DIMENSION A(M,N),AT(N,M)
      DO 100 I=1,M
      DO 50 J=1,N
      AT(J,I)=A(I,J)
 50   CONTINUE
 100  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TRNSFR(X,Y,N)
C******
C     SUBROUTINE TRNSFR (X,Y,N)
C
C     PURPOSE  TO MOVE A NUMBER OF ELEMENTS FROM ONE ARRAY TO ANOTHER
C
C     INPUT    CALLING SEQUENCE
C              X - LOCATION OF THE FIRST ARRAY ELEMENT TO BE MOVED
C              N - NUMBER OF ELEMENTS TO BE MOVED
C
C     OUTPUT   CALLING SEQUENCE
C              Y - ARRAY OF ELEMENTS IDENTICAL TO THE FIRST N ELEMENTS
C                  IN ARRAY X
C
C     SUBROUTINES
C     CALLED   NONE
C
C     DISCUSSION Y(I) IS SET TO X(I) AS I VARIES FROM 1 TO N.
C******
      DIMENSION X(N),Y(N)
      DO 100 I=1,N
  100 Y(I) = X(I)
      RETURN
      END
```

```
      SUBROUTINE UNIPAN(AR,RO,X,Y)
C******
C     SUBROUTINE UNIPAN (AR,RO,X,Y)
C
C     PURPOSE  TO TRANSFORM POINT COORDINATES FROM THE UNIVERSAL
C              SYSTEM TO THE LOCAL PANEL SYSTEM
C
C     INPUT    CALLING SEQUENCE
C              AR- GLOBAL TO LOCAL PANEL SYSTEM TRANSFORMATION MATRIX
C              RO- X,Y,Z COORDINATES OF PANEL CENTER (UNIVERSAL)
C              X - X,Y,Z COORDINATES OF POINT TO BE TRANSFORMED
C                  (UNIVERSAL)
C
C     OUTPUT   CALLING SEQUENCE
C              Y - X,Y,Z COORDINATES OF TRANSFORMED POINT (LOCAL)
C
C     SUBROUTINES
C     CALLED   MMULT
C
C     DISCUSSION  THE COORDINATES OF THE PANEL CENTER ARE SUBTRACTED
C              FROM THE COORDINATES OF THE POINT TO BE TRANSFORMED. THIS
C              GLOBAL ARRAY IS THEN MULTIPLIED BY THE MATRIX AR USING
C              SUBROUTINE MMULT TO PRODUCE THE LOCAL PANEL COORDINATES.
C******
      DIMENSION AR(3,3),RO(3),X(3),Y(3),W(3)
      DO 10 I=1,3
 10   W(I)=X(I)-RO(I)
      CALL MMULT(AR,W,Y,3,3,1)
      RETURN
      END
```

```
      SUBROUTINE UVECT(A)
C******
C     SUBROUTINE UVECT (A)
C
C     PURPOSE  TO CALCULATE THE DIRECTION COSINES OF A VECTOR
C
C     INPUT    CALLING SEQUENCE
C              A - DIRECTION NUMBERS OF A VECTOR
C
C     OUTPUT   CALLING SEQUENCE
C              A - DIRECTION COSINES OF A VECTOR
C
C     SUBROUTINES
C     CALLED   NONE
C
C     DISCUSSION  UVECT PERFORMS THE FOLLOWING CALCULATIONS-
C                 A(I) / SQRT(A(1)*A(1)+A(2)*A(2)+A(3)*A(3)) ,WHERE
C                 I VARIES FROM 1 TO 3.
C******
      DIMENSION A(3)
      Z=SQRT(A(1)**2+A(2)**2+A(3)**2)
      DO 10 I=1,3
 10   A(I)=A(I)/Z
      RETURN
      END
```

```
          IDENT   VIP    (A,INCA,B,INCB,N,C)
*******
*      SUBROUTINE VIP   (A,INCA,B,INCB,N,C)
*                 VIPA  (A,INCA,B,INCB,N,C)
*                 VIPS  (A,INCA,B,INCB,N,C)
*
*      PURPOSE  TO PERFORM VECTOR INNER PRODUCT CALCULATION (VIP) AND TO
*               ADD (VIPA) TO OR SUBTRACT (VIPS) FROM AN INCOMING VALUE
*               (COMPASS)
*
*      INPUT    CALLING SEQUENCE
*               A - VECTOR A
*               INCA - INCREMENT BETWEEN SUCCESSIVE ELEMENTS OF A
*               B - VECTOR B
*               INCB - INCREMENT BETWEEN SUCCESSIVE ELEMENTS OF B
*               N - NUMBER OF ELEMENTS TO BE MULTIPLIED
*               C - AN INCOMING VALUE TO BE ADDED TO (VIPA) OR TO BE SUB-
*                   TRACTED FROM (VIPS)
*
*      OUTPUT   CALLING SEQUENCE
*               C - RESULT  C = A.B (VIP), C = C + A.B (VIPA), AND
*                   C = C - A.B (VIPS)
*
*      SUBROUTINES
*      CALLED   NONE
*
*      DISCUSSION  THE INNER PRODUCT OF TWO VECTORS A AND B IS CALCULATED
*               AND STORED IN C (VIP).  THE RESULT IS ADDED TO (VIPA) OR
*               SUBTRACTED FROM (VIPS) AN INCOMING VALUE C AND THE SUM OR
*               THE DIFFERENCE IS STORED BACK IN C.
*                  THIS ROUTINE IS A MODIFIED VERSION OF A COMPASS ROUT-
*               INE IN THE SUBROUTINE LIBRARY OF THE BOEING COMPUTER SER-
*               VICES CO.
*******
          ENTRY     VIP,VIPA,VIPS
          VFD       36/0HVIPA,24/6
 VIPA     BSSZ      1
          SA4       ENDVA               FETCH TRANSFER
          SA1       B1                  FETCH PRE-OP
          ZR        B0,GO
 +        VFD       36/0HVIPS,24/6
 VIPS     BSSZ      1
          SA4       ENDVS               FETCH TRANSFER
          SA1       B1                  FETCH PRE-OP
          ZR        B0,GO
 +        VFD       36/0HVIP,24/6
 VIP      BSSZ      1
          SA4       ENDV                FETCH TRANSFER
          SA1       B1                  FETCH PRE-OP
 GO       SA2       B3                  FETCH POST-OP
          BX7       X4                  PUT TRANSFER IN X7
          IX0       X0-X0               CLEAR PRODUCT REGISTER
```

```
          SX6     B0             CLEAR SUMMING REGISTER
          SA7     THRU           STORE TRANSFER
          SA3     B2        .GET INCA IN X3
          SA4     B4        .GET INCB IN X4
          SA5     B5             GET N IN X5
          SB2     X3        .PUT INCA IN B2
          SB4     X4        .PUT INCB IN B4
          SB5     X5             PUT N IN B5
          SX7      B0            CLEAR FOR FIRST ADD OF X6+X7
          SB7      B0+2          GET 2 FOR DECR N.
          SX0     B0+1           MASK TO DETERMINE ODD/EVEN
          BX0      X0*X5         MASK LOWER BIT
          ZR       X0,EVEN       IF EVEN, BY PASS ODD ELEM SETUP
          FX0      X1*X2         ODD, DO FIRST ELEMENT
          SB5     B5-1           DECR N
          ZR      B5,DONE        IF N=1,DONE
          FX5      X0+X6         ADD FIRST TERM IN ODD CASE
          SA1      A1+B2         GET NEXT ELEM OF PRE-OP
          SA2      A2+B4         GET NEXT ELEM OF POST-OP
          NX6      X5            NORMALIZE FIRST ADD
EVEN      LX3      1             GET INCA*2
          LX4      1             GET INCB*2
          SB1      X3            B1= 2*INCA
          SB3      X4            B3= 2*INCB
          SA3      A1+B2         GET 2ND ELEM OF PRE-OP
          SA4      A2+B4         GET 2ND ELEM OF POST-OP
          SB2      B1            B2= 2*INCA
          SB4      B3            B4= 2*INCB
          ZR       B0,START
OVER      SA1     A1+B2     .LOAD PRE-OP
          RX5      X0+X6         SUM FIRST MULTIPLY
          SA2     A2+B4     .LOAD POST-OP
          SA3      A3+B2         LOAD 2ND PRE-OP
          NX6      X5            NORMALIZE FIRST ADD
          SA4      A4+B4         LOAD 2ND POST-OP
START     SB5     B5-B7          DECREMENT N
          FX5      X6+X7         SUM 2ND MPY (DUMMY FIRST TIME)
          RX0      X1*X2         FIRST MULTIPLY
          RX7      X3*X4         SECOND MULTIPLY
          NX6      X5            NORMALIZE SECOND ADD
          NZ      B5,OVER        DONE WHEN N=0
DONE      RX5      X0+X6         FINAL ADD - FIRST MPY
          NX6      X5            NORM
          RX5      X7+X6         FINAL ADD - 2ND MPY(IF N=1, X7=0)
          NX6      X5            NORM
THRU      BSSZ    1              STUFFED WITH A TRANSFER
OUTVA     SA1     B6             FETCH C
          RX7     X1+X6          ADD C
          NX6     X7
          SA6     B6             STORE
          ZR      B0,VIPA
OUTVS     SA1     B6
```

```
         RX7     X1-X6          SUB  INNER PRODUCT
         NX6     X7
         SA6     B6
         ZR      B0,VIPS
ENDVA    ZR      B0,OUTVA
ENDVS    ZR      B0,OUTVS
ENDV     SA6     B6
         ZR      B0,VIP
         END
```

```
          IDENT   ZERO        (A,N)
*******
*      SUBROUTINE  ZERO (A,N)
*
*      PURPOSE   TO SET THE ELEMENTS OF AN ARRAY TO ZERO (COMPASS)
*
*      INPUT     CALLING SEQUENCE
*                A - LOCATION OF FIRST ELEMENT TO BE SET TO ZERO
*                N - NUMBER OF ELEMENTS TO BE SET TO ZERO
*
*      OUTPUT    CALLING SEQUENCE
*                A - ARRAY OF ZERO ELEMENTS
*
*      SUBROUTINES
*      CALLED    NONE
*
*      DISCUSSION  A(I) IS SET TO ZERO AS I VARIES FROM 1 TO N.
*******
 +        VFD       36/0HZERO,24/6
          ENTRY     ZERO
 ZERO     BSSZ      1
          SA2       B2
          SB4       X2
          SB3       B0
          SB5       1
          SX6       B0
 CYCLE    SA6       B1+B3
          SB3       B3+B5
          LT        B3,B4,CYCLE
          EQ        ZERO
          END
```


Boeing Commercial Airplane Company
P.O. Box 3707
Seattle, Washington 98124
September 24, 1975

APPENDIX

ENGINEERING AND PROGRAM VARIABLES

Engineering variable	Description	Program variable	Reference program or subroutine
a	Parameter of Smith's conical solution	A	SHEGEN
A_{kj}	Aerodynamic influence coefficient	DVDFS	VINFCC
$[B]$	Matrix defined by eq. (42) of Engineering Document	VS	VINFCC
c_m	Pitching moment coefficient	CM	OUTPUT
c_N	Normal force coefficient	CN	OUTPUT
c_p	Pressure coefficient	CPU,CPL	OUTPUT
Δc_p	Jump in pressure coefficient	DCP	OUTPUT
$[c]$	Matrix for coefficients of quadratic doublet distribution on a panel, defined by eq. (23) of Engineering Document	AST	SING
$[D]$	Matrix, defined by eq. (19) of Engineering Document	DVDV	PIVC
$[DK]$	Matrix, defined by eq. (40) of Engineering Document	DVDS	PIVC
E	Function, see eq. (46) of Engineering Document	EMUE, EMU	FGCAL
$f^{(i)}$	Vector, defined by eq. (52) of Engineering Document	RX	ITFLOW
F	Function, see eq. (46) of Engineering Document	FVZ	FGCAL
G	Function, see eq. (46) of Engineering Document	GVZ	FGCAL
J	Jacobian	AJ	ITFLOW

Engineering variable	Description	Program variable	Reference program or subroutine
$[K]$	Matrix, defined by eq. (37) of Engineering Document	DVS	PIVC
$\ell_{i,m}$	Chord length of panel segment in transverse geometry cut	ZL	INPUT
M	Number of panels on one-half of configuration	NPANT	INPUT
N	Number of doublet parameters in neighborhood of panel	INS	SING
N_D	Number of doublet parameters on one-half of configuration	NSNGT	TSING
N_{FS}	Number of free sheet panels on one-half of the configuration	NP(2)	INPUT
N_W	One-half of the number of wing panels	NP(1)	INPUT
P_{ij}	Panel corner point	CP	GEOMC
Po_{ij}	Panel center	PC	SURFIT
$Q(\xi,\eta)$	Position of elementary doublet	ZPK	SING
$S(X)$	Local wing semispan	YLE	INPUT
S	Panel area	SP	OUTPUT
S_W	Wing area	SW	OUTPUT
$\vec{U}_\infty$	Freestream velocity	FSV	INPUT
$\vec{V}$	Velocity	VEL VG	OUTPUT FGCAL
$\vec{V}^s$	Average sheet velocity	VEL VG	OUTPUT FGCAL
X_P	Pitch axis	XPITCH	INPUT

Engineering variable	Description	Program variable	Reference program or subroutine
α	Angle of attack	ALPHA	INPUT
$\bar{\gamma}$	Constant in quasi-Newton method	GAMA	ITFLOW
$\vec{\sigma}$	Vorticity	Z	OUTPUT
Γ	Strength of line vortex along terminated edge	TSC(1)	OUTPUT
$\delta^{(i)}$	Scaling parameter	CALFA	ITFLOW
θ	Panel inclination in transverse cut	ZA	INPUT
μ_e	Doublet parameters along edges of networks	S(1), . . . S(NEQ)	FGCAL
μ_r	Doublet parameters not located along edges of networks	S(NEQ+1), . S(NSNGT)	FGCAL
μ_j	Doublet parameters on one-half of configuration	S	TEA378 FGCAL OUTPUT

REFERENCES

1. Weber, J. A.; Brune, G. W.; Johnson, F. T.; Lu, P.; and Rubbert, P. E.: "A Three-Dimensional Solution of Flows Over Wings With Leading-Edge Vortex Separation." AIAA paper 75-866, presented at the AIAA Eighth Fluid and Plasma Dynamics Conference, Hartford, Connecticut, 16-18 June 1975.

2. Johnson, F. T.; and Rubbert, P. E.: "Advanced Panel-Type Influence Coefficient Methods Applied to Subsonic Flows." AIAA paper 75-50, presented at the AIAA 13th Aerospace Science Meeting, Pasadena, California, 20-22 January 1975.