

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(12)

AD A 1 23285



A MICROCOMPUTER-BASED METHOD FOR
 PREDICTING VISCOUS FLOW
 ABOUT A SHIP HULL

DCW INDUSTRIES

DTIC
 SEL
 JAN 12 1983
 A

This document has been approved
 for public release and sale; its
 distribution is unlimited.

4367 TROOST AVENUE, STUDIO CITY, CALIFORNIA 91604

88 01 12 040

DTIC FILE COPY

REPORT ONR-CR062-735-1F



A MICROCOMPUTER-BASED METHOD FOR
PREDICTING VISCOUS FLOW
ABOUT A SHIP HULL

David C. Wilcox

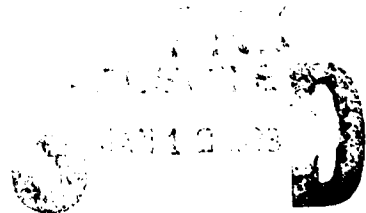
DCW INDUSTRIES, INC.
Studio City
California
91604

CONTRACT N00014-82-C-0432
ONR TASK 062-735

December 1982

FINAL REPORT

Approved for public release; distribution unlimited.



A



PREPARED FOR THE
OFFICE OF NAVAL RESEARCH ● 800 N. QUINCY ST. ● ARLINGTON ● VA ● 22217

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ONR-CR062-735-1F	2. GOVT ACCESSION NO. AD-A123285	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A MICROCOMPUTER-BASED METHOD FOR PREDICTING VISCOUS FLOW ABOUT A SHIP HULL	5. TYPE OF REPORT & PERIOD COVERED FINAL REPORT 20 MAY 1982 - 15 NOV 1982	
	6. PERFORMING ORG. REPORT NUMBER DCW-R-28-01	
7. AUTHOR(s) DAVID C. WILCOX	8. CONTRACT OR GRANT NUMBER(s) N00014-82-C-0432	
9. PERFORMING ORGANIZATION NAME AND ADDRESS DCW INDUSTRIES, INC. 4367 Troost Avenue Studio City, CA 91604	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR 062-735	
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Code 430, 800 North Quincy Street Arlington, Virginia 22217	12. REPORT DATE DECEMBER 1982	
	13. NUMBER OF PAGES 97	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report) UNCLASSIFIED	
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
16. DISTRIBUTION STATEMENT (of this Report) Approved for Public Release, Distribution Unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) SHIP HYDRODYNAMICS, THREE-DIMENSIONAL BOUNDARY LAYERS, MICROCOMPUTERS		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A microcomputer-based computational method has been devised for numerically simulating viscous flow past an arbitrary ship hull. The method uses (a) thin-ship theory to compute hull-surface velocities and (b) a three-dimensional boundary-layer program which embodies the Wilcox-Rubesin two-equation turbulence model to compute boundary-layer development. The method makes provision for including displacement effects.		

DD FORM 1473

1 JAN 73

EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

-11- SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

CONTENTS

SECTION	PAGE
ABSTRACT.....	11
CONTENTS.....	iv
1 INTRODUCTION.....	1
2 ANALYSIS.....	3
2.1 Overview of the Computational Procedure.....	3
2.2 Program Input/Output.....	5
2.2.1 Program PRMESH.....	5
2.2.2 Program SPLOT.....	8
2.2.3 Program SHIP.....	10
2.2.4 Program SHPMSH.....	12
2.2.5 Program VELOC.....	15
2.2.6 Program EDDY3.....	18
2.2.7 Program INTERP.....	21
2.3 Applications.....	22
2.3.1 Michell's Example.....	22
2.3.2 SSPA Model 720.....	23
3 DISCUSSION.....	30
REFERENCES.....	32
APPENDIX: PROGRAM LISTINGS.....	33
PRMESH.....	34
SPLOT.....	36
SHIP.....	39
SHPMSH.....	46
VELOC.....	56
EDDY3.....	63
INTERP.....	97

1. INTRODUCTION

During the past five years the microcomputer has developed into a powerful computational aid to the engineer. Currently, Z80-based microcomputers are capable of performing most day-to-day numerical-fluid-mechanics tasks such as data preparation/reduction, two-dimensional boundary-layer computations with very fine finite-difference grids, and many other practical computations including, on relatively crude finite-difference grids, time-averaged Navier-Stokes solutions. With the exception of the latter example, these computations usually can be done as fast on a microcomputer as running on a scientific mainframe computer, if you include the time required for transmitting the input data and final printout via a terminal interface.

This is true because of the overall real time spent using a mainframe computer with the microcomputer as a terminal, typically only 5% is devoted to the actual execution of the program. The other 95% consists of signing on, transmitting input data, and eventually transmitting the final printout. Obviously the time distribution varies depending on the overall execution time and the amount of print taken. However, our experience over a two-year period has shown a 5%-95% split to be quite representative for numerical-fluid-mechanics applications.

On the basis of the great power of modern microcomputers, this project has been undertaken with the overall aim of developing a microcomputer-based computational procedure for predicting viscous flow about a ship hull. The theoretical foundation of the procedure is (a) classical thin-ship theory¹ for the inviscid velocity distribution on the ship hull and (b) a three-dimensional boundary-layer program, EDDY3, which incorporates the Wilcox-Rubesin² two-equation turbulence model. In a straightforward manner, the computational procedure corrects for boundary-layer displacement effects. With the exception of the boundary-layer computation, all phases of the

procedure are accomplished on a TRS-80 Model II (Z80-based) micro-computer. While the boundary-layer computations could also be done on the TRS-80, run times are too long (30 hours) to render such runs useful in a design environment.

With the exception of a plotting routine (SPLOT) and an interpolation routine (INTERP), all programs have been developed in FORTRAN (ANSI-66) and, with trivial modifications, can thus be run on virtually all scientific computers.

In Section 2 we first present an overview of the computational procedure. Next we describe the various programs including input and output (program listings are located in the Appendix). Then, we present results of two sample computations.

Section 3 summarizes and discusses results of the project.

2. ANALYSIS

2.1 OVERVIEW OF THE COMPUTATIONAL PROCEDURE

In devising the computational procedure, our plan has been to create a method which lays the foundation for a ship designer to effect design modifications in an interactive mode. Because we are creating the foundation, the procedure has been developed in several discrete modules each of which performs a straightforward task.

Our starting point is a disk file containing the three-dimensional hull coordinates from bow to stern. This file presumably has been created by the designer. As illustrated in Figure 1, we first use program PRMESH to generate a preliminary orthogonal, rectangular mesh for use in computing the inviscid velocities on the hull. Before proceeding to computation of the hull velocities, program SPLOT can be used to visually inspect the coordinates via the micro-computer video display. This step provides an opportunity to spot obvious input data errors. Once we are confident the input data are satisfactorily smooth, we use program SHIP to compute the inviscid velocities on the ship-hull surface. This completes the inviscid phase of the computational procedure.

The second (viscous) phase of the procedure begins with program SHPMSH which generates a nonorthogonal finite-difference grid for use in the 3-D boundary-layer computation, including all required geodesic curvatures and metrics. Then, using program VELOC we interpolate the inviscid velocities onto the nonorthogonal grid. At this point, all required input data for program EDDY3 have been prepared. The final step in the viscous phase is execution of program EDDY3 which computes boundary-layer development over the hull.

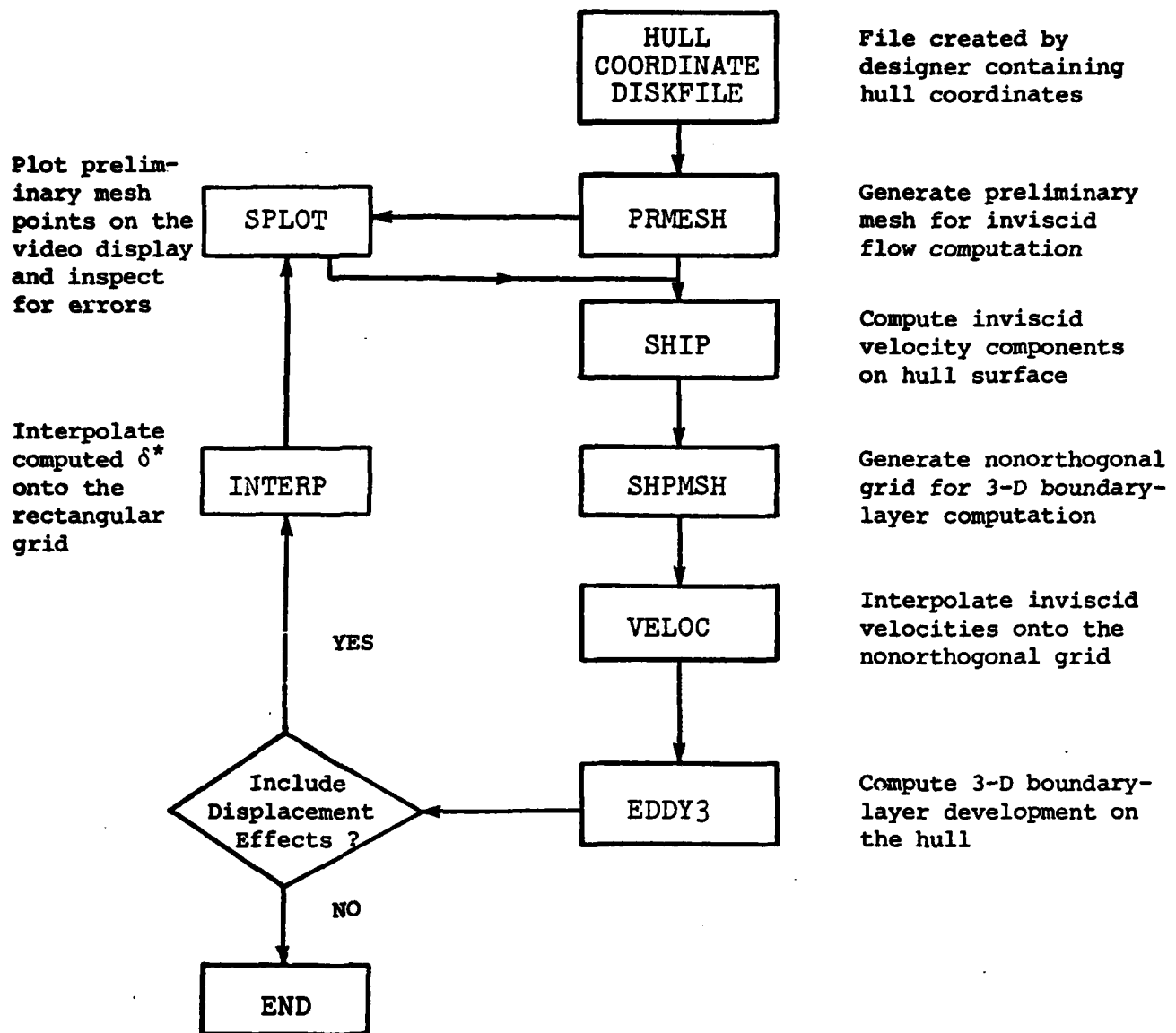


Figure 1. Schematic representation of the computational procedure.

Upon completion of the viscous phase, the designer can opt to include effects of boundary-layer displacement. Should this option be selected, a disk file is available from EDDY3 which contains the computed displacement thicknesses on the nonorthogonal grid. Program INTERP interpolates the contents of this file onto the rectangular grid used by program SHIP. The complete computational sequence is then repeated.

Certainly visual inspection of the inviscid velocities, nonorthogonal grid coordinates, final output from EDDY3, etc. would be helpful, if not essential, in a design application. Simple modifications to SPLOT, for example, would provide the vehicle for such inspection. Even more powerful microcomputer graphics techniques are available for both video and hard-copy displays and there seems little point to going to great lengths at this point to rival existing software. Thus, we have not devised the needed modifications because of the preliminary nature of this project which has been, to some extent, a feasibility study. Nevertheless, the capability for inspection of the input data has been included to give an indication of the practicability of a microcomputer-based approach.

2.2 PROGRAM INPUT/OUTPUT

We turn now to the input and output of the seven program modules. The Appendix contains complete listings of each program. In the interest of clarity, for each program, we reiterate its overall purpose below.

2.2.1 Program PRMESH

The purpose of this program is to generate a rectangular finite-difference grid for use in computing the inviscid velocity components.

Program input data consists of two disk files, viz, MESH and PRMESHIN. The first of these two files contains the coordinates \bar{x} , \bar{y} , \bar{z} of

the ship (Figure 2), where the coordinates are most conveniently supplied in nondimensional form (normalized, for example, by hull length, L). Note that flow past the hull is toward positive \bar{x} , positive \bar{y} is to starboard, and \bar{z} increases as we move from keel to waterline.

In disk file MESH, the coordinates are presented on constant- \bar{x} sections with each line giving two pairs of coordinates. The format is (3F10.5,1X,3F10.5). Note that the space between coordinate pairs (the 1X) is included for consistency with the input data format specified for the 1980 ITTC/SSPA Workshop on Ship Boundary Layers. Thus, a few typical lines of data assume the following form:

SAMPLE 'MESH" INPUT DATA					
\bar{x}	\bar{y}	\bar{z}	\bar{x}	\bar{y}	\bar{z}
-0.40000	-0.13152	-0.07847	-0.40000	-0.13425	-0.06825
-0.40000	-0.13603	-0.05375	-0.40000	-0.13668	-0.03680
-0.40000	-0.13705	-0.01872	-0.40000	-0.13738	0.00000
-0.35000	0.00000	-0.11808	-0.35000	-0.00690	-0.11803
-0.35000	-0.01498	-0.11790	-0.35000	-0.02537	-0.11767
-0.35000	-0.03952	-0.11732	-0.35000	-0.05698	-0.11668

Note that points on each section are input from keel (where $\bar{y} = 0$) to waterline (where $\bar{z} = 0$). In its current form, program PRMESH will accept data on up to 50 sections with a maximum of 20 (\bar{y}, \bar{z}) pairs per section. One restriction on the input data is that an even number of (\bar{y}, \bar{z}) pairs must be prescribed. Note also that the input coordinates must define the ship hull from bow to stern; a partial mesh is insufficient.

The second input disk file, PRMESHIN, contains the following data:

- IMAX = Number of sections (constant \bar{x}) in the computed rectangular mesh (cannot exceed 41).
- JMAX = Number of points per section in the computed rectangular mesh (cannot exceed 20).
- NXTL = Number of sections in the input mesh of disk file MESH (cannot exceed 50).
- NZTL = Number of (\bar{y}, \bar{z}) pairs per section in the input mesh of disk file MESH (cannot exceed 20).

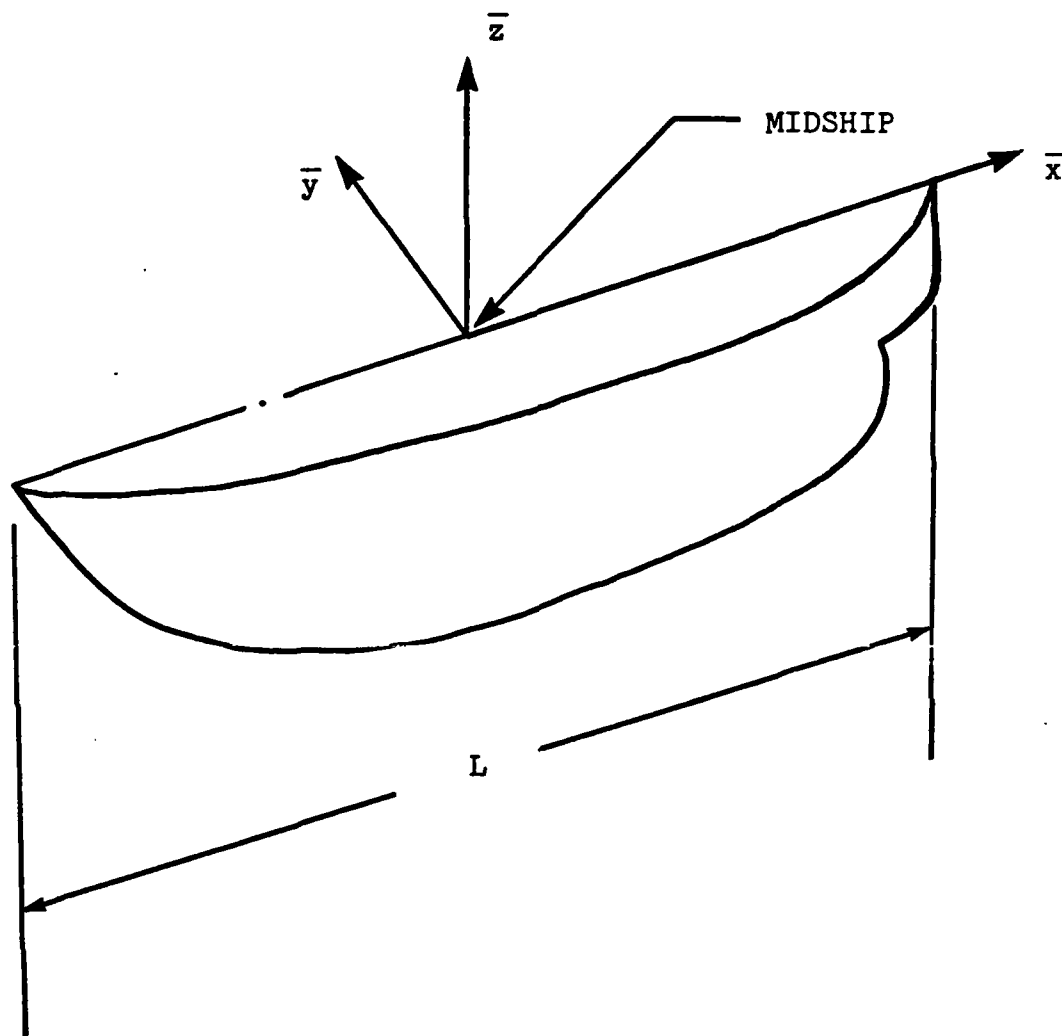


Figure 2. Schematic of ship-hull coordinate system.

- X = Specified streamwise (\bar{x} -coordinate) array for the computed rectangular mesh (a total of IMAX values).
- Z = Specified vertical (\bar{z} -coordinate) array for the computed rectangular mesh (a total of JMAX values).

The input format for this disk file has been designed to resemble NAMELIST input. For IMAX, JMAX, NXTL and NZTL, the format is (1X,3A4,I4) while for X and Z it is (1X,3A4,E13.6). A typical input file is shown below:

SAMPLE 'PRMESHIN' INPUT DATA

```

IMAX      =      5
JMAX      =      4
NXTL     =     40
NZTL     =     20
X(1)     = -0.500000E 00
X(2)     = -0.250000E 00
X(3)     =  0.000000E 00
X(4)     =  0.250000E 00
X(5)     =  0.500000E 00
Z(1)     =  0.000000E 00
Z(2)     = -0.030000E 00
Z(3)     = -0.065000E 00
Z(4)     = -0.100000E 00

```

As illustrated, the (1X3A4) part of the format permits input of the variable name. Unlike NAMELIST input, all variables must be input and the order of input cannot be mixed. An exception to this is that only IMAX (JMAX) values of X(Z) must be input; the unused parts of these arrays needn't be defined.

Program output is (1) a printed listing of input disk file PRMESHIN and (2) a disk file named MESHPR containing the coordinates of the computed rectangular mesh. Inspection of the latter can be accomplished with program SPLOT described in the next Subsection.

2.2.2 Program SPLOT

The purpose of program SPLOT is to permit section-by-section visual inspection of the rectangular mesh generated by program PRMESH.

Such an inspection will aid in identifying smoothness of the mesh as well as input data errors.

The program is written in BASIC, makes use of special features of the TRS-80 Model II, and cannot be readily adapted to another machine. However, for reasons noted in the Introduction, it has been included for demonstration purposes and is not essential in the overall computational procedure. Input to the program consists of (1) prompted requests for IMAX and JMAX and (2) disk file MESHPR. Also, as the visual inspection proceeds, additional prompted requests are made as will be discussed below.

All input and output appears on the video display. Upon execution of SPLOT, the program will print "IMAX=?". In response, the value of IMAX used in running PRMESH must be entered. Next the message "JMAX=?" is displayed. Again, the value used in running PRMESH must be entered.

The program will now proceed section by section. On each section it will print the streamwise section number and the value of \bar{x} on the section. This will be followed by a printout identifying the maximum and minimum values of \bar{y} and \bar{z} on the section. The program then requests the appropriate plotting range, viz, the desired values of \bar{y}_{\min} , \bar{y}_{\max} , \bar{z}_{\min} , \bar{z}_{\max} . After the first section has been investigated, the program first asks if the same plotting range used for the previous plot is desired. Entering "Y" obviates the need to reenter \bar{y}_{\min} , etc. Entering "N" permits changing the plotting range. This feature has been included to permit viewing all sections on the same scale.

The plot for a given section will be displayed until a character is entered. The following options are available:

- C = Continue inspection on this section;
- N = Next section - proceed to same;

For any other ASCII character, the run terminates. The "C" option permits changing the plotting range. This might be done to expand

the scale in a particular region for closer inspection. Note that if this is done, the entered \bar{y}_{\min} , etc. become the new plotting range for all following sections. Hence, returning to the original plotting range requires the appropriate input once the user proceeds to the next section. The "C" option can be used as many times as desired on each section.

As a safeguard against incorrect entries, the program asks if "ANY ERRORS ?" have been made each time \bar{y}_{\min} , etc. are entered. If no plotting-range errors have been made, enter "N" and the program continues. If an error has been made, enter "Y" and the correct values can then be typed in. Finally, any points which are off scale will be indicated in the lefthand margin of the plot.

When the user is satisfied the mesh is sufficiently smooth and free of errors, he/she is ready to proceed to computation of the inviscid flow. This end is accomplished with the program described in the next subsection.

2.2.3 Program SHIP

The purpose of program SHIP is to compute the inviscid velocity components on the surface of the ship hull. As an auxiliary output, the program computes the wave drag.

Input to program SHIP consists of two disk files, viz: (1) file MESHPR created by program PRMESH; and (2) disk file SHIPIN. The latter file has again been created in a format similar to NAMELIST. The input data are:

- FR = Froude number defined by $F_r = U/\sqrt{gL}$ where U is free-stream velocity, g is gravitational acceleration and L is the length scale used to nondimensionalize the ship coordinates x,y,z.
- IMAX = Number of sections (constant \bar{x}) in the rectangular mesh (same value used in disk file PRMESHIN - cannot exceed 41);
- JMAX = Number of points per section in the rectangular mesh (same value used in disk file PRMESHIN - cannot exceed 10);

- MINT2 = Number of points used in Chebyshev quadrature to evaluate an integral whose dimensionless wave-number range is 1 to ∞ (cannot exceed 80);
- MINT3 = Number of points used in Chebyshev quadrature to evaluate an integral whose dimensionless wave-number range is zero to 1 (cannot exceed 80);
- NZOT = Print flag. For zero the cell geometry and velocity components are printed. Otherwise, only the wave drag is printed.

As with disk file PRMESHIN of Subsection 2.2.1, the format is (1X,3A4,E13.6) for FR and (1X,3A4,I4) for all integer input quantities. A typical input file is as follows

SAMPLE 'SHIPIN' INPUT DATA

```
FR          = 2.500000E-01
IMAX        = 5
JMAX        = 4
MINT2       = 20
MINT3       = 20
NZOT        = 0
```

Note that the maximum value of JMAX allowed has been limited to 10 to keep execution times on a TRS-80 Model II within practical limits.

Output from program SHIP consists of: (1) video messages indicating when each of three stages of the computation is completed; (2) printed output depending on the value assigned to input flag NZOT; and (3) a disk file named UEWE containing mesh coordinates and computed velocity components.

The first part of the printout is diskfile SHIPIN. This provides an immediate check that input data have been correctly entered.

Printed messages during the computation appear on the video display upon completion of calls to subroutines VELOC1, VELOC2 and VELOC3. The message after completing a call to VELOCn is "VELOCn COMPUTATIONS

COMPLETE". These messages permit the user to monitor the progress of the run.

The next part of the printout depends upon the value assigned to NZOT. If NZOT is zero the program prints: the three coordinates of the rectangular cell centroids; the computed dimensionless velocity components u/U , v/U , w/U ; and the dimensionless total velocity q/U . The print proceeds section by section including incremental wave drag on the section and total wave drag computed up to the section.

The final part of the printout is the total dimensionless wave drag (RWAVE) defined in terms of the drag R by

$$RWAVE = \frac{R}{\frac{1}{2}\rho U^2 L^2} \quad (1)$$

where ρ is density, while U and L are as defined above.

Finally, diskfile UEWE is written for use by program VELOC described in Subsection 2.2.5. Upon completion of this phase of the computation, we have determined the inviscid velocity components on an orthogonal rectangular mesh.

2.2.4 Program SHPMSH

The purpose of this program is to generate a nonorthogonal finite-difference mesh for use by the three-dimensional boundary-layer program EDDY3 (Subsection 2.2.6). The program generates least-squares cubic-spline fits⁴ to accurately and smoothly compute all required geodesic curvatures and metric coefficients.

Input data for SHPMSH consists of two diskfiles, viz, MESH and NAMLIST. Diskfile MESH is the original file containing the ship-hull coordinates. In the interest of economy, this file needn't cover the entire hull. Rather, it need only cover the part of the hull over which the boundary layer is to be computed. Diskfile

NAMLIST contains the following input data:

- ISYM = Symmetry flag. To force mesh symmetry ($d\bar{y}/d\bar{z}=0$) at the waterline use ISYM=0. Otherwise the program automatically infers the appropriate value of $d\bar{y}/d\bar{z}$ at the waterline.
- IU1 = Logical unit number for input diskfile MESH.
- IU2 = Logical unit number for temporary diskfile SCRATCH1.
- IU3 = Logical unit number for temporary diskfile SCRATCH2.
- IU4 = Logical unit number for output diskfile GEOMETRY.
- IU5 = Logical unit number for final printout.
- IMAX = Number of sections in the input mesh of diskfile MESH (cannot exceed 40).
- JMAX = Number of (\bar{y}, \bar{z}) pairs per section in the input mesh of diskfile MESH (cannot exceed 40).
- NXTL = Number of sections in the computed nonorthogonal mesh (cannot exceed 40).
- NZTL = Number of points, per section in the nonorthogonal mesh (cannot exceed 21).
- JFZ = Number of key points used for the least-squares cubic-spline fit in the girthwise direction (cannot exceed 30).
- LZ = Array of indices of the key points to be used in the girthwise direction.
- JFX = Number of key points used for the least-square cubic-spline fit in the streamwise direction (cannot exceed 30).
- LX = Array of indices of the key points to be used in the streamwise direction.
- PSIZ = Nondimensional distance (measured from the bow) of the most upstream section in the nonorthogonal mesh.
- XSTART = Value of \bar{x} at the most upstream section in the nonorthogonal mesh.
- XEND = Value of \bar{x} at the most downstream section in the nonorthogonal mesh.
- DX = Array of streamwise stepsizes, $\Delta\bar{x}_1$, for the nonorthogonal mesh (cannot exceed 10).
- NX = Array of indices indicating the number of times $\Delta\bar{x}_1$ is repeated, viz, the mesh will have $\Delta\bar{x}=\text{DX}(1)$ for the first NX(1) stations, $\Delta\bar{x}=\text{DX}(2)$ for the next NX(2) Stations, etc.

- DZ = Array of girthwise stopsizes, $\Delta \bar{z}_j$, for the non-orthogonal mesh (cannot exceed 10). Note that the mesh is constructed so that $\bar{z}=0$ on the keel and $\bar{z}=1$ on the waterline.
- NZ = Array of indices indicating the number of times $\Delta \bar{z}_j$ is repeated.

As in programs PRMESH and SHIP, the format for integer quantities is (1X,3A4,I4) while floating point variables use (1X,3A4,E13.6). Sample input is as follows:

SAMPLE 'NAMLIST' INPUT DATA

ISYM	=	1	DX(5)	=	0.000000E 00
IU1	=	7	DX(6)	=	0.000000E 00
IU2	=	8	DX(7)	=	0.000000E 00
IU3	=	9	DX(8)	=	0.000000E 00
IU4	=	10	DX(9)	=	0.000000E 00
IU5	=	2	DX(10)	=	0.000000E 00
IMAX	=	30	NX(1)	=	5
JMAX	=	18	NX(2)	=	2
NXTL	=	15	NX(3)	=	7
NZTL	=	11	NX(4)	=	0
JFZ	=	8	NX(5)	=	0
LZ(1)	=	1	NX(6)	=	0
LZ(2)	=	3	NX(7)	=	0
LZ(3)	=	5	NX(8)	=	0
LZ(4)	=	7	NX(9)	=	0
LZ(5)	=	10	NX(10)	=	0
LZ(6)	=	12	ZSTART	=	0.000000E 00
LZ(7)	=	15	ZEND	=	1.000000E 00
LZ(8)	=	18	DZ(1)	=	1.000000E-01
JFX	=	12	DZ(2)	=	0.000000E 00
LX(1)	=	1	DZ(3)	=	0.000000E 00
LX(2)	=	3	DZ(4)	=	0.000000E 00
LX(3)	=	6	DZ(5)	=	0.000000E 00
LX(4)	=	9	DZ(6)	=	0.000000E 00
LX(5)	=	12	DZ(7)	=	0.000000E 00
LX(6)	=	15	DZ(8)	=	0.000000E 00
LX(7)	=	18	DZ(9)	=	0.000000E 00
LX(8)	=	21	DZ(10)	=	0.000000E 00
LX(9)	=	24	NZ(1)	=	10
LX(10)	=	26	NZ(2)	=	0
LX(11)	=	28	NZ(3)	=	0
LX(12)	=	30	NZ(4)	=	0
PSIZ	=	3.000000E-01	NZ(5)	=	0
XSTART	=	-7.000000E-01	NZ(6)	=	0
XEND	=	9.000000E-01	NZ(7)	=	0
DX(1)	=	1.000000E-01	NZ(8)	=	0
DX(2)	=	1.500000E-01	NZ(9)	=	0
DX(3)	=	1.000000E-01	NZ(10)	=	0
DX(4)	=	0.000000E 00			

Output from program SHPMSH consists of: video display messages indicating when various phases of the computation are complete; printed output of computed quantities; and diskfile GEOMETRY for use by programs VELOC (Subsection 2.2.5) and EDDY3 (Subsection 2.2.6).

The first printed output is a listing of diskfile NAMLIST which permits checking for input data errors.

Three messages are then printed on the video display during the run to permit monitoring the progress of the run. The messages are:

1. "All Section Data Smoothed and Differentiated";
2. "Geometrical Parameters Computed for J=j";
3. " $d(\text{Theta})/dz$ Computed for All Sections".

The second message is repeated for each girthwise station where j varies from 1 to JMAX.

The next output is the computed quantities along each girthwise station. This information is routed to the printer if $IU5=2$ or saved on disk provided $IU5$ is greater than or equal to 7. The data is saved on diskfile FORTu5/DAT where "u5" is the numerical value of $IU5$ when $IU5 \geq 7$. This option permits reiterating to include displacement effects via program INTERP (Subsection 2.2.7).

The final output is diskfile GEOMETRY which is used by programs VELOC and EDDY3.

2.2.5 Program VELOC

The purpose of this program is to interpolate the inviscid velocities onto the nonorthogonal coordinate mesh for use in the three-dimensional boundary-layer computation.

Input data for this program consists of three diskfiles, viz, NAMVEL, UEWE and GEOMETRY. Diskfiles UEWE and GEOMETRY are created by programs SHIP and SHPMSH, respectively. Diskfile NAMVEL

contains the following input data:

- ISYM = Symmetry flag. To treat the waterline as a symmetry plane, use ISYM=0. Otherwise, the program will automatically infer velocities at the free surface.
- IU1 = Logical unit number for input diskfile UEWE.
- IU2 = Not used.
- IU3 = Logical unit number for output diskfile VELOCITY.
- IU4 = Logical unit number for input diskfile GEOMETRY.
- IU5 = Logical unit number for final printout.
- IMAX = Number of sections in the input mesh of diskfile UEWE (cannot exceed 40).
- JMAX = Number of points per section in the input mesh of diskfile UEWE (cannot exceed 19).
- NXTL = Number of sections in the nonorthogonal mesh of diskfile GEOMETRY (cannot exceed 40).
- NZTL = Number of points per section in the nonorthogonal mesh of diskfile GEOMETRY (cannot exceed 21).
- JFZ = Number of key points for the least-squares cubic-spline fit in the girthwise direction (cannot exceed 30).
- LZ = Array of indices of the key points to be used in the girthwise direction.
- JFX = Number of key points used for the least-squares cubic-spline fit in the streamwise direction (cannot exceed 30).
- LX = Array of indices of the key points to be used in the streamwise direction.

Note that because program SHIP does not provide keel or waterline velocities, this program computes both. In so doing, the program replaces JMAX by JMAX+2. Also, JFZ, JFX, LZ and LX needn't be the same as used in program SHPMSH. As in all similar input data files of preceding programs, the format is (1X,3A4,I4) for integer input quantities and (1X,3A4,E13.6) for floating point data. A typical listing of diskfile NAMVEL follows.

SAMPLE 'NAMVEL' INPUT DATA

ISYM	=	1	LZ(6)	=	13
IU1	=	7	LZ(7)	=	16
IU2	=	8	LZ(8)	=	19
IU3	=	9	JFX	=	12
IU4	=	10	LX(1)	=	1
IU5	=	2	LX(2)	=	3
IMAX	=	31	LX(3)	=	6
JMAX	=	17	LX(4)	=	9
NXTL	=	15	LX(5)	=	12
NZTL	=	11	LX(6)	=	15
JFZ	=	8	LX(7)	=	18
LZ(1)	=	1	LX(8)	=	21
LZ(2)	=	3	LX(9)	=	24
LZ(3)	=	5	LX(10)	=	27
LZ(4)	=	7	LX(11)	=	29
LZ(5)	=	10	LX(12)	=	31

Program output consists of: video messages indicating when specific phases of the computation are complete; printed output listing interpolated nonorthogonal velocity components; and diskfile VELOCITY for use by program EDDY3.

The first output is a printout of input diskfile NAMVEL for inspection to insure no input data errors have been made.

The next output phase consists of the following messages on the video display:

1. "Section Data Smoothed & Differentiated for I=i";
2. "All Section Data Smoothed & Differentiated".

The first message is displayed for each section where i ranges from 1 to NXTL.

The program then provides printed output for all computed velocity components and for $\partial\bar{w}/\partial\bar{z}$ on the keel and waterline.

Finally, diskfile VELOCITY is written for use by program EDDY3.

2.2.6 Program EDDY3

The purpose of this program is to compute three-dimensional boundary-layer development on the ship hull. While all of the other programs listed in the Appendix are shown in their forms appropriate to the TRS-80 Model II microcomputer, the listing of EDDY3 is appropriate to a UNIVAC 1108 computer. Required changes for CDC or IBM computers are confined to SUBROUTINE STORIT and are included via Comment lines.

Input data for program EDDY3 consists of: diskfile GEOMETRY; diskfile VELOCITY; three NAMELIST files, viz, NAME, DATA and STRT; and two lines of card-type data. Diskfiles GEOMETRY and VELOCITY are created by programs SHPMSH and VELOC, respectively. The rest of the input is described below.

The first line of input is read from unit TAPEIN which has a default value of 5. This input is the title of the case to be run which can be up to 54 characters long (format 9A6).

The next segment of the input is also read from unit TAPEIN and is in NAMELIST format. The data are as follows:

NAMELIST/NAME/: This namelist includes primary parameters defining the grid, print logic, etc.

- DETA = Array of $\Delta\eta$ values; only the $\Delta\eta$ nearest the surface need be input. Note that η is the conventional Levy-Lees-type normal coordinate.
- EPS = Convergence criterion for turbulence-model equations (DEFAULT=.01).
- EPST = Convergence criterion for crossflow momentum equation (DEFAULT=.01).
- EPSV = Convergence criterion for streamwise momentum equation (DEFAULT=.01).

ETAE = Boundary-layer edge value for η at initial station (DEFAULT=8).
 ICHORD = Not used (DEFAULT=1).
 IFLOW = Flag indicating crossflow integration direction. IFLOW=1 for integration from keel to waterline and IFLOW=2 for integration from waterline to keel (DEFAULT=1).
 IFPRNT = Print flag. IFPRNT=0 deletes printout of contents of diskfiles GEOMETRY and VELOCITY. Otherwise the contents of these two input files are printed (DEFAULT=0).
 IPX = Intervals at which profiles are printed in streamwise direction, e.g., IPX=10 means print every tenth step.
 IPZ = Intervals at which profiles are printed in girthwise direction.
 ISPAN = Not used (DEFAULT=1).
 ITMAX = Maximum number of iterations allowed (DEFAULT=20).
 NPT = Maximum number of mesh points allowed normal to hull (must not exceed 101... DEFAULT=101).
 NTR = Laminar/turbulent flow flag. To start from laminar profiles use NTR=0. Otherwise, start from turbulent profiles (DEFAULT=1).
 NXSTRT = Index of initial section. (must be 1).
 NXT = Index of final section to which computation proceeds.
 NZSTRT = Index of initial girthwise line (Must be 1).
 NZT = Index of final girthwise line to which computation proceeds.
 TAPEIN = Logical unit number for subsequent NAMELIST input (DEFAULT=5).
 TAPEOT = Logical unit number for printed output (DEFAULT=6).
 TAPEGP = Logical unit number for input diskfile GEOMETRY (DEFAULT=16).
 TAPEPF = Logical unit number for temporary diskfile-this is a very large file (DEFAULT=17).
 TAPEDT = Logical unit number for temporary diskfile (DEFAULT=18).
 TAPEVL = Logical unit number for input diskfile VELOCITY (DEFAULT=19).
 VGP = Geometric progression ratio for normal grid.
 ISHORT = Flag for slightly abbreviated print. ISHORT=0 deletes details of convergence parameters. Otherwise all details are given (DEFAULT=0).

- RFTRB = Relaxation coefficient for turbulence model equations (DEFAULT=.8).
- RFVEL = Relaxation coefficient for momentum equations (DEFAULT=1).

NAMELIST/DATA/: This namelist defines the various turbulence-model closure coefficients.

- SALFA = The closure coefficient α (DEFAULT=1.11111).
- SALFAS = The closure coefficient α^* (DEFAULT=.3).
- SBETA = The closure coefficient β (DEFAULT=.15).
- SBETAS = The closure coefficient β^* (DEFAULT=.09).
- SLAMDA = The closure coefficient λ (DEFAULT=.091).
- SSIGMA = The closure coefficient σ (DEFAULT=.5).
- SSGMAS = The closure coefficient σ^* (DEFAULT=.5).
- USTOP = The value of $u^+ = u/u_{\tau}$ below which turbulent dissipation rate is analytically prescribed (DEFAULT=4).
- ZIOTAE = Dimensionless value of turbulent mixing energy at the boundary-layer edge (DEFAULT= $3.75 \cdot 10^{-5}$).
- ZIOTAL = Ratio of turbulent length scale to boundary layer thickness (DEFAULT=.09).
- RW2 = The closure coefficient R_{ω} (DEFAULT=4.).

NAMELIST/STRT/: This namelist provides input parameters needed to define turbulent starting profiles.

- ALAMI = Array of values at each girthwise station for ratio of turbulent mixing energy to the corresponding flat-plate value.
- CFXI = Array of streamwise skin friction, $C_{f_x} = \tau_{w_x} / (\frac{1}{2}\rho U^2)$, at each girthwise station.
- CFZI = Array of crossflow skin friction, $C_{f_z} = \tau_{w_z} / (\frac{1}{2}\rho U^2)$, at each girthwise station.
- DELTAI = Array of boundary-layer thickness at each girthwise station.
- THETXI = Array of streamwise momentum thickness at each girthwise station.
- THETZI = Array of crossflow momentum thickness at each girthwise station (NOTE: This array must be the same as THETXI in the present form of the program.).

The final input is a line defining the freestream velocity, static pressure and kinematic viscosity. These quantities are dimensional and the format is (3E12.4).

Printed program output is mostly self explanatory and will thus be described only briefly here. The first part of the printout lists all input parameters. Then, on every section, key integral parameters such as c_{f_x} , c_{f_z} , δ_x^* , δ_z^* , θ_x , θ_z are given at all girthwise stations. At the specified IPX and IPZ intervals, velocity and turbulence parameter profiles are printed.

The program also writes a special diskfile (use logical unit number 9) containing computed displacement thickness. This diskfile is used in program INTERP (next Subsection) to allow repeating the computation with displacement effects included.

2.2.7 Program INTERP

The purpose of this program is to interpolate the computed displacement thickness distribution onto the orthogonal rectangular coordinate system used by program SHIP. The whole computational procedure can then be repeated to account for displacement effects.

The program is written in BASIC and uses an extremely efficient MACHINE LANGUAGE interpolation routine referred to as USR1 in the program.

The first two inputs to the program are prompted by the video messages "NXTL=?" and "NZTL=?"; the values of NXTL and NZTL used in running EDDY3 must be entered. The next input is prompted by the message "FILE NAME FOR NONORTHOGONAL MESH ?"; the response is the name given to the print file generated by program SHPMSH, viz, FORTu5/DAT. The option is included to enter the file name in case the user opts to rename the file after running program SHPMSH. The final input is prompted by the messages "IMAX=?" and "JMAX=?"; the values of IMAX and JMAX used in running program PRMESH must be used.

Program output is a revised diskfile MESHPR containing the modified ship-hull coordinates. The original MESHPR is renamed MESHPR/OLD. The contents of the new MESHPR diskfile can be examined using program SPLOT (Subsection 2.2.2). Then, the entire computational sequence is repeated to determine the effect of including displacement thickness in the inviscid-flow computation.

2.3 APPLICATIONS

In this section we exercise the computational package to exhibit accuracy and capabilities for two test cases. We first compute the inviscid flow past a very simple hull investigated by Michell¹. Then, we do both inviscid and viscous computations for the SSPA 720 ship hull³.

2.3.1 Michell's Example

Our first test case is the sinusoidal hull form investigated by Michell. The hull coordinates are defined by

$$\bar{y} = - .02 [1 + \cos (2\pi\bar{x})] [1 + \cos (10\pi\bar{z})] \quad (2)$$

Michell found that for a Froude number based on hull length of 0.25, the wave-drag coefficient is

$$C_R \equiv \frac{R}{\frac{1}{2}\rho U^2 L^2} \doteq 0.60 \cdot 10^{-4} \quad (3)$$

In our computations we use 10 equally-spaced mesh points in the vertical direction. The number of points used in the streamwise direction has been varied to determine the number needed to achieve acceptable engineering accuracy. Again using equally-spaced points, computations have been made for 11, 21 and 41 points. Table 1 summarizes total wave-drag coefficient.

Table 1. Predicted Wave Drag

No. of Streamwise Mesh Points	$10^4 C_R$
11	0.49
21	0.52
41	0.56
Michell	0.60

Figure 3 provides an interesting view of the rate of convergence of the method. The figure shows the local drag coefficient as a function of distance along the hull. Note that the peak values near $2x/L = -0.25$ and $2x/L = +0.25$ are almost an order-of-magnitude larger than C_R .

As a test of the sensitivity to vertical-mesh-point spacing, we have repeated the 11-streamwise-point computation with only 6 points in the vertical direction. The predicted C_R is 1.3% lower than the value listed in Table 1.

Based on results obtained in these computations, accuracy acceptable for preliminary design (10%) can be obtained on a mesh of about 15 streamwise points by 6 vertical points. Such a run takes about 45 minutes on a TRS-80 Model II microcomputer.

2.3.2 SSPA Model 720

Turning now to the SSPA Model 720, we exercise both the inviscid and the viscous segments of the computational package. However, for reasons which become evident below, we do not exercise the displacement-thickness correction phase for this hull.

Figures 4 and 5 show computed streamwise velocity distribution near the free surface and along the keel, respectively. In all cases

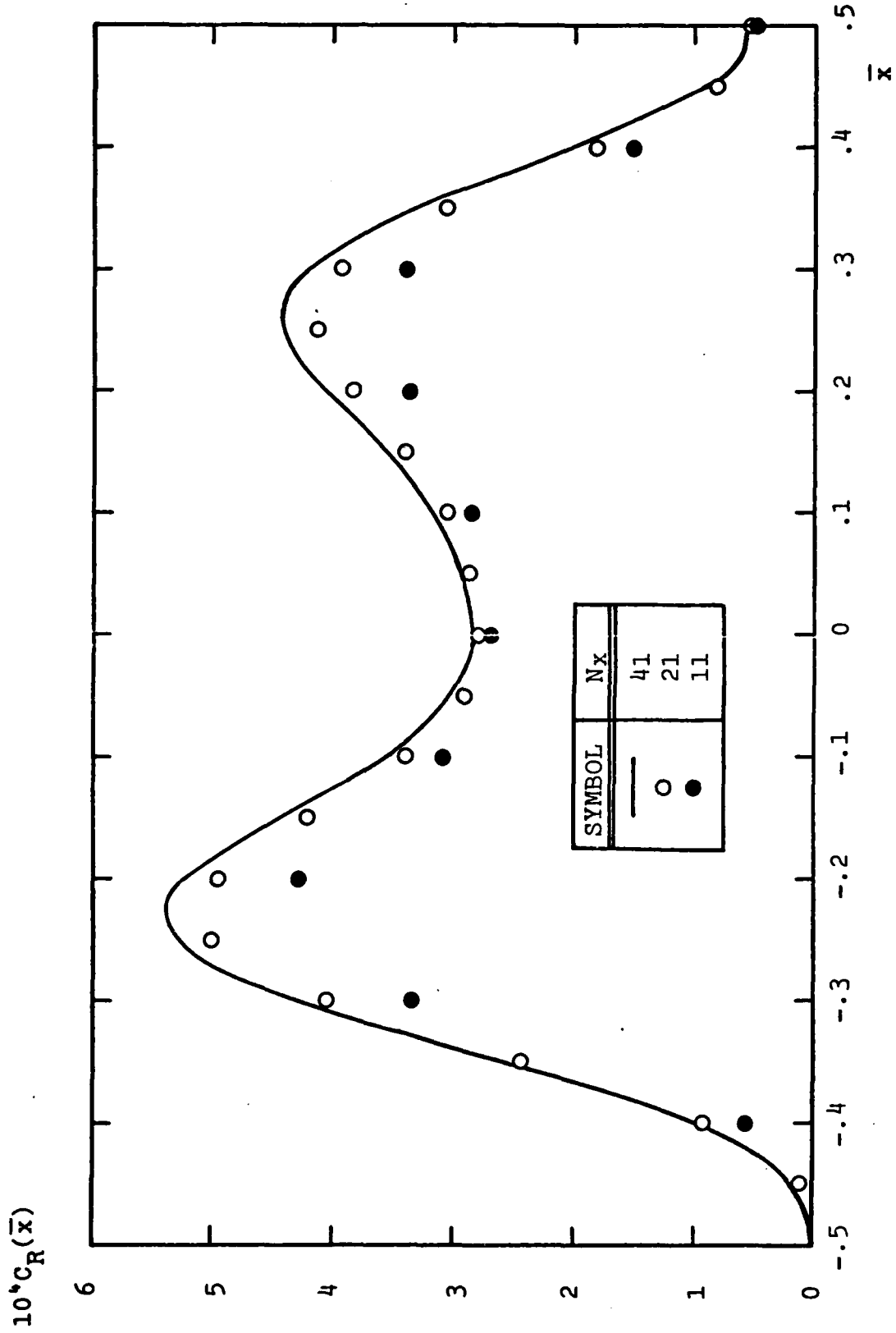


Figure 3. Variation of predicted wave-drag distribution with streamwise mesh-point number, N_x .

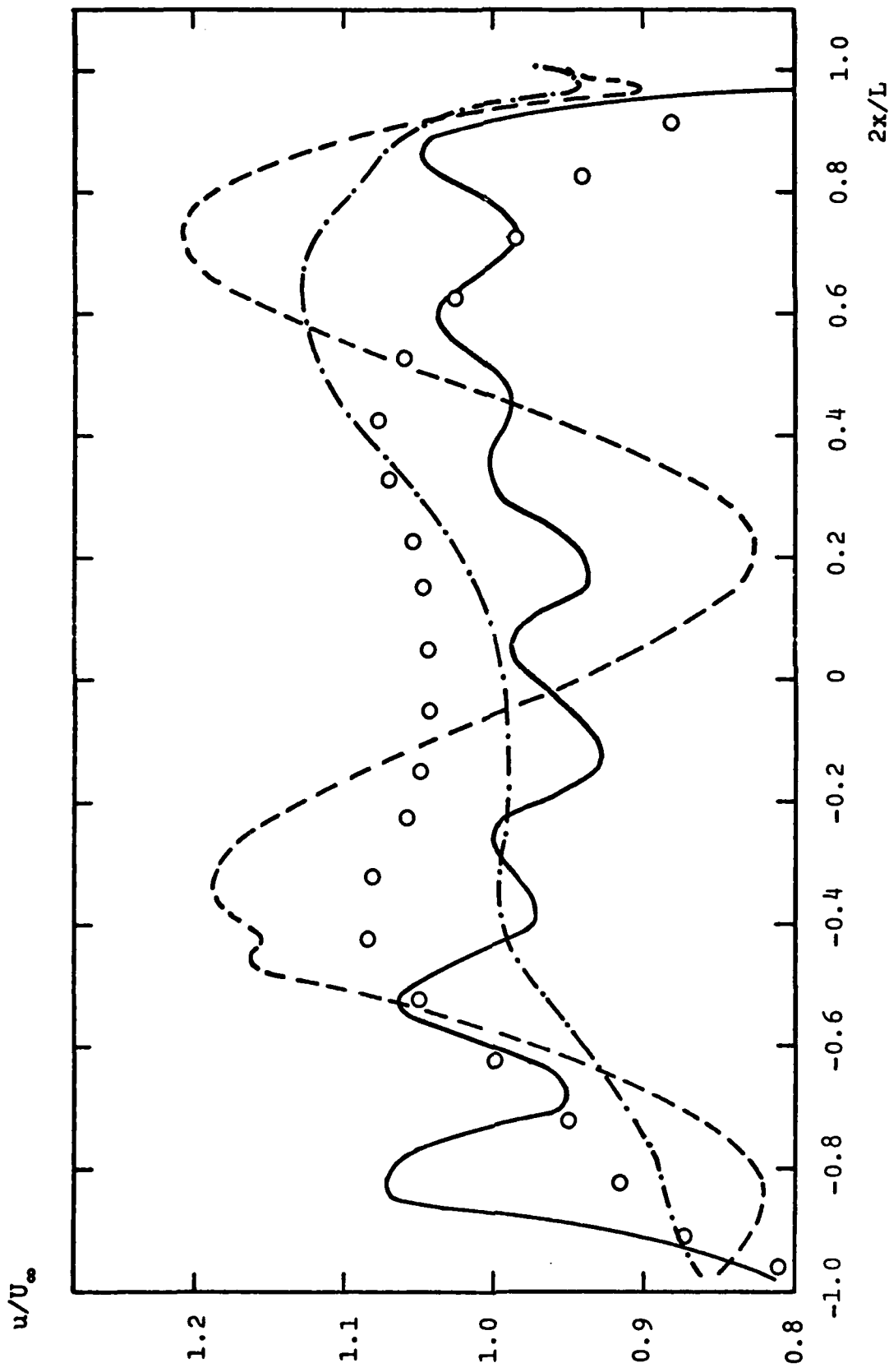


Figure 4. Computed velocity distribution on a line near the waterline ($2z/L=-0.025$):
 —, $Fr=0.15$; - - - -, $Fr=0.30$; - · - · -, $Fr=0.60$; O, Douglas-Neumann.

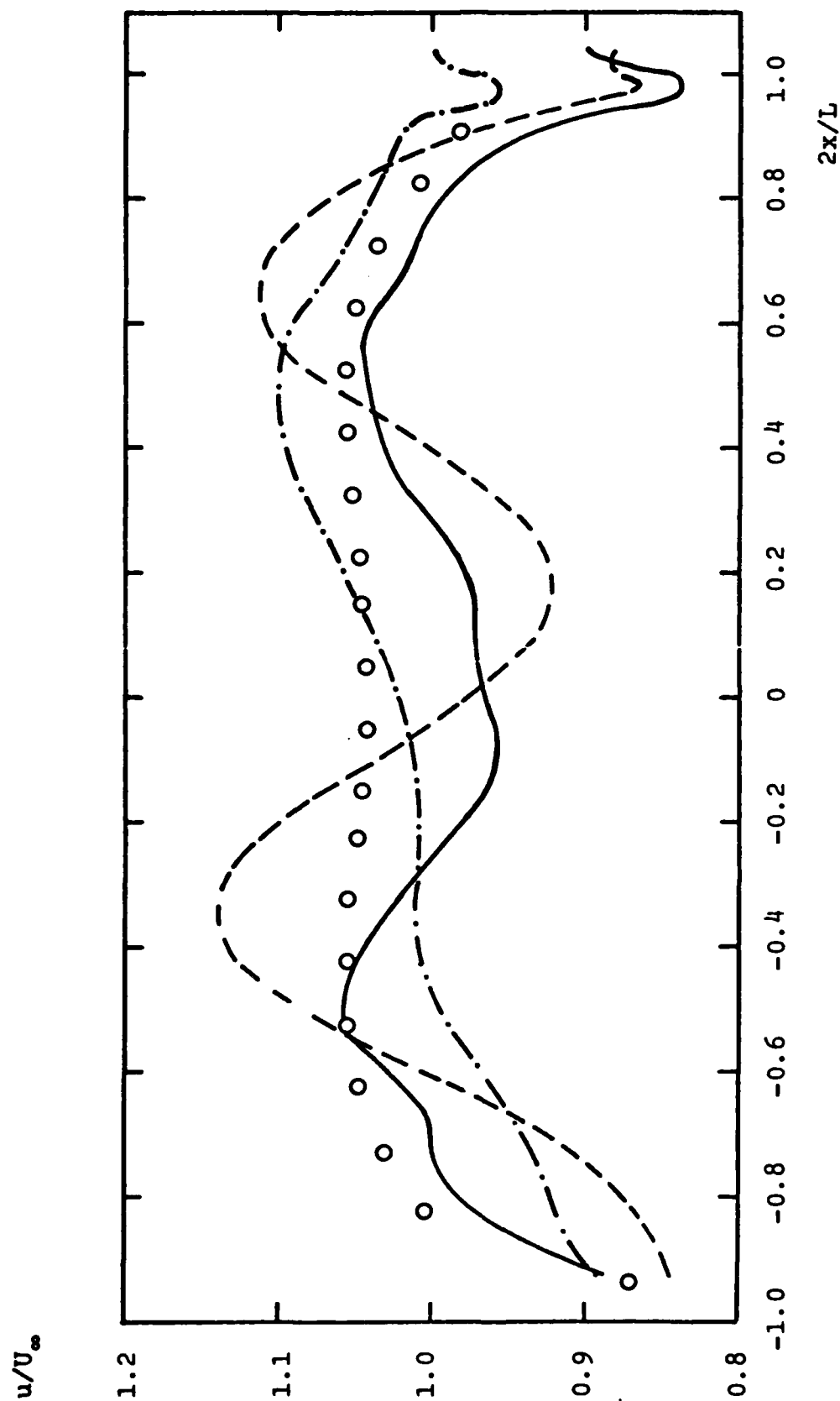


Figure 5. Computed velocity distribution along the keel ($2z/L=0$): —, $Fr=0.15$; - - -, $Fr=0.30$; - · - ·, $Fr=0.60$; O, Douglas-Neumann.

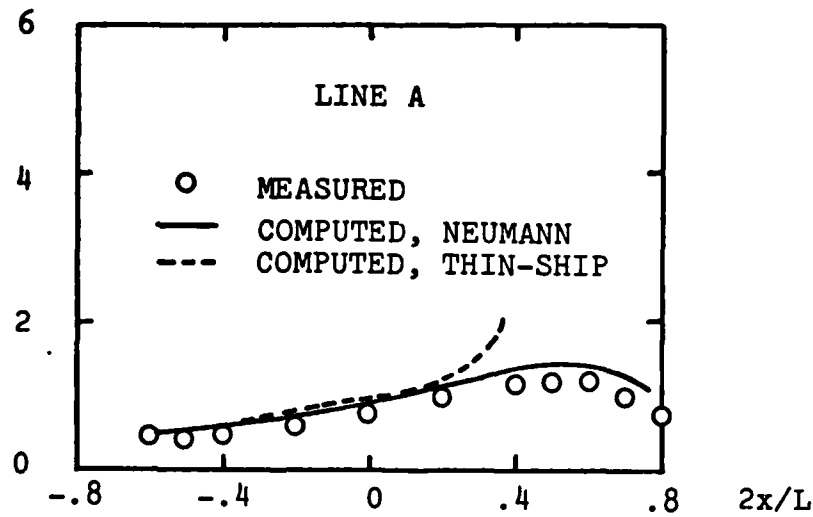
the finite-difference grid consists of 41 points in the stream-wise direction and 11 points girthwise. Each figure shows velocity distributions for Froude numbers based on total hull length of 0.15, 0.30 and 0.60. Also shown in the figures are corresponding velocity distributions obtained from a Douglas-Neumann⁵ "double-hull" computation.

Near the waterline the $Fr = 0.15$ distribution is quite oscillatory, a feature characteristic of thin-ship theory in the limit of small Froude number. In the region between $2x/L = -0.30$ and $2x/L = 0.20$, the $Fr = 0.30$ distribution has a strong adverse gradient which almost certainly would lead to boundary-layer separation. For $Fr = 0.60$, the near-waterline distribution corresponds to flow acceleration over the first 70% of the hull. In contrast to all of the thin-ship-theory predictions, the "double-hull" computation indicates a more-or-less symmetrical distribution with nearly constant velocity over the central 40% of the hull. Figure 5 indicates similar trends along the keel with the exception that no oscillations appear for the $Fr = 0.15$ computation.

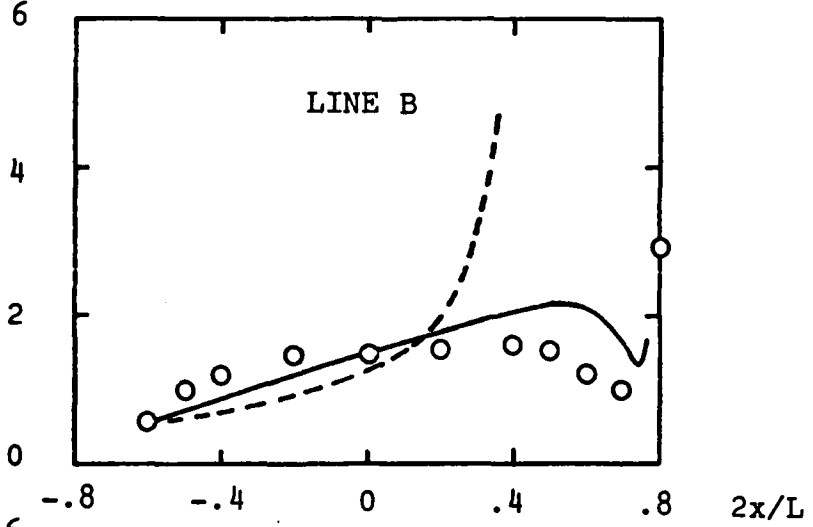
Because the $Fr = 0.60$ distributions appear to be the least likely to cause numerical difficulty in running EDDY3, we have selected $Fr = 0.60$ as most appropriate for the test case (the actual value is not available in our literature on this hull).

Figure 6 compares computed and measured momentum thickness, θ . The figure also shows computed results obtained using Douglas-Neumann velocities. Lines A, B and C lie along the keel, about 15% of the way between keel and waterline, and about midway between keel and waterline, respectively. As shown, the most notable difference is the rapid increase in θ along Lines A and B downstream of $2x/L = 0.20$. We predict boundary-layer separation at $2x/L = 0.36$ over a significant portion of the hull. Although this result at first appears inconsistent with the velocity distributions of Figures 4 and 5, there is no contradiction. The predicted crossflow velocities are quite large ($w/U=0.25$) and along streamlines we indeed predict relatively large adverse gradients approaching $2x/L = 0.36$.

2000 θ/L



2000 θ/L



2000 θ/L

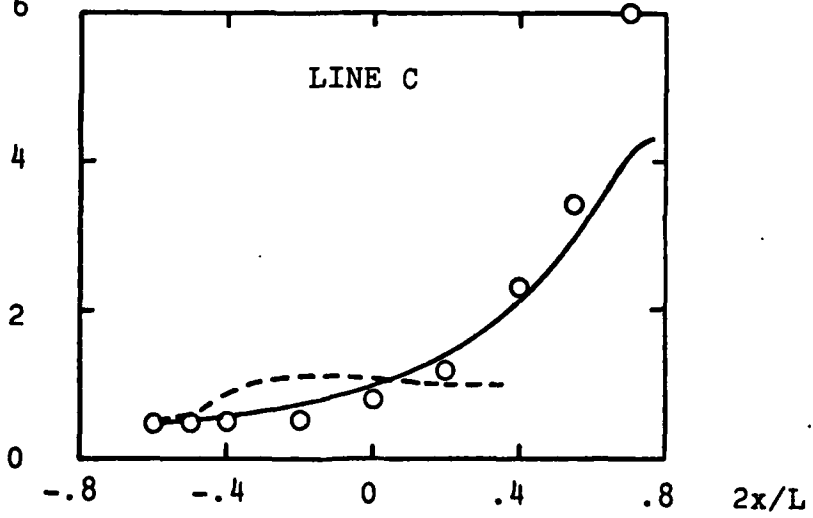


Figure 6. Comparison of computed and measured momentum thickness for the SSPA Model 720.

Because the numerical predictions fail to cover such a large region of the hull, there is little point in reiterating to account for displacement effects at this time.

3. DISCUSSION

The foundation has been developed for a microcomputer-based interactive design procedure for general ship hulls. Typical computing time for all but the boundary-layer computation are of the order of 3 to 4 hours on a TRS-80 Model II microcomputer. Using a 16-bit microprocessor, these computing times would be reduced to less than a half hour. Typically, EDDY3 runs in about 20 minutes on a UNIVAC 1108 which translates to about 3 hours on a 16-bit microcomputer. Thus, the computational package constitutes a practical "numerical towing tank" which can be used to effect and test design modifications in a single workday.

On the one hand, the weakest link in the computational package is program SHIP which determines the inviscid velocities. Based on classical thin-ship theory, its limitations are well known. On the other hand, program EDDY3 embodies one of the most tested and validated turbulence models available. The model has even been shown to accurately predict properties of "thick" boundary layers typical of those near the stern of a ship hull⁶.

Research in the immediate future should focus upon improving the ability to predict the inviscid velocities. The formulation of Noblesse and Dagan⁷, which amount to a coordinate transformation using the thin-ship velocity potential, should be incorporated in program SHIP. Also, normal pressure gradient in the boundary-layer computation should be accounted for to provide accurate predictions when the boundary layer becomes "thick". These two steps alone would yield a computational package which could be immediately applied by serious hull designers. To further aid the designer, a master program should be devised to coordinate input data preparation for the various program modules. Additionally, available graphics packages should be adapted to the computational procedure to permit more-detailed display and examination of calculated mesh and flow details.

Long range research topics should include: (a) numerical studies to aid development of a more-accurate integral-method treatment of the boundary layers, and (b) development of an approximate engineering-oriented procedure for treating hulls on which the boundary layer separates. The modular structure of the computational package developed in this project is readily adaptable to such advanced developments.

REFERENCES

1. Michell, J.H., "The Wave-Resistance of a Ship", Proc Roy Soc, Phil Mag (5), Vol 45, pp 106-123 (1898).
2. Wilcox, D.C. and Rubesin, M.W., "Progress in Turbulence Modeling for Complex Flow Fields Including Effects of Compressibility," NASA TP 1517 (1980).
3. Larsson, L., "SSPA-ITTC Workshop on Ship Boundary Layers" (1980).
4. Wilcox, D.C., "User's Manual for PROGRAM SMOOTH: A Least-Squares Cubic-Spline Fit Routine," DCW Industries Report DCW-R-16-02 (1976).
5. Hess, J.L. and Smith, A.M.O., "Calculation of Potential Flow About Arbitrary Bodies," Prog Aero Sci, Vol 8, Pergamon Press, Oxford and New York (1966).
6. Wilcox, D.C., "Boundary-Layer Development on Ship Hulls," DCW Industries Report DCW-R-26-01 (1983).
7. Noblesse, F. and Dagan, G., "Nonlinear Ship Wave Theories by Continuous Mapping," Institute of Hydraulic Research, Univ of Iowa (1975).

APPENDIX: PROGRAM LISTINGS

This Appendix contains listings of the seven program modules which constitute the computational package. In the listings, the number zero appears with a slash to distinguish it from the letter "O". The various programs are located on the following pages:

<u>PROGRAM NAME</u>	<u>PAGES</u>
PRMESH.....	34-35
SPLOT.....	36-38
SHIP.....	39-45
SHPMESH.....	46-55
VELOC.....	56-62
EDDY3.....	63-96
INTERF.....	97


```

00100      PROGRAM PRMESH
00200 C *****
00300 C CREATE RECTANGULAR MESH FOR USE IN PROGRAM SHIP
00400 C *****
00500      DIMENSION XB(50),Y(41,20),YB(50,20),YI(50,20),ZB(20)
00600      COMMON/DATA/ X(41),Z(20)
00700      COMMON/NDATA/ IMAX,JMAX,NXTL,NZTL
00800      CALL NAMIN
00900      NZTLM=NZTL-1
01000      DO 10 I=1,NXTL
01100      READ(9,100) XB(I),YB(I,NZTL),ZB(NZTL),YB(I,NZTLM),ZB(NZTLM)
01200      DO 1 J=3,NZTLM,2
01300      JJ=NZTL-J+1
01400      JM=J-1
01500 1      READ(9,110) YB(I,JJ),ZB(JJ),YB(I,JM),ZB(JM)
01600      JN=1
01700      JS=2
01800      ZN=ZB(1)
01900      ZS=ZB(2)
02000      DO 5 J=1,JMAX
02100 2      IF(Z(J).LT.ZS.OR.ZN.EQ.ZS) GO TO 3
02200      YI(I,J)=YB(I,JN)+(YB(I,JS)-YB(I,JN))*(Z(J)-ZN)/(ZS-ZN)
02300      GO TO 5
02400 3      JN=JN+1
02500      IF(JN.GE.NZTL) GO TO 4
02600      JS=JS+1
02700      ZN=ZB(JN)
02800      ZS=ZB(JS)
02900      GO TO 2
03000 4      YI(I,J)=0.
03100 5      CONTINUE
03200 10     CONTINUE
03300      IW=1
03400      IE=2
03500      XW=XB(1)
03600      XE=XB(2)
03700      DO 15 I=1,IMAX
03800 11     IF(X(I).GT.XE.OR.XW.EQ.XE) GO TO 13
03900      DO 12 J=1,JMAX
04000 12     Y(I,J)=YI(IW,J)+(YI(IE,J)-YI(IW,J))*(X(I)-XW)/(XE-XW)
04100      GO TO 15
04200 13     IW=IW+1
04300      IE=IW+1
04400      XW=XB(IW)
04500      XE=XB(IE)
04600      GO TO 11
04700 15     CONTINUE
04800      DO 20 I=1,IMAX
04900 20     WRITE(8,200) X(I)
05000      DO 30 J=1,JMAX
05100 30     WRITE(8,200) Z(J)
05200      DO 40 I=1,IMAX
05300      DO 40 J=1,JMAX
05400 40     WRITE(8,200) Y(I,J)
05500 100    FORMAT(3F10.5,11X,2F10.5)
05600 110    FORMAT(10X,2F10.5,11X,2F10.5)
05700 200    FORMAT(1PE13.6)
05800      END
05900      SUBROUTINE NAMIN
06000 C *****

```

```

06100 C READ INPUT DATA FILE
06200 C *****
06300 C DESCRIPTION OF 'PRMESHIN' INPUT DATA:
06400 C IMAX...NUMBER OF POINTS IN STREAMWISE DIRECTION: OUTPUT
06500 C JMAX...NUMBER OF POINTS IN VERTICAL DIRECTION: OUTPUT
06600 C NXTL...NUMBER OF POINTS IN STREAMWISE DIRECTION: INPUT
06700 C NZTL...NUMBER OF POINTS IN VERTICAL DIRECTION: INPUT
06800 C X(I)...SPECIFIED STREAMWISE COORDINATE ARRAY: OUTPUT
06900 C Z(J)...SPECIFIED VERTICAL COORDINATE ARRAY: OUTPUT
07000 C *****
07100 COMMON/DATA/ X(41),Z(20)
07200 COMMON/NDATA/ N(4)
07300 CALL OPEN(7,'PRMESHIN ',1)
07400 CALL OPEN(8,'MESHPR ',1)
07500 CALL OPEN(9,'MESH ',1)
07600 DO 1 I=1,4
07700 N(I) = 0
07800 1 CALL NAMLST(7,2,N(I),DUMMY)
07900 IMAX=N(1)
08000 JMAX=N(2)
08100 DO 2 I=1,IMAX
08200 2 CALL NAMLST(7,2,1,X(I))
08300 DO 3 J=1,JMAX
08400 3 CALL NAMLST(7,2,1,Z(J))
08500 RETURN
08600 END
08700 SUBROUTINE NAMLST(II,IO,N,X)
08800 C *****
08900 C READ VARIABLE NAME AND VALUE
09000 C *****
09100 DIMENSION A(3)
09200 IF(N.NE.0) GO TO 1
09300 READ(II,2) A(1),A(2),A(3),N
09400 WRITE(IO,2) A(1),A(2),A(3),N
09500 RETURN
09600 1 READ(II,3) A(1),A(2),A(3),X
09700 WRITE(IO,3) A(1),A(2),A(3),X
09800 RETURN
09900 2 FORMAT(1X,3A4,I4)
10000 3 FORMAT(1X,3A4,1PE13.6)
10100 END

```

```

10 REM *****
20 REM ***** PLOTTING ROUTINE FOR SHIP SECTIONS *****
30 REM ***** DISK FILE 'MESHPR' CREATED BY *****
40 REM ***** PROGRAM PRMESH *****
50 REM *****
60 DIM X(41),Y(20),Z(20)
70 NZ=70
80 NY=20
90 NL=9
100 IC=0
110 OPEN "I", 1, "MESHPR"
120 INPUT "IMAX ="; IM
130 INPUT "JMAX ="; JM
140 FOR I=1 TO IM
150 INPUT#1, X(I)
160 NEXT I
170 FOR J=1 TO JM
180 INPUT#1, Z(J)
190 NEXT J
200 IC=IC+1
210 IF IC>IM THEN 1600
220 FOR J=1 TO JM
230 INPUT#1, Y(J)
240 Y(J)=-Y(J)
250 NEXT J
260 GOSUB 620
270 CLS
280 PRINT "I ="; IC; " X(I) ="; X(IC)
290 PRINT:PRINT
300 PRINT "ZMIN = "; ZL; " ZMAX = "; ZU
310 PRINT
320 PRINT "YMIN = "; YL; " YMAX = "; YU
330 PRINT:PRINT
340 PRINT "SPECIFY PLOTTING RANGE"
350 PRINT
360 IF IC=1 THEN 430
370 PRINT "SAME RANGE ?"
380 Q$=INKEY$
390 IF Q$="" THEN 380
400 IF LEFT$(Q$,1)<>"Y" THEN 430
410 GOSUB 1550
420 GOTO 510
430 INPUT "ZMIN ="; ZL
440 INPUT "ZMAX ="; ZU
450 INPUT "YMIN ="; YL
460 INPUT "YMAX ="; YU
470 GOSUB 1490
480 PRINT
490 INPUT "ANY ERRORS"; Q$
500 IF LEFT$(Q$,1)="Y" THEN 340
510 CLS
520 GOSUB 770
530 GOSUB 1060
540 GOSUB 1130
550 A$=INKEY$
560 IF A$="" THEN 550
570 IF A$="C" THEN 260
580 IF A$="N" THEN 200
590 END

```

```

600 REM ***** FIND MIN & MAX VALUES *****
610 REM --- MIN VALUE ---
620 ZL=1.E20
630 YL=1.E20
640 FOR I=1 TO JM
650 IF Z(I)<ZL THEN ZL=Z(I)
660 IF Y(I)<YL THEN YL=Y(I)
670 NEXT I
680 REM --- MAX VALUE ---
690 ZU=ZL
700 YU=YL
710 FOR I=1 TO JM
720 IF Z(I)>ZU THEN ZU=Z(I)
730 IF Y(I)>YU THEN YU=Y(I)
740 NEXT I
750 RETURN
760 REM ***** DRAW AXES AND TIC MARKS *****
770 ZZ=ZU-ZL
780 YY=YU-YL
790 REM --- Y-AXIS TIC MARKS ---
800 IY%=1
810 FOR I=1 TO 5
820 PRINT@ (IY%,NL-2), CHR$(95);
830 PRINT@ (IY%,NL-1), CHR$(95)
840 IY%=IY%+5
850 NEXT I
860 REM --- DRAW THE Y-AXIS ---
870 IY%=2
880 II=NY+1
890 FOR I=IY% TO II
900 PRINT@ (I,NL), CHR$(156)
910 NEXT I
920 REM --- Z-AXIS TIC MARKS ---
930 II=NZ+NL
940 IS=NZ/5
950 FOR I=NL TO II STEP IS
960 PRINT@ (NY+2,I), CHR$(156);
970 NEXT I
980 REM --- DRAW THE Z-AXIS ---
990 II=NZ+NL-1
1000 NN=NL+1
1010 FOR I=NN TO II
1020 PRINT@ (NY+1,I), CHR$(95)
1030 NEXT I
1040 RETURN
1050 REM ***** LABEL AXES *****
1060 PRINT@ (0,0), YU;
1070 PRINT@ (NY+1,0), YL;
1080 PRINT@ (NY+3,NL-2), ZL;
1090 PRINT@ (NY+3,NZ+NL-6), ZU;
1100 PRINT@ (0,NZ/2+NL-4), "X ="; X(IC);
1110 RETURN
1120 REM ***** PLOT THE POINTS *****
1130 ID=0
1140 FOR I=1 TO JM
1150 C=NL+INT(NZ*(Z(I)-ZL)/ZZ+.5)
1160 R=NY+1-INT(NY*(Y(I)-YL)/YY)
1170 IF C>(NZ+NL) THEN 1270
1180 IF C<NL THEN 1270

```

```

1190 IF R>(NY+1) THEN 1270
1200 IF R<1 THEN 1270
1210 GOSUB 1360
1220 IF GC=160 THEN 1240
1230 PRINT@ (R,C), CHR$(GC);
1240 PRINT@ (R+1,C), CHR$(26)+CHR$(GC);
1250 PRINT CHR$(25);
1260 GOTO 1280
1270 ID=ID+1
1280 NEXT I
1290 PRINT@ (7,1), ID;
1300 PRINT@ (8,0), "POINTS";
1310 PRINT@ (9,1), "OFF";
1320 PRINT@ (10,0), "SCALE";
1330 PRINT CHR$(02);
1340 RETURN
1350 REM ***** INTERPOLATE WITHIN BOX *****
1360 RR=R-(NY+1-NY*(Y(I)-YL)/YY)
1370 GC=154
1380 IF RR<.2 THEN RETURN
1390 GC=155
1400 IF RR<.4 THEN RETURN
1410 R=R-1
1420 GC=160
1430 IF RR<.6 THEN RETURN
1440 GC=152
1450 IF RR<.8 THEN RETURN
1460 GC=153
1470 RETURN
1480 REM ***** SAVE PLOTTING RANGES *****
1490 XL=ZL
1500 XU=ZU
1510 WL=YL
1520 WU=YU
1530 RETURN
1540 REM ***** RETRIEVE PLOTTING RANGES *****
1550 ZL=XL
1560 ZU=XU
1570 YL=WL
1580 YU=WU
1590 RETURN
1600 END

```

```

00100      PROGRAM SHIP
00200 C *****
00300 C THIS PROGRAM USES LINEAR SHIP THEORY TO COMPUTE THE WAVE DRAG
00400 C AND SURFACE VELOCITY COMPONENTS FOR AN ARBITRARY VESSEL.
00500 C *****
00600      COMMON/COORD/ XB(41),YB(41,10),ZB(10),FB(40,9)
00700      COMMON/DATA/  FR,IMAX,JMAX,MINT2,MINT3,NZOT,IMAXM,JMAXM
00800      COMMON/QUADRA/AM(80),RSAV(80),TSAV(80),WM
00900      COMMON/VELO/  U(40,9),W(40,9)
01000      CALL NAMIN
01100      FR2R=1./FR**2
01200      DO 1 I=1,IMAX
01300      READ(7,100) XB(I)
01400 1      XB(I)=FR2R*XB(I)
01500      DO 2 J=1,JMAX
01600      READ(7,100) ZB(J)
01700 2      ZB(J)=FR2R*ZB(J)
01800      DO 3 I=1,IMAX
01900      DO 3 J=1,JMAX
02000      READ(7,100) YB(I,J)
02100 3      YB(I,J)=FR2R*YB(I,J)
02200      CALL GEOM(IMAX,JMAX)
02300      RWAVE=0.
02400      DO 4 I=1,IMAXM
02500      DO 4 J=1,JMAXM
02600      U(I,J)=0.
02700 4      W(I,J)=0.
02800      CALL VELOC1
02900      WRITE(1,510)
03000      CALL VELOC2
03100      WRITE(1,520)
03200      CALL VELOC3
03300      WRITE(1,530)
03400      DO 10 I=1,IMAXM
03500      CALL DRAG(DRWAVE,I)
03600      RWAVE=RWAVE+DRWAVE
03700      IF(NZOT.EQ.0) CALL OUTPUT(I)
03800      WRITE(2,400) DRWAVE,RWAVE
03900 10      CONTINUE
04000      WRITE(2,200) RWAVE
04100      WRITE(1,200) RWAVE
04200 100     FORMAT(E13.6)
04300 200     FORMAT(////////,1X18HCOMPUTED WAVE DRAG///4X7HRWAVE =,1PE12.4)
04400 400     FORMAT(//,1X,'INCREMENTAL WAVE DRAG ON THIS SECTION IS ',1PE13.4/
04500 *           ,1X,'TOTAL DRAG COMPUTED UP TO THIS SECTION IS',E13.4)
04600 510     FORMAT(1X,'VELOC1 COMPUTATIONS COMPLETE')
04700 520     FORMAT(1X,'VELOC2 COMPUTATIONS COMPLETE')
04800 530     FORMAT(1X,'VELOC3 COMPUTATIONS COMPLETE')
04900      END
05000      FUNCTION ACOS(X)
05100 C *****
05200 C THIS SUBROUTINE COMPUTES THE ARC COSINE
05300 C *****
05400      DATA PH/1.5707963268/
05500      ACOS=PH-ASIN(X)
05600      RETURN
05700      END
05800      FUNCTION ASIN(X)
05900 C *****
06000 C THIS SUBROUTINE COMPUTES THE ARC SINE

```

```

06100 C *****
06200 DATA A0,A1,A2/1.5707963050,-.2145988016,.0889789874/
06300 DATA A3,A4,A5/-.0501743046,.0308918810,-.0170881256/
06400 DATA A6,A7,PH/.0066700901,-.0012624911,1.5707963268/
06500 Y=ABS(X)
06600 ASIN=PH-SQRT(1.-Y)*(A0+Y*(A1+Y*(A2+Y*(A3+Y*(A4+Y*(A5
06700 * +Y*(A6+Y*A7))))))
06800 IF(X.LT.0.) ASIN=-ASIN
06900 RETURN
07000 END
07100 SUBROUTINE DRAG(DRWAVE,I)
07200 C *****
07300 C THIS SUBROUTINE COMPUTES THE WAVE DRAG
07400 C *****
07500 COMMON/COORD/ XB(41),YB(41,10),ZB(10),FB(40,9)
07600 COMMON/DATA/ FR,IMAX,JMAX,MINT2,MINT3,NZOT,IMAXM,JMAXM
07700 COMMON/VELO/ U(40,9),W(40,9)
07800 DRWAVE=0.
07900 DX=XB(I+1)-XB(I)
08000 DO 10 J=2,JMAX
08100 DZ=ZB(J-1)-ZB(J)
08200 10 DRWAVE=DRWAVE+U(I,J-1)*FB(I,J-1)*DX*DZ
08300 DRWAVE=4.*DRWAVE*FR**4
08400 RETURN
08500 END
08600 SUBROUTINE GEOM(IMAX,JMAX)
08700 C *****
08800 C THIS SUBROUTINE COMPUTES CELL SLOPE
08900 C *****
09000 COMMON/COORD/ XB(41),YB(41,10),ZB(10),FB(40,9)
09100 DO 10 I=2,IMAX
09200 DX=XB(I)-XB(I-1)
09300 DO 10 J=2,JMAX
09400 10 FB(I-1,J-1)=.5*(YB(I,J)+YB(I,J-1)-YB(I-1,J)-YB(I-1,J-1))/DX
09500 RETURN
09600 END
09700 SUBROUTINE INTK0(G,Z,RMU)
09800 C *****
09900 C THIS SUBROUTINE APPROXIMATES THE BESSEL FUNCTION INTEGRAL WITH A
10000 C GROUP OF CLOSED FORM ANALYTICAL FUNCTIONS.
10100 C *****
10200 ZMU=Z*RMU
10300 IF(ZMU.LE.0) GO TO 20
10400 IF(ZMU.LE..5) G=ZMU*(1.-ALOG(ZMU))
10500 IF(ZMU.GT..5) G=1.2*SQRT(ZMU)
10600 ZMU2=ZMU*RMU
10700 IF(Z.LT.3.) GO TO 40
10800 XI=3./Z
10900 GMAX=1.+0.571*(1.-XI)*(.37+.63*EXP(-10.*XI))
11000 IF(GMAX.LT.G) GO TO 10
11100 RETURN
11200 10 G=GMAX
11300 IF(Z.GE.10.) GO TO 80
11400 GO TO 60
11500 C ----- MU*Z IS ZERO -----
11600 20 G=0.
11700 RETURN
11800 C ----- Z BETWEEN 0 AND 3 -----
11900 40 RMUT=1.+1.114*Z**(-.6666667)
12000 IF(RMU.GE.RMUT) GO TO 55

```

```

12100      IF(Z.GE.1) GO TO 45
12200      ZP25=Z**.25
12300      GM=.83*ZP25
12400      RN=.62+1.68*ZP25
12500      GO TO 50
12600  45   GM=.83+(Z-1.)*(.12-.02*(Z-1.))
12700      RN=2.+3*Z
12800  50   G2=GM/(1.+25*ZMU2**RN)
12900      GO TO 100
13000  55   G=SQRT(1.5707963/ZMU)*EXP(-ZMU)/(RMU-1.)
13100      RETURN
13200  C  ----- Z BETWEEN 3 AND 10 -----
13300  60   RMUT=1.+1.114*Z**(-.6666667)
13400      IF(RMU.GE.RMUT) GO TO 65
13500      XI=.1*(Z-3.)
13600      A=1.38+XI*(.76-.54*XI)
13700      RM=.68+XI*(.53-.41*XI)
13800      G2=A*EXP(-RM*ZMU2)
13900      GO TO 100
14000  65   G=SQRT(1.5707963/ZMU)*EXP(-ZMU)/(RMU-1.)
14100      RETURN
14200  C  ----- Z GREATER THAN 10 -----
14300  80   G2=0.
14400      IF(RMU.LT.1.) G2=(1.571+ASIN(RMU))*EXP(-ZMU2)
14500      *      /SQRT(1.-RMU*RMU)
14600  100  IF(G2.LT.G) G=G2
14700      RETURN
14800      END
14900      SUBROUTINE NAMIN
15000  C  *****
15100  C  READ INPUT DATA FILE
15200  C  *****
15300  C  DESCRIPTION OF 'SHIPIN' INPUT DATA:
15400  C      FR....FROUDE NUMBER = U/SQRT(G*L)
15500  C      IMAX...NUMBER OF MESH POINTS IN STREAMWISE DIRECTION
15600  C      JMAX...NUMBER OF MESH POINTS IN VERTICAL DIRECTION
15700  C      MINT2..NUMBER OF POINTS IN QUADRATURE...INFINITE INTEGRAL
15800  C      MINT3..NUMBER OF POINTS IN QUADRATURE....FINITE INTEGRAL
15900  C      NZOT...0 TO PRINT CELL GEOMETRY AND VELOCITY COMPONENTS
16000  C      ...OTHERWISE PRINT ONLY THE WAVE DRAG
16100  C  *****
16200      COMMON/DATA/ FR,N(5),IMAXM,JMAXM
16300      CALL OPEN(7,'MESHPR ',1)
16400      CALL OPEN(8,'SHIPIN ',1)
16500      CALL OPEN(9,'UEWE ',1)
16600      CALL NAMLST(8,2,1,FR)
16700      DO 1 I=1,5
16800      N(I) = 0
16900  1     CALL NAMLST(8,2,N(I),DUMMY)
17000      IMAXM=N(1)-1
17100      JMAXM=N(2)-1
17200      RETURN
17300      END
17400      SUBROUTINE NAMLST(II,IO,N,X)
17500  C  *****
17600  C  READ VARIABLE NAME AND VALUE
17700  C  *****
17800      DIMENSION A(3)
17900      IF(N.NE.0) GO TO 1
18000      READ(II,2) A(1),A(2),A(3),N

```



```

18100      WRITE(I0,2) A(1),A(2),A(3),N
18200      RETURN
18300  1    READ(II,3) A(1),A(2),A(3),X
18400      WRITE(I0,3) A(1),A(2),A(3),X
18500      RETURN
18600  2    FORMAT(1X,3A4,I4)
18700  3    FORMAT(1X,3A4,1PE13.6)
18800      END
18900      SUBROUTINE OUTPUT(I)
19000 C *****
19100 C THIS SUBROUTINE PRINTS SURFACE VELOCITY COMPONENTS AND CREATES DISK
19200 C FILE 'UEWE' FOR PROGRAM VELOC
19300 C *****
19400      COMMON/COORD/ XB(41),YB(41,10),ZB(10),FB(40,9)
19500      COMMON/DATA/  FR,IMAX,JMAX,MINT2,MINT3,NZOT,IMAXM,JMAXM
19600      COMMON/VELO/  U(40,9),W(40,9)
19700      FR2=FR*FR
19800      XI=.5*FR2*(XB(I)+XB(I+1))
19900      WRITE(2,100) XI
20000      DO 1 J=1,JMAXM
20100      JJ=JMAX-J
20200      ZETA=.5*FR2*(ZB(JJ)+ZB(JJ+1))
20300      ETA=.5*FR2*(YB(I,JJ)+YB(I,JJ+1))
20400      UP=1.+U(I,JJ)
20500      VP=FB(I,JJ)
20600      WP=W(I,JJ)
20700      WRITE(2,200) J,ZETA,UP,VP,WP
20800      Q=SQRT(UP*UP+VP*VP+WP*WP)
20900      UP=UP/Q
21000      VP=VP/Q
21100      WP=WP/Q
21200  1    WRITE(9,1000) XI,ETA,ZETA,UP,VP,WP,Q
21300  100  FORMAT(/,1X38HCOMPUTED VELOCITY COMPONENTS ALONG X =,1PE12.4//
21400      * 4X1HJ,7X4HZ(J),11X4HU(J),11X4HV(J),11X4HW(J)/)
21500  200  FORMAT(I5,1P4E15.4)
21600  1000 FORMAT(3F10.5,4F10.6)
21700      RETURN
21800      END
21900      SUBROUTINE QUAD(M)
22000 C *****
22100 C THIS SUBROUTINE COMPUTES QUADRATURE WEIGHTS AND ABSCISSAS
22200 C *****
22300      COMMON/DATA/  FR,IMAX,JMAX,MINT2,MINT3,NZOT,IMAXM,JMAXM
22400      COMMON/QUADRA/AM(80),RSAV(80),TSAV(80),WM
22500      DATA PI/3.14159265/,CON/.2026423/
22600 C EQUAL WEIGHT CHEBYSHEV INTEGRATION FOR DU3 AND DW3
22700      IF(M.EQ.2) GO TO 15
22800      COEF=PI/(4.*FLOAT(MINT3))
22900      WM=2.*CON*COEF
23000      DO 10 I=1,MINT3
23100      RI=FLOAT(I)
23200      AM(I)=COS((2.*RI-1.)*COEF)
23300      RSAV(I)=SQRT(1.-AM(I)**2)
23400  10    TSAV(I)=ACOS(AM(I))+PI
23500      RETURN
23600 C EQUAL WEIGHT CHEBYSHEV INTEGRATION FOR DU2 AND DW2
23700  15    COEF=PI/(4.*FLOAT(MINT2))
23800      WM=2.*CON*COEF
23900      DO 20 I=1,MINT2
24000      RI=FLOAT(I)

```

```

24100      AM(I)=COS((2.*RI-1.)*COEF)
24200      EM2=AM(I)**2
24300      RSAV(I)=SQRT(1.-EM2)/AM(I)
24400      ARGLOG=1./AM(I)-RSAV(I)
24500  20   TSAV(I)=ALOG(ARGLOG)
24600      RETURN
24700      END
24800      SUBROUTINE VELOC1
24900  C *****
25000  C THIS SUBROUTINE COMPUTES THE CONTRIBUTION OF THE SOURCE-LIKE TERMS
25100  C TO THE SURFACE VELOCITY COMPONENTS
25200  C *****
25300      COMMON/COORD/ XB(41),YB(41,10),ZB(10),FB(40,9)
25400      COMMON/DATA/  FR,IMAX,JMAX,MINT2,MINT3,NZOT,IMAXM,JMAXM
25500      COMMON/VELO/  U(40,9),W(40,9)
25600      DATA PI/3.14159265/
25700  C ----- INITIALIZATION -----
25800      DO 40 I=1,IMAXM
25900      XI=.5*(XB(I)+XB(I+1))
26000      DO 40 J=1,JMAXM
26100      ZETA=.5*(ZB(J)+ZB(J+1))
26200      UINT=0.
26300      WINT=0.
26400      DO 30 II=1,IMAXM
26500      DO 30 JJ=1,JMAXM
26600      XP=XB(II)-XI
26700      XM=XB(II+1)-XI
26800      ZP=-(ZETA+ZB(JJ+1))
26900      ZM=-(ZETA+ZB(JJ))
27000      Z1P=ZB(JJ+1)-ZETA
27100      Z1M=ZB(JJ)-ZETA
27200      RA=SQRT(XP*XP+ZP*ZP)
27300      RB=SQRT(XM*XM+ZP*ZP)
27400      RC=SQRT(XM*XM+ZM*ZM)
27500      RD=SQRT(XP*XP+ZM*ZM)
27600      R1A=SQRT(XP*XP+Z1P*Z1P)
27700      R1B=SQRT(XM*XM+Z1P*Z1P)
27800      R1C=SQRT(XM*XM+Z1M*Z1M)
27900      R1D=SQRT(XP*XP+Z1M*Z1M)
28000      UINT=UINT+ALOG((R1A-Z1P)*(R1C-Z1M)*(RD+ZM)*(RB+ZP)
28100      * /((R1B-Z1P)*(R1D-Z1M)*(RC+ZM)*(RA+ZP)))*FB(II,JJ)
28200      WINT=WINT+ALOG((R1A-XP)*(R1C-XM)*(RD-XP)*(RB-XM)
28300      * /((R1D-XP)*(R1B-XM)*(RC-XM)*(RA-XP)))*FB(II,JJ)
28400  30   CONTINUE
28500      U(I,J)=U(I,J)+.5*UINT/PI
28600      W(I,J)=W(I,J)+.5*WINT/PI
28700  40   CONTINUE
28800      RETURN
28900      END
29000      SUBROUTINE VELOC2
29100  C *****
29200  C THIS SUBROUTINE COMPUTES THE CONTRIBUTION OF THE INFINITE INTEGRAL
29300  C TO THE SURFACE VELOCITY COMPONENTS
29400  C *****
29500      COMMON/COORD/ XB(41),YB(41,10),ZB(10),FB(40,9)
29600      COMMON/DATA/  FR,IMAX,JMAX,MINT2,MINT3,NZOT,IMAXM,JMAXM
29700      COMMON/INTG/  TCOS(41),TSIN(41),TEXP(9,10),TBES(9,10)
29800      COMMON/QUADRA/AM(80),RSAV(80),TSAV(80),WM
29900      COMMON/VELO/  U(40,9),W(40,9)
30000      DATA PI/3.14159265/

```

```

30100 C ----- INITIALIZATION -----
30200 CALL QUAD(2)
30300 DO 100 M=1,MINT2
30400 EM=AM(M)
30500 EM2=EM*EM
30600 DO 10 J=1,JMAXM
30700 ZETA=.5*(ZB(J)+ZB(J+1))
30800 DO 10 JJ=1,JMAX
30900 ZED=-(ZB(JJ)+ZETA)
31000 EMR=1./EM
31100 CALL INTK0(G,ZED,EMR)
31200 TBES(J,JJ)=G*RSV(M)
31300 10 TEXP(J,JJ)=EXP(-ZED/EM2)
31400 C ----- COMPUTE FOURIER INTEGRAL -----
31500 DO 40 I=1,IMAXM
31600 XI=.5*(XB(I)+XB(I+1))
31700 DO 20 II=1,IMAX
31800 TCOS(II)=COS((XB(II)-XI)/EM)
31900 20 TSIN(II)=SIN((XB(II)-XI)/EM)
32000 DO 40 J=1,JMAXM
32100 UINT=0.
32200 WINT=0.
32300 DO 30 II=1,IMAXM
32400 COSXOM=TCOS(II)-TCOS(II+1)
32500 SINXOM=TSIN(II)-TSIN(II+1)
32600 DO 30 JJ=1,JMAXM
32700 C=(TEXP(J,JJ+1)-TEXP(J,JJ))*FB(II,JJ)
32800 D=(TBES(J,JJ+1)-TBES(J,JJ))*FB(II,JJ)
32900 UINT=UINT+C*(PI*SINXOM+TSV(M)*COSXOM)-D*COSXOM
33000 30 WINT=WINT+(C*(PI*COSXOM-TSV(M)*SINXOM)-D*SINXOM)/EM
33100 U(I,J)=U(I,J)+WM*UINT
33200 W(I,J)=W(I,J)+WM*WINT
33300 40 CONTINUE
33400 100 CONTINUE
33500 RETURN
33600 END
33700 SUBROUTINE VELOC3
33800 C *****
33900 C THIS SUBROUTINE COMPUTES THE CONTRIBUTION OF THE FINITE INTEGRAL
34000 C TO THE SURFACE VELOCITY COMPONENTS
34100 C *****
34200 COMMON/COORD/ XB(41),YB(41,10),ZB(10),FB(40,9)
34300 COMMON/DATA/ FR,IMAX,JMAX,MINT2,MINT3,NZOT,IMAXM,JMAXM
34400 COMMON/INTG/ TCOS(41),TSIN(41),TEXP(9,10),TBES(9,10)
34500 COMMON/QUADRA/AM(80),RSV(80),TSV(80),WM
34600 COMMON/VELO/ U(40,9),W(40,9)
34700 C ----- INITIALIZATION -----
34800 CALL QUAD(3)
34900 DO 100 M=1,MINT3
35000 EM=AM(M)
35100 EM2=EM*EM
35200 DO 10 J=1,JMAXM
35300 ZETA=.5*(ZB(J)+ZB(J+1))
35400 DO 10 JJ=1,JMAX
35500 ZED=-(ZB(JJ)+ZETA)
35600 CALL INTK0(G,ZED,EM)
35700 10 TEXP(J,JJ)=TSV(M)*EXP(-ZED*EM2)+RSV(M)*G
35800 C ----- COMPUTE FOURIER INTEGRAL -----
35900 DO 40 I=1,IMAXM
36000 XI=.5*(XB(I)+XB(I+1))

```

```

36100      DO 20 II=1,IMAX
36200      TCOS(II)=COS((XB(II)-XI)*EM)
36300  20    TSIN(II)=SIN((XB(II)-XI)*EM)
36400      DO 40 J=1,JMAXM
36500      UINT=0.
36600      WINT=0.
36700      DO 30 II=1,IMAXM
36800      COSXMU=TCOS(II)-TCOS(II+1)
36900      SINXMU=TSIN(II)-TSIN(II+1)
37000      DO 30 JJ=1,JMAXM
37100      C=(TEXP(J,JJ+1)-TEXP(J,JJ))*FB(II,JJ)
37200      UINT=UINT-C*COSXMU/EM
37300  30    WINT=WINT+C*SINXMU
37400      U(I,J)=U(I,J)+WM*UINT
37500      W(I,J)=W(I,J)+WM*WINT
37600  40    CONTINUE
37700  100   CONTINUE
37800      RETURN
37900      END

```

```

00100      PROGRAM SHPMSH
00200 C *****
00300 C THIS PROGRAM CALCULATES METRIC AND CURVATURE PARAMETERS FOR ARBITRAR
00400 C SHIP HULLS AND WRITES A DISK FILE FOR USE BY PROGRAM EDDY3
00500 C *****
00600      COMMON/COORD/ Z(40)
00700      COMMON/COORDB/XB(40),YB(40),ZB(40)
00800      COMMON/INTRP/ XI(40),ZI(40)
00900      COMMON/GEOMP/ H1(40),H2(40),THETA(40,40),CK1(40),CK2(40),
01000      *          CK12(40),CK21(40),PSI(40)
01100      COMMON/TEMPZ/ DYBDZ(40),D2YBDZ(40),DZBDZ(40),D2ZBDZ(40)
01200      COMMON/TEMPX/ DYBDX(40),D2YBDX(40),DZBDX(40),D2ZBDX(40)
01300      COMMON/DTHET/ DTHDX(40),DTHDZ(40),THET(40),LT(30)
01400      COMMON/XDATA/ PSIZ,XSTART,XEND,DX(10),NX(10)
01500      COMMON/ZDATA/ ZSTART,ZEND,DZ(10),NZ(10)
01600      COMMON/NDATA/ IU1,IU2,IU3,IU4,IU5,IMAX,JMAX,NXTL,NZTL,JFZ,LZ(30)
01700      COMMON/MDATA/ JFX,LX(30)
01800      COMMON/LDATA/ ISYM,ISET
01900      CALL OPEN(6,'NAMLIST ',1)
02000      CALL NAMIN
02100      CALL OPEN(IU1,'MESH ',1)
02200      CALL OPEN(IU2,'SCRATCH1 ',12)
02300      CALL OPEN(IU3,'SCRATCH2 ',28)
02400      CALL OPEN(IU4,'GEOMETRY ',32)
02500 C SET UP COMPUTATIONAL GRID
02600      CALL GRID
02700      JMAXM=JMAX-1
02800      DO 1 I=1,IMAX
02900      ISET=0
03000 C READ SECTION COORDINATES
03100      READ(IU1,1000) XB(I),YB(1),ZB(1),YB(2),ZB(2)
03200      DO 2 J=3,JMAXM,2
03300      JP=J+1
03400 2   READ(IU1,1100) YB(J),ZB(J),YB(JP),ZB(JP)
03500 C COMPUTE AND RESCALE LOCAL ARCLENGTH
03600      CALL ARC(DYDX1)
03700 C SMOOTH SECTION DATA AND COMPUTE D/DZ
03800      IF(ISYM.EQ.1) ISET=1
03900      CALL SMOOTH(Z,YB,LZ,JFZ,JMAX,1,NZTL,IU2,ZI,0,DYDX1)
04000      CALL SMOOTH(Z,ZB,LZ,JFZ,JMAX,1,NZTL,IU2,ZI,0,0.)
04100 1   CONTINUE
04200      WRITE(1,200)
04300      REWIND IU2
04400      DO 6 J=1,NZTL
04500 C READ SMOOTHED DATA FOR CONSTANT Z
04600      CALL READYZ(J)
04700 C SMOOTH ALONG Z=CONSTANT AND COMPUTE D/DX
04800      CALL SMOOTH(XB,YB,LX,JFX,IMAX,0,NXTL,IU3,XI,1,DUMMY)
04900      CALL SMOOTH(XB,ZB,LX,JFX,IMAX,0,NXTL,IU3,XI,2,DUMMY)
05000      CALL SMOOTH(XB,DYBDZ,LX,JFX,IMAX,0,NXTL,IU3,XI,4,DUMMY)
05100      CALL SMOOTH(XB,DZBDZ,LX,JFX,IMAX,0,NXTL,IU3,XI,5,DUMMY)
05200      CALL SMOOTH(XB,D2YBDZ,LX,JFX,IMAX,0,NXTL,IU3,XI,6,DUMMY)
05300      CALL SMOOTH(XB,D2ZBDZ,LX,JFX,IMAX,0,NXTL,IU3,XI,7,DUMMY)
05400 C COMPUTE H1,H2,THETA,K1,K2
05500      CALL GEOMET(J)
05600      WRITE(1,300) J
05700      REWIND IU2
05800 6   CONTINUE
05900 C COMPUTE DERIVATIVES OF THETA ON EACH SECTION
06000      REWIND IU2

```

```

06100      JFT=NZTL/2+1
06200      LT(1)=1
06300      JUP=NZTL-1
06400      LL=1
06500      DO 8 J=2,JUP,2
06600      LL=LL+1
06700  8    LT(LL)=J
06800      LT(JFT)=NZTL
06900      DO 9 I=1,NXTL
07000      DO 7 J=1,NZTL
07100  7    THET(J)=THETA(I,J)
07200      CALL SMOOTH(ZI,THET,LT,JFT,NZTL,0,NZTL,IU2,ZI,8,DUMMY)
07300      CALL SMOOTH(ZI,DTHDZ,LT,JFT,NZTL,0,NZTL,IU2,ZI,0,DUMMY)
07400  9    CONTINUE
07500      WRITE(1,400)
07600      REWIND IU2
07700      REWIND IU3
07800      JFT=NXTL/2+1
07900      LT(1)=1
08000      IUP=NXTL-1
08100      LL=1
08200      DO 13 I=2,IUP,2
08300      LL=LL+1
08400  13   LT(LL)=I
08500      LT(JFT)=NXTL
08600      DO 12 J=1,NZTL
08700  C    READ D(THETA)/DZ FROM DISK FILE FOR CONSTANT Z
08800      CALL READTH(J)
08900  C    SMOOTH THEIA ALONG Z=CONSTANT AND COMPUTE D/DX
09000      DO 10 I=1,NXTL
09100  10   READ(IU3) YB(I),ZB(I),H1(I),H2(I),THET(I),CK1(I),CK2(I)
09200      CALL SMOOTH(XI,THET,LT,JFT,NXTL,0,NXTL,IU3,XI,3,DUMMY)
09300  C    COMPUTE K12 AND K21 THEN WRITE FINAL DISK FILE
09400      WRITE(IU5,1200) ZI(J)
09500      PSI(1)=PSIZ+H1(1)*DX(1)
09600      DO 11 I=1,NXTL
09700      IF(I.LT.NXTL) PSI(I+1)=PSI(I)+.5*(H1(I)+H1(I+1))*(XI(I+1)-XI(I))
09800      FACT1=CK1(I)+DTHDX(I)/H1(I)
09900      FACT2=CK2(I)+DTHDZ(I)/H2(I)
10000      SINTH=SIN(THETA(I,J))
10100      COSTH=COS(THETA(I,J))
10200      CK12(I)=(-FACT1+COSTH*FACT2)/SINTH
10300      CK21(I)=(-FACT2+COSTH*FACT1)/SINTH
10400      WRITE(IU4) H1(I),H2(I),CK1(I),CK2(I),CK12(I),
10500      *          CK21(I),THETA(I,J),PSI(I)
10600      WRITE(IU5,100) I,XI(I),YB(I),ZB(I),H1(I),H2(I),THETA(I,J),
10700      *          CK1(I),CK2(I),CK12(I),CK21(I),PSI(I)
10800  11   CONTINUE
10900      REWIND IU2
11000  12   CONTINUE
11100  100  FORMAT(I4,5X,1P11E11.3)
11200  200  FORMAT(1X,'All Section Data Smoothed & Differentiated'//)
11300  300  FORMAT(1X,'Geometrical Parameters Computed for J = ',I3)
11400  400  FORMAT(/1X,'d(Theta)/dz Computed for all Sections'//)
11500  1000 FORMAT(3F10.5,11X,2F10.5)
11600  1100 FORMAT(10X,2F10.5,11X,2F10.5)
11700  1200 FORMAT(/1X,'Z = ',F7.3/3X1HI,10X2HXB,9X2HYB,9X2HZB,
11800      *          9X2HH1,9X2HH2,8X5HTHETA,7X2HK1,9X2HK2,
11900      *          9X3HK12,8X3HK21,8X3HPSI)
12000      END

```

```

12100      SUBROUTINE ANGLE(COSTH,SINTH,THETA)
12200 C *****
12300 C THIS SUBROUTINE CALCULATES AN ANGLE IN EITHER THE FIRST OR SECOND
12400 C QUADRANT GIVEN THE COSINE OF THE ANGLE
12500 C *****
12600      IF(COSTH.EQ.0.) GO TO 2
12700      IF(COSTH.GT.0.) GO TO 1
12800      SINPHI=-COSTH
12900      COSPHI=SQRT(1.-SINPHI**2)
13000      TANPHI=SINPHI/COSPHI
13100      PHI=ATAN(TANPHI)
13200      THETA=355./226.+PHI
13300      SINTH=COSPHI
13400      RETURN
13500 1     SINTH=SQRT(1.-COSTH**2)
13600      TANTH=SINTH/COSTH
13700      THETA=ATAN(TANTH)
13800      RETURN
13900 2     SINTH=1.
14000      THETA=355./226.
14100      RETURN
14200      END
14300      SUBROUTINE ARC(DYDX1)
14400 C *****
14500 C THIS SUBROUTINE COMPUTES LOCAL ARCLENGTH AND RESCALES TO UNITY
14600 C *****
14700      COMMON/COORD/ Z(40)
14800      COMMON/COORDB/XB(40),YB(40),ZB(40)
14900      COMMON/NDATA/ IU1,IU2,IU3,IU4,IU5,IMAX,JMAX,NXTL,NZTL,JFZ,LZ(30)
15000      Z(1)=0.
15100      DO 1 J=2,JMAX
15200      JM=J-1
15300      DYB=YB(J)-YB(JM)
15400      DZB=ZB(J)-ZB(JM)
15500      DZ=SQRT(DYB*DYB+DZB*DZB)
15600 1     Z(J)=Z(JM)+DZ
15700      ZSCALE=Z(JMAX)
15800      DO 2 J=2,JMAX
15900 2     Z(J)=Z(J)/ZSCALE
16000      DYDX1=-ZSCALE
16100      RETURN
16200      END
16300      SUBROUTINE GEOMET(J)
16400 C *****
16500 C THIS SUBROUTINE CALCULATES THE METRIC AND CURVATURE PARAMETERS
16600 C *****
16700      COMMON/COORDB/XB(40),YB(40),ZB(40)
16800      COMMON/GEOMP/ H1(40),H2(40),THETA(40,40),CK1(40),CK2(40),
16900      *          CK12(40),CK21(40),PSI(40)
17000      COMMON/TEMPZ/ DYBDZ(40),D2YBDZ(40),DZBDZ(40),D2ZBDZ(40)
17100      COMMON/TEMPX/ DYBDX(40),D2YBDX(40),DZBDX(40),D2ZBDX(40)
17200      COMMON/NDATA/ IU1,IU2,IU3,IU4,IU5,IMAX,JMAX,NXTL,NZTL,JFZ,LZ(30)
17300      DO 3 I=1,NXTL
17400      H1(I)=SQRT(1.+DZBDX(I)**2+DYBDX(I)**2)
17500      H2(I)=SQRT(DYBDZ(I)**2+DZBDZ(I)**2)
17600      COSTH=(DYBDZ(I)*DYBDX(I)+DZBDX(I)*DZBDZ(I))/(H1(I)*H2(I))
17700      CALL ANGLE(COSTH,SINTH,THETA(I,J))
17800      FACT1=DYBDZ(I)*DZBDX(I)-DYBDX(I)*DZBDZ(I)
17900      FACT2=DZBDX(I)*D2YBDX(I)-DYBDX(I)*D2ZBDX(I)
18000      FACT3=DYBDZ(I)*D2YBDX(I)+DZBDZ(I)*D2ZBDX(I)

```

```

18100      DENOM=H1(I)**4*H2(I)*SINTH
18200      CK1(I)=(FACT1*FACT2+FACT3)/DENOM
18300      FACT2=DZBDZ(I)*D2YBDZ(I)-DYBDZ(I)*D2ZBDZ(I)
18400      DENOM=H1(I)*H2(I)**4*SINTH
18500      CK2(I)=-FACT1*FACT2/DENOM
18600      WRITE(IU3) YB(I),ZB(I),H1(I),H2(I),THETA(I,J),CK1(I),CK2(I)
18700  3    CONTINUE
18800      RETURN
18900      END
19000      SUBROUTINE GRID
19100  C *****
19200  C THIS SUBROUTINE SETS UP THE COMPUTATIONAL GRID
19300  C *****
19400      COMMON/INTRP/ XI(40),ZI(40)
19500      COMMON/XDATA/ PSIZ,XSTART,XEND,DX(10),NX(10)
19600      COMMON/ZDATA/ ZSTART,ZEND,DZ(10),NZ(10)
19700      COMMON/NDATA/ IU1,IU2,IU3,IU4,IU5,IMAX,JMAX,NXTL,NZTL,JFZ,LZ(30)
19800      XI(1)=XSTART+DX(1)
19900      K=1
20000      MCUT=NX(1)
20100      NXTLM=NXTL-1
20200      DO 1 I=1,NXTLM
20300      IF(I.LE.MCUT) GO TO 1
20400      K=K+1
20500      MCUT=MCUT+NX(K)
20600  1    XI(I+1)=XI(I)+DX(K)
20700      IF(ABS(XI(NXTL)-XEND).GT..1*DX(K)) GO TO 7
20800      XI(NXTL)=XEND
20900      ZI(1)=ZSTART
21000      K=1
21100      MCUT=NZ(1)
21200      NZTLM=NZTL-1
21300      DO 3 I=1,NZTLM
21400      IF(I.LE.MCUT) GO TO 3
21500      K=K+1
21600      MCUT=MCUT+NZ(K)
21700  3    ZI(I+1)=ZI(I)+DZ(K)
21800      IF(ABS(ZI(NZTL)-ZEND).GT..1*DZ(K)) GO TO 8
21900      ZI(NZTL)=ZEND
22000      DO 4 I=1,NXTL
22100  4    WRITE(IU4) XI(I)
22200      DO 5 I=1,NZTL
22300  5    WRITE(IU4) ZI(I)
22400      RETURN
22500  7    WRITE(IU5,100) XI(NXTL),XEND
22600      STOP
22700  8    WRITE(IU5,200) ZI(NZTL),ZEND
22800      STOP
22900  100  FORMAT(/1X,'STEPSIZE ARRAY IMPLIES XEND =',1PE13.6,
23000      *' WHICH DOES NOT MATCH THE DESIRED VALUE...',E13.6)
23100  200  FORMAT(/1X,'STEPSIZE ARRAY IMPLIES ZEND =',1PE13.6,
23200      *' WHICH DOES NOT MATCH THE DESIRED VALUE...',E13.6)
23300      END
23400      SUBROUTINE INITL(X,Y,N,L,JF,ISTART,DYDX1)
23500  C *****
23600  C THIS SUBROUTINE SETS UP INITIAL VALUES FOR THE VARIOUS ARRAYS
23700  C *****
23800      DIMENSION X(40),Y(40),L(30)
23900      COMMON/QUAN/ A(30),B(30),PH0PH0(30),PH0PH1(30),
24000      * PH0PS0(30),PH0PS1(30),PH1PH1(30),PH1PS0(30),

```



```

24100 *          PH1PS1(30),P1(30),P2(30),P3(30),Q1(30),
24200 *          Q2(30),Q3(30),XI(30),YPHI0(30),YPHI1(30),
24300 *          H(30)
24400 COMMON/DIMN/ JFM
24500 COMMON/LDATA/ ISYM,ISET
24600 JFM=JF-1
24700 Y1=Y(1)
24800 YN=Y(N)
24900 IF(ISTART.NE.0) GO TO 2
25000 X32=X(3)-X(2)
25100 X31=X(3)-X(1)
25200 X21=X(2)-X(1)
25300 AA=-(X21+X31)/(X21*X31)
25400 BB=X31/(X21*X32)
25500 CC=-X21/(X31*X32)
25600 DYDX1=AA*Y(1)+BB*Y(2)+CC*Y(3)
25700 2 X32=X(N-2)-X(N-1)
25800 X31=X(N-2)-X(N)
25900 X21=X(N-1)-X(N)
26000 AA=-(X21+X31)/(X21*X31)
26100 BB=X31/(X21*X32)
26200 CC=-X21/(X31*X32)
26300 DYDXN=AA*Y(N)+BB*Y(N-1)+CC*Y(N-2)
26400 IF(ISET.EQ.0) GO TO 3
26500 DYDXN=0.
26600 YN=-(BB*Y(N-1)+CC*Y(N-2))/AA
26700 3 ISET=0
26800 P1(1)=Y1
26900 P2(1)=0.
27000 P3(1)=0.
27100 Q1(1)=DYDX1
27200 Q2(1)=0.
27300 Q3(1)=0.
27400 A(JF)=YN
27500 B(JF)=DYDXN
27600 DO 1 J=1,JF
27700 K=L(J)
27800 XI(J)=X(K)
27900 PH0PH0(J)=0.
28000 PH0PH1(J)=0.
28100 PH0PS0(J)=0.
28200 PH0PS1(J)=0.
28300 PH1PH1(J)=0.
28400 PH1PS0(J)=0.
28500 PH1PS1(J)=0.
28600 YPHI0(J)=0.
28700 YPHI1(J)=0.
28800 1 CONTINUE
28900 RETURN
29000 END
29100 SUBROUTINE INTERP(I,J,XX,IU2,IFLAG)
29200 C *****
29300 C THIS SUBROUTINE USES THE SPLINE FIT TO INTERPOLATE THE ORIGINAL
29400 C DATA AND TO COMPUTE ITS FIRST AND SECOND DERIVATIVES
29500 C *****
29600 COMMON/QUAN/ A(30),B(30),PH0PH0(30),PH0PH1(30),
29700 *          PH0PS0(30),PH0PS1(30),PH1PH1(30),PH1PS0(30),
29800 *          PH1PS1(30),P1(30),P2(30),P3(30),Q1(30),
29900 *          Q2(30),Q3(30),XI(30),YPHI0(30),YPHI1(30),
30000 *          H(30)

```

```

30100 COMMON/TEMPX/ DYBDX(40),D2YBDX(40),DZBDX(40),D2ZBDX(40)
30200 COMMON/TEMPZ/ DYBDZ(40),D2YBDZ(40),DZBDZ(40),D2ZBDZ(40)
30300 COMMON/COORDB/XB(40),YB(40),ZB(40)
30400 COMMON/DTHET/ DTHDX(40),DTHDZ(40),THET(40),LT(30)
30500 COMMON/DIMN/ JFM
30600 IF(XX.LE.XI(I+1)) GO TO 1
30700 I=I+1
30800 1 IP=I+1
30900 DX=(XX-XI(I))/H(I)
31000 DXP=(XI(IP)-XX)/H(I)
31100 PHI1=(3.-2.*DX)*DX**2
31200 PHI0=1.-PHI1
31300 PSI1=-H(I)*DXP*DX**2
31400 PSI0=H(I)*DX*DXP**2
31500 PHI1P=6.*DX*(1.-DX)/H(I)
31600 PHI0P=-PHI1P
31700 PSI1P=-DX*(2.*DXP-DX)
31800 PSI0P=DXP*(DXP-2.*DX)
31900 PH1PP=6.*(1.-2.*DX)/H(I)**2
32000 PH0PP=-PH1PP
32100 PS1PP=2.*(2.*DX-DXP)/H(I)
32200 PS0PP=2.*(DX-2.*DXP)/H(I)
32300 YY=A(I)*PHI0 +A(IP)*PHI1 +B(I)*PSI0 +B(IP)*PSI1
32400 YP=A(I)*PHI0P+A(IP)*PHI1P+B(I)*PSI0P+B(IP)*PSI1P
32500 Y2=A(I)*PH0PP+A(IP)*PH1PP+B(I)*PS0PP+B(IP)*PS1PP
32600 IF(IFLAG.GT.0) GO TO 2
32700 WRITE(IU2) YY,YP,Y2
32800 RETURN
32900 2 IF(IFLAG.GT.2) GO TO 4
33000 IF(IFLAG.EQ.2) GO TO 3
33100 YB(J)=YY
33200 DYBDX(J)=YP
33300 D2YBDX(J)=Y2
33400 RETURN
33500 3 ZB(J)=YY
33600 DZBDX(J)=YP
33700 D2ZBDX(J)=Y2
33800 RETURN
33900 4 IF(IFLAG.EQ.3) DTHDX(J)=YP
34000 IF(IFLAG.EQ.4) DYBDZ(J)=YY
34100 IF(IFLAG.EQ.5) DZBDZ(J)=YY
34200 IF(IFLAG.EQ.6) D2YBDZ(J)=YY
34300 IF(IFLAG.EQ.7) D2ZBDZ(J)=YY
34400 IF(IFLAG.EQ.8) DTHDZ(J)=YP
34500 RETURN
34600 END
34700 SUBROUTINE NAMIN
34800 C *****
34900 C THIS SUBROUTINE COORDINATES READING OF INPUT DATA
35000 C *****
35100 COMMON/NDATA/ N(40)
35200 COMMON/MDATA/ M(31)
35300 COMMON/LDATA/ ISYM, ISET
35400 COMMON/XDATA/ X(13),NX(10)
35500 COMMON/ZDATA/ Z(12),NZ(10)
35600 ISYM=0
35700 CALL NAMLST(6,2, ISYM, DUMMY)
35800 DO 1 I=1,10
35900 N(I)=0
36000 1 CALL NAMLST(6,2,N(I), DUMMY)

```

```

36100      IUP=10+N(10)
36200      DO 2 I=11,IUP
36300      N(I)=0
36400  2    CALL NAMLST(6,2,N(I),DUMMY)
36500      M(1)=0
36600      CALL NAMLST(6,2,M(1),DUMMY)
36700      IUP=1+M(1)
36800      DO 3 I=2,IUP
36900      M(I)=0
37000  3    CALL NAMLST(6,2,M(I),DUMMY)
37100      DO 4 I=1,13
37200  4    CALL NAMLST(6,2,1,X(I))
37300      DO 5 I=1,10
37400      NX(I)=0
37500  5    CALL NAMLST(6,2,NX(I),DUMMY)
37600      DO 6 I=1,12
37700  6    CALL NAMLST(6,2,1,Z(I))
37800      DO 7 I=1,10
37900      NZ(I)=0
38000  7    CALL NAMLST(6,2,NZ(I),DUMMY)
38100      RETURN
38200      END
38300      SUBROUTINE NAMLST(II,IO,N,X)
38400  C *****
38500  C THIS SUBROUTINE READS INPUT DATA
38600  C *****
38700      DIMENSION A(3)
38800      IF(N .NE. 0) GO TO 1
38900      READ (II,2) A(1),A(2),A(3),N
39000      WRITE(IO,2) A(1),A(2),A(3),N
39100      RETURN
39200  1    READ (II,3) A(1),A(2),A(3),X
39300      WRITE(IO,3) A(1),A(2),A(3),X
39400      RETURN
39500  2    FORMAT(1X,3A4,I4)
39600  3    FORMAT(1X,3A4,1PE13.6)
39700      END
39800      SUBROUTINE READTH(J)
39900  C *****
40000  C THIS SUBROUTINE READS D(THETA)/DZ ON A SECTION FROM DISK
40100  C *****
40200      COMMON/DTHET/ DTHDX(40),DTHDZ(40),THET(40),LT(30)
40300      COMMON/NDATA/ IU1,IU2,IU3,IU4,IU5,IMAX,JMAX,NXTL,NZTL,JFZ,LZ(30)
40400      DO 3 I=1,NXTL
40500      DO 1 JJ=1,J
40600  1    READ(IU2) YY,D,D
40700      DTHDZ(I)=YY
40800      IF(J.EQ.NZTL) GO TO 3
40900      JP=J+1
41000      DO 2 JJ=JP,NZTL
41100  2    READ(IU2) D,D,D
41200  3    CONTINUE
41300      RETURN
41400      END
41500      SUBROUTINE READYZ(J)
41600  C *****
41700  C THIS SUBROUTINE READS SMOOTHED SECTION DATA FROM DISK AND FILLS
41800  C ARRAYS IN PREPARATION FOR SMOOTHING IN THE STREAMWISE DIRECTION
41900  C *****
42000      COMMON/COORDB/XB(40),YB(40),ZB(40)

```

```

42100      COMMON/TEMPZ/ DYBDZ(40),D2YBDZ(40),DZBDZ(40),D2ZBDZ(40)
42200      COMMON/NDATA/ IU1,IU2,IU3,IU4,IU5,IMAX,JMAX,NXTL,NZTL,JFZ,LZ(30)
42300      DO 6 I=1,IMAX
42400      DO 1 JJ=1,J
42500  1    READ(IU2) YY,YP,Y2
42600      YB(I)=YY
42700      DYBDZ(I)=YP
42800      D2YBDZ(I)=Y2
42900      IF(J.EQ.NZTL) GO TO 3
43000      JP=J+1
43100      DO 2 JJ=JP,NZTL
43200  2    READ(IU2) D,D,D
43300  3    CONTINUE
43400      DO 4 JJ=1,J
43500  4    READ(IU2) ZZ,ZP,Z2
43600      ZB(I)=ZZ
43700      DZBDZ(I)=ZP
43800      D2ZBDZ(I)=Z2
43900      IF(J.EQ.NZTL) GO TO 6
44000      DO 5 JJ=JP,NZTL
44100  5    READ(IU2) D,D,D
44200  6    CONTINUE
44300      RETURN
44400      END
44500      SUBROUTINE SMOOTH(X,Y,L,JF,N,ISTART,NN,IU2,XT,IFLAG,DYDX1)
44600  C *****
44700  C THIS SUBROUTINE FITS A TABULATED FUNCTION WITH CUBIC SPLINES USING
44800  C LEAST-SQUARES ERROR MINIMIZATION
44900  C *****
45000      DIMENSION X(40),Y(40),L(30),XT(40)
45100      COMMON/QUAN/ A(30),B(30),PH0PH0(30),PH0PH1(30),
45200      *             PH0PS0(30),PH0PS1(30),PH1PH1(30),PH1PS0(30),
45300      *             PH1PS1(30),P1(30),P2(30),P3(30),Q1(30),
45400      *             Q2(30),Q3(30),XI(30),YPHI0(30),YPHI1(30),
45500      *             H(30)
45600      COMMON/DIMN/ JFM
45700  C INITIATE THE COMPUTATION
45800      CALL INITL(X,Y,N,L,JF,ISTART,DYDX1)
45900      CALL SUMUP(X,Y,N,L,JF)
46000  C COMPUTE THE P'S AND Q'S
46100      DO 1 J=2,JFM
46200      M=J-1
46300  C COEFFICIENTS IN THE ORIGINAL EQUATIONS
46400      A1=PH0PH1(M)
46500      B1=PH1PH1(M)+PH0PH0(J)
46600      C1=PH0PH1(J)
46700      D1=PH1PS0(M)
46800      E1=PH1PS1(M)+PH0PS0(J)
46900      F1=PH0PS1(J)
47000      G1=YPHI1(M)+YPHI0(J)
47100      A2=3./H(M)**2
47200      B2=3.*(1./H(J)**2-1./H(M)**2)
47300      C2=-3./H(J)**2
47400      D2=1./H(M)
47500      E2=2.*(1./H(J)+1./H(M))
47600      F2=1./H(J)
47700      G2=0.
47800  C MODIFIED COEFFICIENTS
47900      B1S=B1+A1*P2(M)+D1*Q2(M)
48000      E1S=E1+A1*P3(M)+D1*Q3(M)

```

```

48100      G1S=G1-A1*P1(M)-D1*Q1(M)
48200      B2S=B2+A2*P2(M)+D2*Q2(M)
48300      E2S=E2+A2*P3(M)+D2*Q3(M)
48400      G2S=G2-A2*P1(M)-D2*Q1(M)
48500 C    THE P'S AND Q'S
48600      DEL=1./ (B1S*E2S-B2S*E1S)
48700      P1(J)=(E2S*G1S-E1S*G2S)*DEL
48800      P2(J)=(E1S*C2 -E2S*C1 )*DEL
48900      P3(J)=(E1S*F2 -E2S*F1 )*DEL
49000      Q1(J)=(B1S*G2S-B2S*G1S)*DEL
49100      Q2(J)=(B2S*C1 -B1S*C2 )*DEL
49200      Q3(J)=(B2S*F1 -B1S*F2 )*DEL
49300 1    CONTINUE
49400 C    SOLVE FOR THE A'S AND B'S
49500      K=JFM
49600      KP=JF
49700      DO 2 J=1,JFM
49800      A(K)=P1(K)+P2(K)*A(KP)+P3(K)*B(KP)
49900      B(K)=Q1(K)+Q2(K)*A(KP)+Q3(K)*B(KP)
50000      KP=K
50100      K=K-1
50200 2    CONTINUE
50300 C    INTERPOLATE TO DESIRED MESH
50400      I=1
50500      DO 3 J=1,NN
50600      CALL INTERP(I,J,XT(J),IU2,IFLAG)
50700 3    CONTINUE
50800      RETURN
50900      END
51000      SUBROUTINE SUMUP(X,Y,N,L,JF)
51100 C    *****
51200 C    THIS SUBROUTINE PERFORMS SUMMING OPERATIONS ON THE CUBIC-SPLINE
51300 C    WEIGHTING FUNCTIONS
51400 C    *****
51500      DIMENSION X(40),Y(40),L(30)
51600      COMMON/QUAN/ A(30),B(30),PH0PH0(30),PH0PH1(30),
51700      *             PH0PS0(30),PH0PS1(30),PH1PH1(30),PH1PS0(30),
51800      *             PH1PS1(30),P1(30),P2(30),P3(30),Q1(30),
51900      *             Q2(30),Q3(30),XI(30),YPHI0(30),YPHI1(30),
52000      *             H(30)
52100      COMMON/DIMN/ JFM
52200      DO 2 J=1,JFM
52300      H(J)=XI(J+1)-XI(J)
52400      KL=L(J)
52500      KU=L(J+1)
52600      DO 1 K=KL,KU
52700      DX=(X(K)-XI(J))/H(J)
52800      DXP=(XI(J+1)-X(K))/H(J)
52900      YY=Y(K)
53000      PHI1=(3.-2.*DX)*DX**2
53100      PHI0=1.-PHI1
53200      PSI1=-H(J)*DXP*DX**2
53300      PSI0= H(J)*DX*DXP**2
53400      PH0PH0(J)=PH0PH0(J)+PHI0*PHI0
53500      PH0PH1(J)=PH0PH1(J)+PHI0*PHI1
53600      PH0PS0(J)=PH0PS0(J)+PHI0*PSI0
53700      PH0PS1(J)=PH0PS1(J)+PHI0*PSI1
53800      PH1PH1(J)=PH1PH1(J)+PHI1*PHI1
53900      PH1PS0(J)=PH1PS0(J)+PHI1*PSI0
54000      PH1PS1(J)=PH1PS1(J)+PHI1*PSI1

```

```
54100      YPHI0 (J)=YPHI0 (J)+PHI0*YY  
54200      YPHI1 (J)=YPHI1 (J)+PHI1*YY  
54300      1      CONTINUE  
54400      2      CONTINUE  
54500      RETURN  
54600      END
```

```

00100      PROGRAM VELOC
00200 C *****
00300 C THIS PROGRAM SMOOTHS AND INTERPOLATES BOUNDARY LAYER EDGE VELOCITY
00400 C COMPONENTS ONTO A RECTANGULAR GRID IN TRANSFORMED COORDINATES AND
00500 C WRITES A DISK FILE FOR USE BY PROGRAM EDDY3
00600 C *****
00700      COMMON/COORD/ Z(40)
00800      COMMON/COORDB/XB(40),YB(40),ZB(40)
00900      COMMON/DERIV/ DYBDZ(40),DZBDZ(40),DWDZ1(40),DWDZN(40)
01000      COMMON/INTRP/ XI(40),ZI(40)
01100      COMMON/VEL/   UE(40),VE(40),WE(40)
01200      COMMON/VELI/  UEI(40,40),WEI(40,40)
01300      COMMON/NDATA/ IU1,IU2,IU3,IU4,IU5,IMAX,JMAX,NXTL,NZTL,JFZ,LZ(30)
01400      COMMON/MDATA/ JFX,LX(30)
01500      COMMON/LDATA/ ISYM,ISET
01600      CALL OPEN(6,'NAMVEL ',1)
01700      CALL NAMIN
01800      CALL OPEN(IU1,'UEWE ',1)
01900      CALL OPEN(IU3,'VELOCITY ',8)
02000      CALL OPEN(IU4,'GEOMETRY ',32)
02100      DO 1 I=1,NXTL
02200 1      READ(IU4) XI(I)
02300      DO 2 I=1,NZTL
02400 2      READ(IU4) ZI(I)
02500      JMAX=JMAX+2
02600      JMAXM=JMAX-1
02700      DO 4 I=1,IMAX
02800      ISET=0
02900 C READ SECTION VELOCITIES
03000      READ(IU1,1000) XB(I),YB(2),ZB(2),CX,CY,CZ,Q
03100      UE(2)=Q*CX
03200      VE(2)=Q*CY
03300      WE(2)=Q*CZ
03400      DO 3 J=3,JMAXM
03500      READ(IU1,1100) YB(J),ZB(J),CX,CY,CZ,Q
03600      UE(J)=Q*CX
03700      VE(J)=Q*CY
03800 3      WE(J)=Q*CZ
03900 C COMPUTE VELOCITIES ON KEEL AND FREE SURFACE
04000      CALL ENDS
04100 C COMPUTE AND RESCALE LOCAL ARCLength
04200      CALL ARC(DYDX1)
04300 C COMPUTE TANGENTIAL VELOCITY ON SECTION
04400      IF(ISYM.EQ.1) ISET=1
04500      CALL SMOOTH(Z,YB,LZ,JFZ,JMAX,1,JMAX,Z,6,DYDX1,IDUM)
04600      CALL SMOOTH(Z,ZB,LZ,JFZ,JMAX,1,JMAX,Z,7,0.,IDUM)
04700      DO 10 J=1,JMAX
04800      H2=SQRT(DYBDZ(J)**2+DZBDZ(J)**2)
04900      Q2=UE(J)**2+VE(J)**2+WE(J)**2
05000      WE(J)=(VE(J)*DYBDZ(J)+WE(J)*DZBDZ(J))/H2
05100 10      UE(J)=SQRT(Q2-WE(J)**2)
05200 C SMOOTH AND INTERPOLATE ORTHOGONAL VELOCITIES
05300      CALL SMOOTH(Z,UE,LZ,JFZ,JMAX,1,NZTL,ZI,0,0.,I)
05400      CALL SMOOTH(Z,WE,LZ,JFZ,JMAX,0,NZTL,ZI,1,DUMMY,I)
05500      WRITE(1,150) I
05600 4      CONTINUE
05700      WRITE(1,200)
05800 C SMOOTH AND INTERPOLATE DATA ALONG Z=CONSTANT
05900      DO 6 J=1,NZTL
06000      DO 5 I=1,IMAX

```

```

06100      UE(I)=UEI(I,J)
06200  5      WE(I)=WEI(I,J)
06300      CALL SMOOTH(XB,UE,LX,JFX,IMAX,0,NXTL,XI,2,DUMMY,IDUM)
06400      CALL SMOOTH(XB,WE,LX,JFX,IMAX,0,NXTL,XI,3,DUMMY,IDUM)
06500  C      COMPUTE NONORTHOGONAL VELOCITY COMPONENTS
06600      CALL ROTATE(J)
06700  6      CONTINUE
06800      CALL SMOOTH(XB,DWDZ1,LX,JFX,IMAX,0,NXTL,XI,4,DUMMY,IDUM)
06900      CALL SMOOTH(XB,DWDZN,LX,JFX,IMAX,0,NXTL,XI,5,DUMMY,IDUM)
07000  C      WRITE DISK FILE FOR SMOOTHED DATA ALONG Z=CONSTANT
07100      DO 7 J=1,NZTL
07200      WRITE(IU5,1200) ZI(J)
07300      DO 8 I=1,NXTL
07400      WRITE(IU3) UEI(I,J),WEI(I,J)
07500  8      WRITE(IU5,100) I,XI(I),UEI(I,J),WEI(I,J)
07600  7      CONTINUE
07700      WRITE(IU5,250)
07800      DO 20 I=1,NXTL
07900      WRITE(IU3) DWDZ1(I),DWDZN(I)
08000  20      WRITE(IU5,100) I,XI(I),DWDZ1(I),DWDZN(I)
08100  100     FORMAT(I4,5X,1P3E11.3)
08200  150     FORMAT(1X,'Section Data Smoothed & Differentiated for I =',I3)
08300  200     FORMAT(1X,'All Section Data Smoothed & Differentiated'//)
08400  250     FORMAT(/3X1HI,10X2HXB,8X5HDWDZ1,7X5HDWDZN)
08500  1000    FORMAT(3F10.5,4F10.6)
08600  1100    FORMAT(10X,2F10.5,4F10.6)
08700  1200    FORMAT(/1X,'Z = ',F7.3/3X1HI,10X2HXB,9X2HUE,9X2HWE)
08800      END
08900      SUBROUTINE ARC(DYDX1)
09000  C      *****
09100  C      THIS SUBROUTINE COMPUTES LOCAL ARCLength AND RESCALES TO UNITY
09200  C      *****
09300      COMMON/COORD/ Z(40)
09400      COMMON/COORDB/XB(40),YB(40),ZB(40)
09500      COMMON/NDATA/ IU1,IU2,IU3,IU4,IU5,IMAX,JMAX,NXTL,NZTL,JFZ,LZ(30)
09600      Z(1)=0.
09700      DO 1 J=2,JMAX
09800      JM=J-1
09900      DYB=YB(J)-YB(JM)
10000      DZB=ZB(J)-ZB(JM)
10100      DZ=SQRT(DYB*DYB+DZB*DZB)
10200  1      Z(J)=Z(JM)+DZ
10300      ZSCALE=Z(JMAX)
10400      DO 2 J=2,JMAX
10500  2      Z(J)=Z(J)/ZSCALE
10600      DYDX1=-ZSCALE
10700      RETURN
10800      END
10900      SUBROUTINE ENDS
11000  C      *****
11100  C      THIS SUBROUTINE COMPUTES VELOCITIES ON THE KEEL AND FREE SURFACE
11200  C      *****
11300      COMMON/COORDB/XB(40),YB(40),ZB(40)
11400      COMMON/VEL/ UE(40),VE(40),WE(40)
11500      COMMON/NDATA/ IU1,IU2,IU3,IU4,IU5,IMAX,JMAX,NXTL,NZTL,JFZ,LZ(30)
11600      Y2SQ=YB(2)**2
11700      Y3SQ=YB(3)**2
11800      D=1./(Y3SQ-Y2SQ)
11900      UE(1)=(Y3SQ*UE(2)-Y2SQ*UE(3))*D
12000      VE(1)=0.

```



```

12100 WE(1)=(Y3SQ*WE(2)-Y2SQ*WE(3))*D
12200 YB(1)=0.
12300 ZB(1)=(Y3SQ*ZB(2)-Y2SQ*ZB(3))*D
12400 JM1=JMAX-1
12500 JM2=JMAX-2
12600 ZNM1=ZB(JM1)
12700 ZNM2=ZB(JM2)
12800 D=1./((ZNM1-ZNM2)
12900 UE(JMAX)=(ZNM1*UE(JM2)-ZNM2*UE(JM1))*D
13000 VE(JMAX)=(ZNM1*VE(JM2)-ZNM2*VE(JM1))*D
13100 WE(JMAX)=0.
13200 YB(JMAX)=(ZNM1*YB(JM2)-ZNM2*YB(JM1))*D
13300 ZB(JMAX)=0.
13400 RETURN
13500 END
13600 SUBROUTINE INITL(X,Y,N,L,JF,ISTART,DYDX1)
13700 C *****
13800 C THIS SUBROUTINE SETS UP INITIAL VALUES FOR THE VARIOUS ARRAYS
13900 C *****
14000 DIMENSION X(40),Y(40),L(30)
14100 COMMON/QUAN/ A(30),B(30),PH0PH0(30),PH0PH1(30),
14200 * PH0PS0(30),PH0PS1(30),PH1PH1(30),PH1PS0(30),
14300 * PH1PS1(30),P1(30),P2(30),P3(30),Q1(30),
14400 * Q2(30),Q3(30),XI(30),YPHI0(30),YPHI1(30),
14500 * H(30)
14600 COMMON/DIMN/ JFM,NFS
14700 COMMON/LDATA/ ISYM,ISET
14800 JFM=JF-1
14900 Y1=Y(1)
15000 YN=Y(N)
15100 IF(ISTART.NE.0) GO TO 2
15200 X32=X(3)-X(2)
15300 X31=X(3)-X(1)
15400 X21=X(2)-X(1)
15500 AA=-(X21+X31)/(X21*X31)
15600 BB=X31/(X21*X32)
15700 CC=-X21/(X31*X32)
15800 DYDX1=AA*Y(1)+BB*Y(2)+CC*Y(3)
15900 2 X32=X(N-2)-X(N-1)
16000 X31=X(N-2)-X(N)
16100 X21=X(N-1)-X(N)
16200 AA=-(X21+X31)/(X21*X31)
16300 BB=X31/(X21*X32)
16400 CC=-X21/(X31*X32)
16500 DYDXN=AA*Y(N)+BB*Y(N-1)+CC*Y(N-2)
16600 IF(ISET.EQ.0) GO TO 3
16700 DYDXN=0.
16800 YN=-(BB*Y(N-1)+CC*Y(N-2))/AA
16900 3 ISET=0
17000 P1(1)=Y1
17100 P2(1)=0.
17200 P3(1)=0.
17300 Q1(1)=DYDX1
17400 Q2(1)=0.
17500 Q3(1)=0.
17600 A(JF)=YN
17700 B(JF)=DYDXN
17800 DO 1 J=1,JF
17900 K=L(J)
18000 XI(J)=X(K)

```

```

18100      PH0PH0(J)=0.
18200      PH0PH1(J)=0.
18300      PH0PS0(J)=0.
18400      PH0PS1(J)=0.
18500      PH1PH1(J)=0.
18600      PH1PS0(J)=0.
18700      PH1PS1(J)=0.
18800      YPHI0 (J)=0.
18900      YPHI1 (J)=0.
19000  1    CONTINUE
19100      RETURN
19200      END
19300      SUBROUTINE INTERP(I,J,XX,IFLAG,II)
19400  C *****
19500  C THIS SUBROUTINE USES THE SPLINE FIT TO INTERPOLATE THE ORIGINAL
19600  C DATA AND TO COMPUTE ITS FIRST DERIVATIVE
19700  C *****
19800      COMMON/QUAN/ A(30),B(30),PH0PH0(30),PH0PH1(30),
19900      *              PH0PS0(30),PH0PS1(30),PH1PH1(30),PH1PS0(30),
20000      *              PH1PS1(30),P1(30),P2(30),P3(30),Q1(30),
20100      *              Q2(30),Q3(30),XI(30),YPHI0(30),YPHI1(30),
20200      *              H(30)
20300      COMMON/DERIV/ DYBDZ(40),DZBDZ(40),DWDZ1(40),DWDZN(40)
20400      COMMON/VEL/   UE(40),VE(40),WE(40)
20500      COMMON/VELI/  UEI(40,40),WEI(40,40)
20600      COMMON/DIMN/  JFM,NFS
20700      IF(XX.LE.XI(I+1)) GO TO 1
20800      I=I+1
20900  1    IP=I+1
21000      DX=(XX-XI(I))/H(I)
21100      DXP=(XI(IP)-XX)/H(I)
21200      IF(IFLAG.GT.5) GO TO 2
21300      PHI1=(3.-2.*DX)*DX**2
21400      PHI0=1.-PHI1
21500      PSI1=-H(I)*DXP*DX**2
21600      PSI0=H(I)*DX*DXP**2
21700      YY=A(I)*PHI0+A(IP)*PHI1+B(I)*PSI0+B(IP)*PSI1
21800      IF(IFLAG.EQ.0) UEI(II,J)=YY
21900      IF(IFLAG.NE.1) GO TO 3
22000      WEI(II,J)=YY
22100      IF(J.EQ.1.OR.J.EQ.NFS) GO TO 2
22200  3    IF(IFLAG.EQ.2) UE(J)=YY
22300      IF(IFLAG.EQ.3) WE(J)=YY
22400      IF(IFLAG.EQ.4) DWDZ1(J)=YY
22500      IF(IFLAG.EQ.5) DWDZN(J)=YY
22600      RETURN
22700  2    PHI1P=6.*DX*(1.-DX)/H(I)
22800      PHI0P=-PHI1P
22900      PSI1P=-DX*(2.*DXP-DX)
23000      PSI0P=DXP*(DXP-2.*DX)
23100      YP=A(I)*PHI0P+A(IP)*PHI1P+B(I)*PSI0P+B(IP)*PSI1P
23200      IF(IFLAG.EQ.1.AND.J.EQ.1) DWDZ1(II)=YP
23300      IF(IFLAG.EQ.1.AND.J.EQ.NFS) DWDZN(II)=YP
23400      IF(IFLAG.EQ.6) DYBDZ(J)=YP
23500      IF(IFLAG.EQ.7) DZBDZ(J)=YP
23600      RETURN
23700      END
23800      SUBROUTINE NAMIN
23900  C *****
24000  C THIS SUBROUTINE COORDINATES READING OF INPUT DATA

```

```

24100 C *****
24200     COMMON/NDATA/ N(40)
24300     COMMON/MDATA/ M(31)
24400     COMMON/LDATA/ ISYM, ISET
24500     ISYM=0
24600     CALL NAMLST(6,2, ISYM, DUMMY)
24700     DO 1 I=1,10
24800         N(I) = 0
24900     1   CALL NAMLST(6,2,N(I), DUMMY)
25000         IUP=10+N(10)
25100     DO 2 I=11, IUP
25200         N(I)=0
25300     2   CALL NAMLST(6,2,N(I), DUMMY)
25400         M(1)=0
25500     CALL NAMLST(6,2,M(1), DUMMY)
25600     IUP=1+M(1)
25700     DO 3 I=2, IUP
25800         M(I)=0
25900     3   CALL NAMLST(6,2,M(I), DUMMY)
26000     RETURN
26100     END
26200     SUBROUTINE NAMLST(II, IO, N, X)
26300 C *****
26400 C THIS SUBROUTINE READS INPUT DATA
26500 C *****
26600     DIMENSION A(3)
26700     IF(N .NE. 0) GO TO 1
26800     READ (II,2)  A(1),A(2),A(3),N
26900     WRITE(IO,2)  A(1),A(2),A(3),N
27000     RETURN
27100     1   READ (II,3)  A(1),A(2),A(3),X
27200     WRITE(IO,3)  A(1),A(2),A(3),X
27300     RETURN
27400     2   FORMAT(1X,3A4, I4)
27500     3   FORMAT(1X,3A4,1PE13.6)
27600     END
27700     SUBROUTINE ROTATE(J)
27800 C *****
27900 C THIS SUBROUTINE TRANSFORMS VELOCITY COMPONENTS TO THE NONORTHOGONAL
28000 C COORDINATE SYSTEM
28100 C *****
28200     COMMON/VEL/  UE(40),VE(40),WE(40)
28300     COMMON/VELI/ UEI(40,40),WEI(40,40)
28400     COMMON/NDATA/ IU1, IU2, IU3, IU4, IU5, IMAX, JMAX, NXTL, NZTL, JFZ, LZ(30)
28500     DO 2 I=1, NXTL
28600     READ(IU4) H1, A, A, A, A, A, THETA, A
28700     CSCTH=1./SIN(THETA)
28800     COTTH=CSCTH*COS(THETA)
28900     UEI(I,J)=UE(I)-WE(I)*COTTH
29000     WEI(I,J)=WE(I)*CSCTH
29100     IF(J.EQ.1.OR.J.EQ.NZTL) WEI(I,J)=0.
29200     2   CONTINUE
29300     RETURN
29400     END
29500     SUBROUTINE SMOOTH(X, Y, L, JF, N, ISTART, NN, XT, IFLAG, DYDX1, II)
29600 C *****
29700 C THIS SUBROUTINE FITS A TABULATED FUNCTION WITH CUBIC SPLINES USING
29800 C LEAST-SQUARES ERROR MINIMIZATION
29900 C *****
30000     DIMENSION X(40), Y(40), L(30), XT(40)

```

```

30100      COMMON/QUAN/ A(30),B(30),PH0PH0(30),PH0PH1(30),
30200      *              PH0PS0(30),PH0PS1(30),PH1PH1(30),PH1PS0(30),
30300      *              PH1PS1(30),P1(30),P2(30),P3(30),Q1(30),
30400      *              Q2(30),Q3(30),XI(30),YPHI0(30),YPHI1(30),
30500      *              H(30)
30600      COMMON/DIMN/ JFM,NFS
30700 C    INITIATE THE COMPUTATION
30800      CALL INITL(X,Y,N,L,JF,ISTART,DYDX1)
30900      CALL SUMUP(X,Y,N,L,JF)
31000 C    COMPUTE THE P'S AND Q'S
31100      DO 1 J=2,JFM
31200      M=J-1
31300 C    COEFFICIENTS IN THE ORIGINAL EQUATIONS
31400      A1=PH0PH1(M)
31500      B1=PH1PH1(M)+PH0PH0(J)
31600      C1=PH0PH1(J)
31700      D1=PH1PS0(M)
31800      E1=PH1PS1(M)+PH0PS0(J)
31900      F1=PH0PS1(J)
32000      G1=YPHI1 (M)+YPHI0 (J)
32100      A2=3./H(M)**2
32200      B2=3.*(1./H(J)**2-1./H(M)**2)
32300      C2=-3./H(J)**2
32400      D2=1./H(M)
32500      E2=2.*(1./H(J)+1./H(M))
32600      F2=1./H(J)
32700      G2=0.
32800 C    MODIFIED COEFFICIENTS
32900      B1S=B1+A1*P2(M)+D1*Q2(M)
33000      E1S=E1+A1*P3(M)+D1*Q3(M)
33100      G1S=G1-A1*P1(M)-D1*Q1(M)
33200      B2S=B2+A2*P2(M)+D2*Q2(M)
33300      E2S=E2+A2*P3(M)+D2*Q3(M)
33400      G2S=G2-A2*P1(M)-D2*Q1(M)
33500 C    THE P'S AND Q'S
33600      DEL=1./(B1S*E2S-B2S*E1S)
33700      P1(J)=(E2S*G1S-E1S*G2S)*DEL
33800      P2(J)=(E1S*C2 -E2S*C1 )*DEL
33900      P3(J)=(E1S*F2 -E2S*F1 )*DEL
34000      Q1(J)=(B1S*G2S-B2S*G1S)*DEL
34100      Q2(J)=(B2S*C1 -B1S*C2 )*DEL
34200      Q3(J)=(B2S*F1 -B1S*F2 )*DEL
34300 1    CONTINUE
34400 C    SOLVE FOR THE A'S AND B'S
34500      K=JFM
34600      KP=JF
34700      DO 2 J=1,JFM
34800      A(K)=P1(K)+P2(K)*A(KP)+P3(K)*B(KP)
34900      B(K)=Q1(K)+Q2(K)*A(KP)+Q3(K)*B(KP)
35000      KP=K
35100      K=K-1
35200 2    CONTINUE
35300 C    INTERPOLATE TO DESIRED MESH
35400      I=1
35500      NFS=NN
35600      DO 3 J=1,NN
35700      CALL INTERP(I,J,XT(J),IFLAG,II)
35800 3    CONTINUE
35900      RETURN
36000      END

```

```

36100      SUBROUTINE SUMUP(X,Y,N,L,JF)
36200 C *****
36300 C THIS SUBROUTINE PERFORMS SUMMING OPERATIONS ON THE CUBIC-SPLINE
36400 C WEIGHTING FUNCTIONS
36500 C *****
36600      DIMENSION X(40),Y(40),L(30)
36700      COMMON/QUAN/ A(30),B(30),PH0PH0(30),PH0PH1(30),
36800      *             PH0PS0(30),PH0PS1(30),PH1PH1(30),PH1PS0(30),
36900      *             PH1PS1(30),P1(30),P2(30),P3(30),Q1(30),
37000      *             Q2(30),Q3(30),XI(30),YPHI0(30),YPHI1(30),
37100      *             H(30)
37200      COMMON/DIMN/ JFM,NFS
37300      DO 2 J=1,JFM
37400      H(J)=XI(J+1)-XI(J)
37500      KL=L(J)
37600      KU=L(J+1)
37700      DO 1 K=KL,KU
37800      DX=(X(K)-XI(J))/H(J)
37900      DXP=(XI(J+1)-X(K))/H(J)
38000      YY=Y(K)
38100      PHI1=(3.-2.*DX)*DX**2
38200      PHI0=1.-PHI1
38300      PSI1=-H(J)*DXP*DX**2
38400      PSI0= H(J)*DX*DXP**2
38500      PH0PH0(J)=PH0PH0(J)+PHI0*PHI0
38600      PH0PH1(J)=PH0PH1(J)+PHI0*PHI1
38700      PH0PS0(J)=PH0PS0(J)+PHI0*PSI0
38800      PH0PS1(J)=PH0PS1(J)+PHI0*PSI1
38900      PH1PH1(J)=PH1PH1(J)+PHI1*PHI1
39000      PH1PS0(J)=PH1PS0(J)+PHI1*PSI0
39100      PH1PS1(J)=PH1PS1(J)+PHI1*PSI1
39200      YPHI0 (J)=YPHI0 (J)+PHI0*YY
39300      YPHI1 (J)=YPHI1 (J)+PHI1*YY
39400      1  CONTINUE
39500      2  CONTINUE
39600      RETURN
39700      END

```

```

00100 C      PROGRAM EDDY3
00200 C      *****
00300 C      THREE-DIMENSIONAL BOUNDARY LAYER PROGRAM FOR FLOW ABOUT SHIP HULLS
00400 C      USING THE WILCOX-RUBESIN TWO-EQUATION MODEL OF TURBULENCE
00500 C      *****
00600      INTEGER      TAPEIN,TAPEOT,TAPEGP,TAPEPF,TAPEDT,TAPEVL
00700 C      DIMENSION  INDEX1(1189),INDEX4(1189)
00800      COMMON/BLC0/  NX,NZ,NP,NXT,NZT,NTR,NPT,NXTL,NZTL,NXSTRT,
00900      *              NZSTRT,KC,IT,IFLOW,ICHORD,ISPAN,INTDIR
01000      COMMON/BLC3/  X(81),Z(41),DETA(101),ETA(101),ETAE,VGP,CEL,BEL1,BEL2
01100      COMMON/BLC5/  F(101,2,2),U(101,2,2),V(101),G(101,2,2),W(101,2,2),
01200      *              T(101),TPROF(620),TPCF(10)
01300      COMMON/BLC9/  NSEP(41),ICASE(41)
01400      COMMON/TURB/  E(101,2,2),WT(101),WT2(101,2,2),ELT(101)
01500      COMMON/STOR/  CON,EPS,ICONVE,ITMAX,WTEDG
01600      COMMON/TAPE/  TAPEIN,TAPEOT,TAPEGP,TAPEPF,TAPEDT,TAPEVL
01700      COMMON/IPRT/  IPZ,IPX,IPRINT,EPSV,EPST
01800      COMMON/CONV/  ICOUNE,ICOUNW,INEGE,INEGW,DELV,DELT
01900      COMMON/SVNM/  IPCF,IPRF
02000      DATA IPCF,IPRF/10,620/
02100 C      CALL OPENMS(16,INDEX1,1189,0)
02200 C      CALL OPENMS(17,INDEX4,1189,0)
02300 C***     DISK FILES TAPEGP(16) AND TAPEPF(17) ARE RANDOM ACCESS FILES
02400 C           TAPEGP...BODY GEOMETRY AND VELOCITY DISTRIBUTION
02500 C           TAPEPF...PROFILES AND PRESSURES
02600 C***     DISK FILE TAPEDT(18) IS CREATED IN PROGRAM SHPM5H
02700 C***     DISK FILE TAPEVL(19) IS CREATED IN PROGRAM VELOC
02800 C***     FILES TAPEIN(5) AND TAPEOT(6) ARE CONVENTIONAL READ AND WRITE
02900 C      -----
03000      CALL INPUT
03100      CALL GRID
03200      DO 10 I=1,NZTL
03300 10      NSEP(I)=0
03400      DO 20 K=1,2
03500      DO 20 J=1,NPT
03600      F(J,K,1)=0.
03700      U(J,K,1)=0.
03800      G(J,K,1)=0.
03900      W(J,K,1)=0.
04000      E(J,K,1)=0.
04100 20      WT2(J,K,1)=0.
04200      INTDIR=IFLOW
04300      IF(IFLOW.EQ.1) WRITE(TAPEOT,690)
04400      IF(IFLOW.EQ.2) WRITE(TAPEOT,680)
04500      NZ1=NZSTRT
04600      NZ2=NZT
04700      DO 500 NX=NXSTRT,NXT
04800      NZ=NZ1
04900      IF(IFLOW.EQ.2) NZ=NZ2
05000      CALL LOGIC(NZ1,NZ2)
05100 30      IF(NX.EQ.NXSTRT) CALL PROFIL(1)
05200      IF(NSEP(NZ).NE.0) GO TO 405
05300      IT=0
05400      ICONVE=0
05500      IPRINT=0
05600      IF((NZ.EQ.NZSTRT.OR.NZ.EQ.NZT).AND.MOD(NX,IPX).EQ.0) IPRINT=1
05700      IF(NX.EQ.NXT.AND.MOD(NZ,IPZ).EQ.0) IPRINT=1
05800      IF(MOD(NZ,IPZ).EQ.0.AND.MOD(NX,IPX).EQ.0) IPRINT=1
05900      KC=2
06000      IF(ICASE(NZ).EQ.4) KC=4

```

```

06100      IF(NZ.EQ.NZT) KC=ICHORD
06200      IF(NZ.EQ.NZSTRT) KC=1
06300      IF(IPRINT.EQ.0) GO TO 290
06400      WRITE(TAPEOT,620) NX,NZ,X(NX),Z(NZ)
06500      IF(KC.EQ.1) WRITE(TAPEOT,640)
06600      IF(KC.EQ.2) WRITE(TAPEOT,650)
06700      IF(KC.EQ.4) WRITE(TAPEOT,670)
06800  290  IT=IT+1
06900      IF(IT.LE.ITMAX) GO TO 300
07000      WRITE(TAPEOT,630) NX,NZ
07100      GO TO 390
07200  300  CALL EDDY
07300      CALL CGEN
07400      CALL SOLV6
07500      CALL SOLVEW
07600      IF(V(1).LT.0.) GO TO 380
07700      IF(ABS(DELV/(V(1)+.5*DELV)).GT.EPSV.OR.
07800      *      ABS(DELT).GT.EPST) GO TO 290
07900      IF(ICONVE.EQ.0) GO TO 290
08000      IF(NP.EQ.NPT) GO TO 390
08100      IF(U(NP-1,2,2).LT..999) GO TO 370
08200      GO TO 390
08300  370  CALL PROFIL(3)
08400      IT=0
08500      GO TO 290
08600  380  WRITE(TAPEOT,720) NX,NZ
08700      CALL SHIFT2(NZ1,NZ2)
08800      GO TO 405
08900  390  CALL OUTPUT(1)
09000      IF(NP.EQ.NPT) GO TO 400
09100      CALL PROFIL(2)
09200  400  CALL SHIFT(NZ1,NZ2)
09300  405  IF(IFLOW.EQ.2) GO TO 407
09400      NZ=NZ+1
09500      IF(NZ.GT.NZ2) GO TO 410
09600      GO TO 30
09700  407  NZ=NZ-1
09800      IF(NZ.LT.NZ1) GO TO 410
09900      GO TO 30
10000  410  CALL OUTPUT(2)
10100  500  CONTINUE
10200  620  FORMAT(1H0,5H**** ,4HNX =,I3,5X4HNZ =,I3,5X3HX =,F10.3,5X3HZ =,
10300      *      F10.3)
10400  630  FORMAT(1H0,18X32HITERATIONS EXCEED ITMAX FOR NX =,I4,5X4HNZ =,I4)
10500  640  FORMAT(1H0,30H** SYMMETRY-PLANE EQUATIONS **)
10600  650  FORMAT(1H0,45H** CHORDWISE INFINITE-SWEPT HULL EQUATIONS **)
10700  670  FORMAT(1H0,18H** GENERAL CASE **)
10800  680  FORMAT(//1H0,38X50H*****----- WATERLINE CALCULATIONS STARTED -----*)
10900      ****//)
11000  690  FORMAT(//1H0,38X52H*****----- KEEL-LINE CALCULATIONS STARTED -----
11100      *****//)
11200  720  FORMAT(1H0,11X36H** BOUNDARY-LAYER SEPARATION AT NX =,I4,5X4HNZ =,
11300      *I4)
11400      END

```

```

00100      SUBROUTINE ARRYMV(LL, I, K)
00200      COMMON/BLC0/  NX, NZ, NP, NXT, NZT, NTR, NPT, NXTL, NZTL, NXSTRT,
00300      *              NZSTRT, KC, IT, IFLOW, ICHORD, ISPAN, INTDIR
00400      COMMON/BLC5/  F(101, 2, 2), U(101, 2, 2), V(101), G(101, 2, 2), W(101, 2, 2),
00500      *              T(101), TPROF(620), TPCF(10)
00600      COMMON/PRSC/  PS1(2, 2), H1(2, 2), H2(2, 2), UE(2, 2),
00700      *              WE(2, 2), CK1(2, 2), CK2(2, 2), CK12(2, 2), CK21(2, 2),
00800      *              THETA(2, 2), PFRS, UFRS, CNUFRS, UREF1, WNP
00900      COMMON/PRES/  P1(2, 2), P2(2, 2), P3(2, 2), P4(2, 2), P5(2, 2),
01000      *              P6(2, 2), P7(2, 2), P8(2, 2), P9(2, 2), P10(2, 2), P11(2, 2),
01100      *              P12(2, 2), P13(2, 2), P14(2, 2)
01200      COMMON/TURB/  E(101, 2, 2), WT(101), WT2(101, 2, 2), ELT(101)
01300      COMMON/UNIV/  J1, J2, J3, J4, J5, N1, N2, N3, N4, N5, N6, N7, N8, N9,
01400      *              N10, N11, N12, N13, N14, NPT2, NPT3, NPT4, NPT5, NPT6

```

```

01500 C -----

```

```

01600      NPT2=NPT*2
01700      NPT3=NPT*3
01800      NPT4=NPT*4
01900      NPT5=NPT*5
02000      NPT6=NPT*6
02100      GO TO(10, 30, 90, 50, 70, 85), LL
02200 C  PUT FLOW ARRAYS IN TPROF IN PREPARATION FOR A DISK WRITE
02300  10  DO 20 J=1, NPT
02400      CALL INDEX(J, NPT)
02500      TPROF(J)=F(J, I, K)
02600      TPROF(J1)=U(J, I, K)
02700      TPROF(J2)=G(J, I, K)
02800      TPROF(J3)=W(J, I, K)
02900      TPROF(J4)=E(J, I, K)
03000  20  TPROF(J5)=WT2(J, I, K)
03100      TPROF(N1)=P1(I, K)
03200      TPROF(N2)=P2(I, K)
03300      TPROF(N3)=P3(I, K)
03400      TPROF(N4)=P4(I, K)
03500      TPROF(N5)=P5(I, K)
03600      TPROF(N6)=P6(I, K)
03700      TPROF(N7)=P7(I, K)
03800      TPROF(N8)=P8(I, K)
03900      TPROF(N9)=P9(I, K)
04000      TPROF(N10)=P10(I, K)
04100      TPROF(N11)=P11(I, K)
04200      TPROF(N12)=P12(I, K)
04300      TPROF(N13)=P13(I, K)
04400      TPROF(N14)=P14(I, K)
04500      RETURN

```

```

04600 C  FILL FLOW ARRAYS AFTER A DISK READ

```

```

04700  30  DO 40 J=1, NPT
04800      CALL INDEX(J, NPT)
04900      F(J, I, K)=TPROF(J)
05000      U(J, I, K)=TPROF(J1)
05100      G(J, I, K)=TPROF(J2)
05200      W(J, I, K)=TPROF(J3)
05300      E(J, I, K)=TPROF(J4)
05400  40  WT2(J, I, K)=TPROF(J5)
05500      P1(I, K)=TPROF(N1)
05600      P2(I, K)=TPROF(N2)
05700      P3(I, K)=TPROF(N3)
05800      P4(I, K)=TPROF(N4)
05900      P5(I, K)=TPROF(N5)
06000      P6(I, K)=TPROF(N6)

```



```

06100      P7(I,K)=TPROF(N7)
06200      P8(I,K)=TPROF(N8)
06300      P9(I,K)=TPROF(N9)
06400      P10(I,K)=TPROF(N10)
06500      P11(I,K)=TPROF(N11)
06600      P12(I,K)=TPROF(N12)
06700      P13(I,K)=TPROF(N13)
06800      P14(I,K)=TPROF(N14)
06900      RETURN
07000 C   SHIFT (K,K) FLOW ARRAYS TO (I,K)
07100   50   DO 60 J=1,NPT
07200       F(J,I,K)=F(J,K,K)
07300       U(J,I,K)=U(J,K,K)
07400       G(J,I,K)=G(J,K,K)
07500       W(J,I,K)=W(J,K,K)
07600       E(J,I,K)=E(J,K,K)
07700   60   WT2(J,I,K)=WT2(J,K,K)
07800       RETURN
07900 C   SHIFT (I,K) FLOW ARRAYS TO (K,K)
08000   70   DO 80 J=1,NPT
08100       F(J,K,K)=F(J,I,K)
08200       U(J,K,K)=U(J,I,K)
08300       G(J,K,K)=G(J,I,K)
08400       W(J,K,K)=W(J,I,K)
08500       E(J,K,K)=E(J,I,K)
08600   80   WT2(J,K,K)=WT2(J,I,K)
08700       RETURN
08800 C   SHIFT (I,I) FLOW ARRAYS TO (I,K)
08900   85   DO 88 J=1,NPT
09000       F(J,I,K)=F(J,I,I)
09100       U(J,I,K)=U(J,I,I)
09200       G(J,I,K)=G(J,I,I)
09300       W(J,I,K)=W(J,I,I)
09400       E(J,I,K)=E(J,I,I)
09500   88   WT2(J,I,K)=WT2(J,I,I)
09600       RETURN
09700 C   FILL GEOMETRY ARRAYS AFTER A DISK READ
09800   90   H1(I,K)=TPCF(1)
09900       H2(I,K)=TPCF(2)
10000      CK1(I,K)=TPCF(3)
10100      CK2(I,K)=TPCF(4)
10200      CK12(I,K)=TPCF(5)
10300      CK21(I,K)=TPCF(6)
10400      THETA(I,K)=TPCF(7)
10500      PS1(I,K)=TPCF(8)
10600      UE(I,K)=TPCF(9)
10700      WE(I,K)=TPCF(10)
10800      RETURN
10900      END

```

```

00100      SUBROUTINE BELCEL
00200      COMMON/BLC0/ NX,NZ,NP,NXT,NZT,NTR,NPT,NXTL,NZTL,NXSTRT,
00300      *          NZSTRT,KC,IT,IFLOW,ICHORD,ISPAN,INTDIR
00400      COMMON/BLC3/ X(81),Z(41),DETA(101),ETA(101),ETAE,VGP,CEL,BEL1,BEL2
00500      COMMON/BLC5/ F(101,2,2),U(101,2,2),V(101),G(101,2,2),W(101,2,2),
00600      *          T(101),TPROF(620),TPCF(10)
00700      COMMON/PRES/ P1(2,2),P2(2,2),P3(2,2),P4(2,2),P5(2,2),
00800      *          P6(2,2),P7(2,2),P8(2,2),P9(2,2),P10(2,2),P11(2,2),
00900      *          P12(2,2),P13(2,2),P14(2,2)
01000 C - - - - -
01100      CALL PRESUR
01200      IF(KC.EQ.4) GO TO 20
01300      BEL1=0.
01400      BEL2=0.
01500      IF(NX.EQ.NXSTRT) GO TO 10
01600      CEL=P10(2,2)/(X(NX)-X(NX-1))
01700      RETURN
01800  10    CEL=P10(2,2)/(X(NX+1)-X(NX))
01900      IF(NTR.EQ.0) CEL=0.
02000      RETURN
02100  20    DELX=X(NX)-X(NX-1)
02200      IF(IFLOW.EQ.2) GO TO 30
02300      DELZ1=Z(NZ)-Z(NZ-1)
02400      DELZ2=Z(NZ+1)-Z(NZ)
02500      IF(NZ.GT.(NZSTRT+1)) GO TO 25
02600      DO 22 J=1,NPT
02700      G(J,2,1)=0.
02800  22    W(J,2,1)=0.
02900      GO TO 40
03000  25    IF(ICHORD.EQ.2.OR.NZ.LT.(NZT-1)) GO TO 40
03100      DO 27 J=1,NPT
03200      G(J,1,1)=0.
03300  27    W(J,1,1)=0.
03400      GO TO 40
03500  30    DELZ1=Z(NZ)-Z(NZ+1)
03600      DELZ2=Z(NZ-1)-Z(NZ)
03700      IF(ICHORD.EQ.2.OR.NZ.LT.(NZT-1)) GO TO 35
03800      DO 32 J=1,NPT
03900      G(J,2,1)=0.
04000  32    W(J,2,1)=0.
04100      GO TO 40
04200  35    IF(NZ.GT.(NZSTRT+1)) GO TO 40
04300      DO 37 J=1,NPT
04400      G(J,1,1)=0.
04500  37    W(J,1,1)=0.
04600  40    CEL=P10(2,2)/DELX
04700      BEL1=.5*P7(2,2)/DELZ1
04800      BEL2=.5*P7(2,2)/DELZ2
04900      RETURN
05000      END

```

```

00100      SUBROUTINE CGEN
00200      COMMON/BLC0/ NX,NZ,NP,NXT,NZT,NTR,NPT,NXTL,NZTL,NXSTRT,
00300      *          NZSTRT,KC,IT,IFLOW,ICHORD,ISPAN,INTDIR
00400      COMMON/BLC3/ X(81),Z(41),DETA(101),ETA(101),ETAE,VGP,CEL,BEL1,BEL
00500      COMMON/BLC5/ F(101,2,2),U(101,2,2),V(101),G(101,2,2),W(101,2,2),
00600      *          T(101),TPROF(620),TPCF(10)
00700      COMMON/PRSC/ PS1(2,2),H1(2,2),H2(2,2),UE(2,2),
00800      *          WE(2,2),CK1(2,2),CK2(2,2),CK12(2,2),CK21(2,2),
00900      *          THETA(2,2),PFRS,UFRS,CNUFRS,UREF1,WNP
01000      COMMON/BLC7/ PU(101),PW(101),QU(101),QW(101)
01100      COMMON/PRES/ P1(2,2),P2(2,2),P3(2,2),P4(2,2),P5(2,2),
01200      *          P6(2,2),P7(2,2),P8(2,2),P9(2,2),P10(2,2),P11(2,2),
01300      *          P12(2,2),P13(2,2),P14(2,2)
01400      COMMON/BLCE/ EDV(101)
01500      COMMON/BLCY/ Y1(101),Y3(101),W1,W2,W3,W4
01600 C - - - - -
01700      IF(IT.EQ.1) CALL BELCEL
01800      U(NP,2,2)=1.
01900      W(NP,2,2)=WE(2,2)/UREF1
02000      IF(KC.EQ.1) W(NP,2,2)=WNP
02100      PU(1)=0.
02200      PW(1)=0.
02300      QU(1)=0.
02400      QW(1)=0.
02500      NPM=NP-1
02600      BM=1.+5*(EDV(1)+EDV(2))
02700      DO 50 J=2,NPM
02800      BP=1.+5*(EDV(J)+EDV(J+1))
02900      UB=U(J,2,2)
03000      WB=W(J,2,2)
03100      E2=P1(2,2)*F(J,2,2)+P6(2,2)*G(J,2,2)
03200      DFB=CEL*(F(J,2,2)-F(J,1,2))
03300      DGB=BEL1*(G(J,2,2)-G(J,2,1))+BEL2*(G(J,1,1)-G(J,1,2))
03400      USB=UB*UB
03500      WSB=WB*WB
03600      DVB=(E2+DFB+DGB)/(DETA(J)+DETA(J-1))
03700      X1=-P2(2,2)*UB+P5(2,2)*WB
03800      X2=-P4(2,2)*UB+P3(2,2)*WB
03900      A1=BM*Y3(J)-DVB
04000      B1=-BP*Y1(J)-BM*Y3(J)+X1-(CEL*UB+BEL1*WB)
04100      C1=BP*Y1(J)+DVB
04200      D1=P8(2,2)*WSB-P11(2,2)-(CEL*UB+BEL2*WB)*U(J,1,2)
04300      *          -BEL1*WB*U(J,2,1)+BEL2*WB*U(J,1,1)
04400      B2=-BP*Y1(J)-BM*Y3(J)+X2-(CEL*UB+BEL1*WB)
04500      D2=P9(2,2)*USB-P12(2,2)-(CEL*UB+BEL2*WB)*W(J,1,2)
04600      *          -BEL1*WB*W(J,2,1)+BEL2*WB*W(J,1,1)
04700      B1S=B1+A1*QU(J-1)
04800      B2S=B2+A1*QW(J-1)
04900      D1S=D1-A1*PU(J-1)
05000      D2S=D2-A1*PW(J-1)
05100      PU(J)=D1S/B1S
05200      PW(J)=D2S/B2S
05300      QU(J)=-C1/B1S
05400      QW(J)=-C1/B2S
05500      50  BM=BP
05600      RETURN
05700      END

```

```

00100      SUBROUTINE EDDY
00200      COMMON/BLC0/  NX,NZ,NP,NXT,NZT,NTR,NPT,NXTL,NZTL,NXSTRT,
00300      *              NZSTRT,KC,IT,IFLOW,ICHORD,ISPAN,INTDIR
00400      COMMON/BLC5/  F(101,2,2),U(101,2,2),V(101),G(101,2,2),W(101,2,2),
00500      *              T(101),TPROF(620),TPCF(10)
00600      COMMON/PRSC/  PS1(2,2),H1(2,2),H2(2,2),UE(2,2),
00700      *              WE(2,2),CK1(2,2),CK2(2,2),CK12(2,2),CK21(2,2),
00800      *              THETA(2,2),PFRS,UFRS,CNUFRS,UREF1,WNP
00900      COMMON/BLCE/  EDV(101)
01000      COMMON/TURB/  E(101,2,2),WT(101),WT2(101,2,2),ELT(101)
01100      COMMON/STOR/  CON,EPS,ICONVE,ITMAX,WTEDG
01200 C  -----
01300      REY=UE(2,2)*PS1(2,2)/CNUFRS
01400      CEP=REY*(UREF1/UE(2,2))**2
01500      EDV(1)=0.
01600      ELT(1)=0.
01700      DO 54 J=2,NP
01800      IF(IT.EQ.1) WT(J)=SQRT(WT2(J,2,2))
01900      RET=CEP*E(J,2,2)/WT(J)
02000      EDV(J)=RET*(1.-CON*EXP(-RET))
02100      54  ELT(J)=SQRT(E(J,2,2))/WT(J)
02200      RETURN
02300      END

```

```

00100      SUBROUTINE GRID
00200      INTEGER TAPEIN, TAPEOT, TAPEGP, TAPEPF, TAPEDT, TAPEVL
00300      COMMON/BLC0/ NX, NZ, NP, NXT, NZT, NTR, NPT, NXTL, NZTL, NXSTRT,
00400      *          NZSTRT, KC, IT, IFLOW, ICHORD, ISPAN, INTDIR
00500      COMMON/BLC3/ X(81), Z(41), DETA(101), ETA(101), ETAE, VGP, CEL, BEL1, BEL
00600      COMMON/BLCY/ Y1(101), Y3(101), W1, W2, W3, W4
00700      COMMON/TAPE/ TAPEIN, TAPEOT, TAPEGP, TAPEPF, TAPEDT, TAPEVL
00800 C - - - - -
00900      ETA(1)=0.
01000      NP=ALOG((ETAE/DETA(1))*(VGP-1.)+1.)/ALOG(VGP)+1.0001
01100      IF(NP.LE.NPT) GO TO 30
01200      WRITE(TAPEOT,50)
01300      STOP 10
01400 30   DO 40 J=2,NPT
01500      DETA(J)=VGP*DETA(J-1)
01600      ETA(J)=ETA(J-1)+DETA(J-1)
01700      C1=2./((DETA(J)+DETA(J-1)))
01800      Y1(J)=C1/DETA(J)
01900 40   Y3(J)=C1/DETA(J-1)
02000      ETAE=ETA(NP)
02100      W5=(1.+VGP)*(1.+VGP*(1.+VGP))*VGP**3
02200      DEN=1./(W5*DETA(1))
02300      W4=(1.+VGP)*DEN
02400      W3=(1.+VGP*(1.+VGP))*2*DEN
02500      W2=VGP*(1.+VGP)*W3
02600      W1=(1.+VGP+(1.+VGP*(1.+VGP))*2*(VGP*(1.+VGP)-1.))*DEN
02700      RETURN
02800 50   FORMAT(1H0,37HNP EXCEEDED NPT -- PROGRAM TERMINATED)
02900      END

```

```

00100      SUBROUTINE INDEX(J,NPT)
00200      COMMON/UNIV/ J1,J2,J3,J4,J5,N1,N2,N3,N4,N5,N6,N7,N8,N9,
00300      *           N10,N11,N12,N13,N14,NPT2,NPT3,NPT4,NPT5,NPT6
00400 C -----
00500      J1=J+NPT
00600      J2=J+NPT2
00700      J3=J+NPT3
00800      J4=J+NPT4
00900      J5=J+NPT5
01000      IF(J.LT.NPT) RETURN
01100      N1=NPT6+1
01200      N2=N1+1
01300      N3=N2+1
01400      N4=N3+1
01500      N5=N4+1
01600      N6=N5+1
01700      N7=N6+1
01800      N8=N7+1
01900      N9=N8+1
02000      N10=N9+1
02100      N11=N10+1
02200      N12=N11+1
02300      N13=N12+1
02400      N14=N13+1
02500      RETURN
02600      END

```

```

00100      SUBROUTINE INPUT
00200      DIMENSION GMPKG(8),TITLE(9),APC(10)
00300      INTEGER TAPEIN,TAPEOT,TAPEGP,TAPEPF,TAPEDT,TAPEVL
00400      COMMON/BLC0/  NX,NZ,NP,NXT,NZT,NTR,NPT,NXTL,NZTL,NXSTRT,
00500      *              NZSTRT,KC,IT,IFLOW,ICHORD,ISPAN,INTDIR
00600      COMMON/BLC3/  X(81),Z(41),DETA(101),ETA(101),ETAE,VGP,BEL1,BEL
00700      COMMON/CHRD/  WNP1(81),WNPN(81)
00800      COMMON/PRSC/  PS1(2,2),H1(2,2),H2(2,2),UE(2,2),
00900      *              WE(2,2),CK1(2,2),CK2(2,2),CK12(2,2),CK21(2,2),
01000      *              THETA(2,2),PFRS,UFRS,CNUFRS,UREF1,WNP
01100      COMMON/TAPE/  TAPEIN,TAPEOT,TAPEGP,TAPEPF,TAPEDT,TAPEVL
01200      COMMON/IPRT/  IPZ,IPX,IPRINT,EPSV,EPST
01300      COMMON/STOR/  CON,EPS,ICONVE,ITMAX,WTEDG
01400      COMMON/CONS/  SALFA,SALFAS,SBETA,SBETAS,SLAMDA,SSIGMA,SSGMAS,USTOP
01500      *              ZIOTAE,ZIOTAL,RW2
01600      COMMON/SVNM/  IPCF,IPRF
01700      COMMON/SHRT/  ISHORT
01800      COMMON/RELX/  RFTRB,RFVEL
01900      COMMON/INTG/  ALAMI(41),CFXI(41),CFZI(41),DELTAI(41),
02000      *              THETXI(41),THETZI(41)
02100      EQUIVALENCE (APC(1),GMPKG(1)),(APC(9),UEUF),(APC(10),WEUF)
02200      DATA  TAPEIN/5/,TAPEOT/6/,TAPEGP/16/,TAPEPF/17/,TAPEDT/18/,
02300      *      TAPEVL/19/,ISHORT/0/,RFTRB/.8/,RFVEL/1./,NPT/101/
02400      DATA  SALFA/1.11111/,SALFAS/.3/,SBETA/.15/,SBETAS/.09/,
02500      *      SLAMDA/.091/,SSIGMA/.5/,SSGMAS/.5/,USTOP/4./,
02600      *      ZIOTAE/3.75E-5/,ZIOTAL/.09/,RW2/4./
02700      DATA  EPS,EPSV,EPST/3*.01/,ETAE/8./,ICHORD/1/,IFLOW/1/,
02800      *      IFPRNT/0/,ISPAN/1/,ITMAX/20/,NTR/1/
02900      NAMELIST/NAME/  DETA  ,  EPS    ,  EPST   ,  EPSV   ,  ETAE   ,
03000      *              ICHORD ,  IFLOW  ,  IFPRNT ,  IPX    ,  IPZ    ,
03100      *              ISPAN  ,  ITMAX  ,  NPT    ,  NTR    ,  NXSTRT ,
03200      *              NXT    ,  NZSTRT ,  NZT    ,  TAPEIN ,  TAPEOT ,
03300      *              TAPEGP ,  TAPEPF ,  TAPEDT ,  TAPEVL ,  VGP    ,
03400      *              ISHORT ,  RFTRB  ,  RFVEL
03500      NAMELIST/DATA/  SALFA  ,  SALFAS ,  SBETA  ,  SBETAS ,  SLAMDA ,
03600      *              SSIGMA ,  SSGMAS ,  USTOP  ,  ZIOTAE ,  ZIOTAL ,
03700      *              RW2
03800      NAMELIST/STRT/  ALAMI  ,  CFXI   ,  CFZI   ,  DELTAI ,  THETXI ,
03900      *              THETZI
04000  C  - - - - -
04100      READ(TAPEIN,120) TITLE
04200      READ(TAPEIN,NAME)
04300      READ(TAPEIN,DATA)
04400      READ(TAPEIN,STRT)
04500      READ(TAPEIN,170) UFRS,PFRS,CNUFRS
04600      READ(TAPEDT) NXTL,NZTL
04700      READ(TAPEDT) (X(I),I=1,NXTL)
04800      READ(TAPEDT) (Z(I),I=1,NZTL)
04900      CON=1.-SLAMDA**2
05000      IF(IFPRNT.EQ.1) WRITE(TAPEOT,210)
05100      NX=NXSTRT
05200      IF(IFLOW.EQ.1) NZ=NZSTRT
05300      IF(IFLOW.EQ.2) NZ=NZT
05400      UREF1=UFRS
05500      IPNTG=0
05600      DO 70 K=1,NZTL
05700      DO 60 I=1,NXTL
05800      IPNTG=IPNTG+1
05900      READ(TAPEDT) GMPKG
06000      READ(TAPEVL) UEUF,WEUF

```

```

06100      PE=PFRS
06200      UEUF=UEUF*UFRS
06300      WEUF=WEUF*UFRS
06400      IF(IFPRNT.EQ.1) WRITE(TAPEOT,220) K,I,GMPKG
06500      CALL STORIT(TAPEGP,APC,IPCF,IPNTG,0)
06600      IF(I.NE.NXSTRT) GO TO 60
06700      IF(K.NE.NZ) GO TO 60
06800      H1(2,2)=APC(1)
06900      H2(2,2)=APC(2)
07000      CK1(2,2)=APC(3)
07100      CK2(2,2)=APC(4)
07200      CK12(2,2)=APC(5)
07300      CK21(2,2)=APC(6)
07400      THETA(2,2)=APC(7)
07500      PS1(2,2)=APC(8)
07600      UE(2,2)=APC(9)
07700      WE(2,2)=APC(10)
07800      60  CONTINUE
07900      70  CONTINUE
08000      DO 75 I=1,NXTL
08100      75  READ(TAPEVL) WNP1(I),WNP1(I)
08200      IF(IFPRNT.EQ.0) GO TO 100
08300      REWIND TAPEVL
08400      WRITE(6,230)
08500      DO 90 K=1,NZTL
08600      WRITE(TAPEOT,240) Z(K)
08700      DO 80 I=1,NXTL
08800      READ(TAPEVL) UEUF,WEUF
08900      UEUF=UEUF*UFRS
09000      WEUF=WEUF*UFRS
09100      WRITE(TAPEOT,250) I,X(I),UEUF,WEUF
09200      80  CONTINUE
09300      90  CONTINUE
09400      WRITE(TAPEOT,252)
09500      DO 95 I=1,NXTL
09600      95  WRITE(TAPEOT,250) I,X(I),WNP1(I),WNP1(I)
09700      WRITE(TAPEOT,255)
09800      100 WRITE(TAPEOT,180)
09900      WRITE(TAPEOT,190) TITLE
10000      WRITE(TAPEOT,200) NXSTRT,NZSTRT,NXT,NZT,NTR,IFLOW,ICHORD,
10100      *      IFPRNT,ISPAN,ITMAX,IPZ,IPX,VGP,ETA,
10200      *      DETA(1),UFRS,PFRS,CNUFRS,EPS,EPV,EPST
10300      WRITE(TAPEOT,260) SALFA,SALFAS,SBETA,SBETAS,SSIGMA,SSGMAS,
10400      *      SLAMDA,RW2,USTOP,ZIOTAE,ZIOTAL
10500      WRITE(TAPEOT,255)
10600      RETURN
10700      120 FORMAT(9A6)
10800      170 FORMAT(1P3E12.4)
10900      180 FORMAT(50X30HTHREE-D BOUNDARY-LAYER PROGRAM/1H ,
11000      *      47X36HFOR TURBULENT FLOW ABOUT A SHIP HULL)
11100      190 FORMAT(////40X,9A6)
11200      200 FORMAT(///1H0,32X8HNXSTRT= ,I3,15X8HNZSTRT= ,I3,15X8HNXT = ,I3
11300      *      /1H0,32X8HNZT = ,I3,15X8HNTR = ,I3,15X8HIFLOW = ,I3
11400      *      /1H0,32X8HICHORD= ,I3,15X8HIFPRNT= ,I3,15X8HISPAN = ,I3
11500      *      /1H0,32X8HITMAX = ,I3,15X8HIPZ = ,I3,15X8HIPX = ,I3
11600      *      /1H0,32X7HVGP =,1PE14.6,5X7HETA =,E14.6,5X7HDETA =,E14.6
11700      *      /1H0,32X7HUFRS =,E14.6,5X7HPFRS =,E14.6,5X7HNUFRS =,E14.6
11800      *      /1H0,32X7HEPS =,E14.6,5X7HEPSV =,E14.6,5X7HEPST =,E14.6)
11900      210 FORMAT(1X,25H** INPUT HULL GEOMETRY **//2X2HNZ,1X2HNX,5X2HH1,
12000      *      10X2HH2,10X2HK1,10X2HK2,10X3HK12,9X3HK21,8X5HTHETA,8X2HS1/)

```



```

12100 220  FORMAT(1H ,2I3,1P8E12.4)
12200 230  FORMAT(1H1,45H** EXTERNAL VELOCITY DISTRIBUTION (FT/SEC) **/
12300      *      1H0,5X1HZ,13X1HX,13X2HUE,12X2HWE/)
12400 240  FORMAT(1H ,1PE12.5)
12500 250  FORMAT(1H ,7X,I3,2X,1P3E14.5)
12600 252  FORMAT(//1H0,19X1HX,12X4HWNP1,11X4HWNP1/)
12700 255  FORMAT(1H1)
12800 260  FORMAT(/////45X48H**WILCOX-RUBESIN TWO-EQUATION TURBULENCE MODEL*
12900      *      ///45X29H THE CLOSURE COEFFICIENTS ARE: //47X8HALPHA  =,1PE12.3
13000      *      5X8HALPHA* =,E12.3//47X8HBETA   =,E12.3,5X8HBETA* =,E12.3/
13100      *      47X8HSIGMA =,E12.3,5X8HSIGMA* =,E12.3//47X8HLAMBDA =,E12.3,
13200      *      5X8HRW2    =,E12.3//
13300      *      /36X54H DISSIPATION RATE PRESCRIBED ANALYTICALLY UP TO UPLUS
13400      *      E12.4//45X24H INITIAL EDGE CONDITIONS://
13500      *      47X8HIOTAE =,E12.4,5X8HIOTAL  =,E12.4)
13600      END

```

```

00100      SUBROUTINE LOGIC(NZ1,NZ2)
00200      INTEGER TAPEIN,TAPEOT,TAPEGP,TAPEPF,TAPEDT,TAPEVL
00300      COMMON/BLC0/ NX,NZ,NP,NXT,NZT,NTR,NPT,NXTL,NZTL,NXSTRT,
00400      *          NZSTRT,KC,IT,IFLOW,ICHORD,ISPAN,INTDIR
00500      COMMON/BLC9/ NSEP(41),ICASE(41)
00600      COMMON/TAPE/ TAPEIN,TAPEOT,TAPEGP,TAPEPF,TAPEDT,TAPEVL
00700 C -----
00800      NZ1P=NZ1+1
00900      NZ2M=NZ2-1
01000      IF(NX.NE.NXSTRT) GO TO 20
01100      DO 10 I=NZ1P,NZ2M
01200 10     ICASE(I)=1
01300      IF(IFLOW.EQ.2) GO TO 15
01400      ICASE(NZ1)=1
01500      ICASE(NZ2)=2
01600      RETURN
01700 15     ICASE(NZ1)=2
01800      ICASE(NZ2)=1
01900      RETURN
02000 20     IF(IFLOW.EQ.2) GO TO 40
02100      ICASE(NZ1)=3
02200      ICASE(NZ2)=5
02300      DO 30 I=NZ1P,NZ2M
02400      ICASE(I)=4
02500      IF(NSEP(I+1).NE.0) ICASE(I)=6
02600      IF(NSEP(I-1).NE.0) ICASE(I)=3
02700      IF(NSEP(I+1).NE.0.AND.NSEP(I-1).NE.0) GO TO 25
02800      GO TO 30
02900 25     IF(NSEP(I).NE.0) GO TO 30
03000      NSEP(I)=1
03100      WRITE(TAPEOT,100) I,NX
03200 30     CONTINUE
03300      RETURN
03400 40     ICASE(NZ2)=3
03500      ICASE(NZ1)=5
03600      DO 50 I=NZ1P,NZ2M
03700      ICASE(I)=4
03800      IF(NSEP(I-1).NE.0) ICASE(I)=6
03900      IF(NSEP(I+1).NE.0) ICASE(I)=3
04000      IF(NSEP(I+1).NE.0.AND.NSEP(I-1).NE.0) GO TO 45
04100      GO TO 50
04200 45     IF(NSEP(I).NE.0) GO TO 50
04300      NSEP(I)=1
04400      WRITE(TAPEOT,100) I,NX
04500 50     CONTINUE
04600      RETURN
04700 100    FORMAT(/9X26H** CALCULATIONS ALONG NZ =,I4,2X1B)INTERMINATED AT NX =
04800      *          ,I4/9X40H  BECAUSE OF MERGING SEPARATION REGIONS//)
04900      END

```

```

00100      SUBROUTINE OUTPUT(LL)
00200      DIMENSION Y(101),SUMTBL(10,41),CVC(101),BETA(101),USUSE(101)
00300      DIMENSION WPRNT(101),ELPRNT(101),SVC(101)
00400      INTEGER TAPEIN,TAPEOT,TAPEGP,TAPEPF,TAPEDT,TAPEVL
00500      COMMON/BLC0/  NX,NZ,NP,NXT,NZT,NTR,NPT,NXTL,NZTL,NXSTRT,
00600      *              NZSTRT,KC,IT,IFLOW,ICHORD,ISPAN,INTDIR
00700      COMMON/BLC3/  X(81),Z(41),DETA(101),ETA(101),ETAE,VGF,CEL,BEL1,BEL
00800      COMMON/BLC5/  F(101,2,2),U(101,2,2),V(101),G(101,2,2),W(101,2,2),
00900      *              T(101),TPROF(620),TPCF(10)
01000      COMMON/BLC9/  NSEP(41),ICASE(41)
01100      COMMON/PRSC/  PS1(2,2),H1(2,2),H2(2,2),UE(2,2),
01200      *              WE(2,2),CK1(2,2),CK2(2,2),CK12(2,2),CK21(2,2),
01300      *              THETA(2,2),PFRS,UFRS,CNUFRS,UREF1,WNP
01400      COMMON/BLCE/  EDV(101)
01500      COMMON/CONS/  SALFA,SALFAS,SBETA,SBETAS,SLAMDA,SSIGMA,SSGMAS,USTOP.
01600      *              ZIOTAE,ZIOTAL,RW2
01700      COMMON/STOR/  CON,EPS,ICONVE,ITMAX,WTEDG
01800      COMMON/TURB/  E(101,2,2),WT(101),WT2(101,2,2),ELT(101)
01900      COMMON/DERV/  DUDX,DWDX,DUDZ,DWDZ
02000      COMMON/IPRT/  IPZ,IPX,IPRINT,EPSV,EPST
02100      COMMON/TAPE/  TAPEIN,TAPEOT,TAPEGP,TAPEPF,TAPEDT,TAPEVL
02200  C - - - - -
02300      GO TO (10,220), LL
02400  10  USE=SQRT(UE(2,2)**2+WE(2,2)**2+2.*UE(2,2)*WE(2,2)*
02500      *      COS(THETA(2,2)))
02600      DUSED5=((UE(2,2)+WE(2,2)*COS(THETA(2,2)))*(UE(2,2)/H1(2,2)*
02700      *      DUDX+WE(2,2)/H2(2,2)*DUDZ)+(WE(2,2)+UE(2,2)*
02800      *      COS(THETA(2,2)))*(UE(2,2)/H1(2,2)*DWDX+WE(2,2)/H2(2,2)*
02900      *      DWDZ)+UE(2,2)*WE(2,2)*SIN(THETA(2,2))*(UE(2,2)*
03000      *      ((CK12(2,2)+CK21(2,2)*COS(THETA(2,2)))/SIN(THETA(2,2))+
03100      *      CK1(2,2))+WE(2,2)*((CK21(2,2)+CK12(2,2)*COS(THETA(2,2)))/
03200      *      SIN(THETA(2,2))+CK2(2,2)))/USE**2
03300      URUE=UREF1/UE(2,2)
03400      US1=(UE(2,2)/USE)**2
03500      US2=(UREF1/USE)**2
03600      US3=2.*UE(2,2)*UREF1/USE**2*COS(THETA(2,2))
03700      B1=URUE*SIN(THETA(2,2))
03800      B2=URUE*COS(THETA(2,2))
03900      RX=UE(2,2)*PS1(2,2)/CNUFRS
04000      TRCMN=SQRT(CNUFRS*PS1(2,2)/UE(2,2))
04100      WSINTH=B1/URUE
04200      IF(KC.EQ.1) WSINTH=0.
04300      NPYES=0
04400      NPM03=(NP-1)/3
04500      ENPM03=(NP-1.)/3.
04600      DIFFNP=ENPM03-NPM03
04700      IF(DIFFNP.GT..1) NPYES=1
04800      DO 20 J=1,NP
04900  20  Y(J)=TRCMN*ETA(J)
05000      CVCT=UREF1/UE(2,2)*COS(THETA(2,2))
05100      CVCD=(1.+WE(2,2)/UE(2,2)*COS(THETA(2,2)))*URUE
05200      DO 80 J=1,NP
05300      USUSE(J)=SQRT(US1*U(J,2,2)**2+US2*W(J,2,2)**2+US3*U(J,2,2)*
05400      *      W(J,2,2))
05500      BETA(J)=0.
05600      IF(J.GT.1.AND.KC.NE.1)
05700      *      BETA(J)=57.29578*ATAN(B1*W(J,2,2)/(U(J,2,2)+B2*W(J,2,2)))
05800      SVC(J)=WSINTH*W(J,2,2)
05900  80  CVC(J)=(U(J,2,2)+W(J,2,2)*CVCT)/CVCD
06000      IF(KC.NE.1) BETA(1)=57.29578*ATAN(B1*T(1)/(U(1)+B2*T(1)))

```

```

06100      IF(IPRINT.EQ.0) GO TO 111
06200      WRITE(TAPEOT,280)
06300      WSCALE=US1/(US2*RX)
06400      ELSCAL=URUE*PS1(2,2)
06500      WPRNT(1)=9.99999E+37
06600      ELPRNT(1)=0.
06700      DO 84 J=2,NP
06800      WPRNT(J)=WT(J)*WSCALE
06900 84      ELPRNT(J)=ELT(J)*ELSCAL
07000      WRITE(TAPEOT,290) (J,Y(J),CVC(J),SVC(J),USUSE(J),BETA(J),
07100      *           E(J,2,2),WPRNT(J),ELPRNT(J),EDV(J), J=1,NP,3)
07200      IF(NPYES.EQ.0) GO TO 111
07300      WRITE(TAPEOT,290) NP,Y(NP),CVC(NP),SVC(NP),USUSE(NP),BETA(NP),
07400      *           E(NP,2,2),WPRNT(NP),ELPRNT(NP),EDV(NP)
07500 111    BETA(NP)=BETA(NP)/57.29578
07600      CTRM=2./SQRT(RX)
07700      UEUF=UE(2,2)/UFRS
07800      S1SORX=PS1(2,2)/SQRT(RX)
07900      COSTH=COS(THETA(2,2))
08000      CTRM2=UE(2,2)*UREF1/UFRS**2*T(1)*COSTH
08100      CFC=CTRM*(UEUF**2*V(1)+CTRM2)
08200      CFN=CTRM*UE(2,2)*UREF1/UFRS**2*T(1)*SIN(THETA(2,2))
08300      CI2=0.
08400      CI3=0.
08500      CI4=0.
08600      WEUECT=WE(2,2)/UE(2,2)*COSTH
08700      URUECT=UREF1/UE(2,2)*COSTH
08800      C2=(U(1,2,2)+URUECT*W(1,2,2))**2
08900      C3=W(1,2,2)**2
09000      C4=U(1,2,2)**2
09100      DO 170 J=2,NP
09200      C22=(U(J,2,2)+URUECT*W(J,2,2))**2
09300      C33=W(J,2,2)**2
09400      C44=U(J,2,2)**2
09500      CI2=CI2+.5*(C2+C22)*DETA(J-1)
09600      CI3=CI3+.5*(C3+C33)*DETA(J-1)
09700      CI4=CI4+.5*(C4+C44)*DETA(J-1)
09800      C2=C22
09900      C3=C33
10000 170   C4=C44
10100      TNUM=F(NP,2,2)+URUECT*G(NP,2,2)
10200      TDEN=1.+WEUECT
10300      DLSTS=S1SORX*(ETA(NP)-(F(NP,2,2)+UREF1/UE(2,2)*G(NP,2,2)
10400      *           *COSTH)/TDEN)
10500      THTAS=S1SORX*(TNUM/TDEN-CI2/TDEN**2)
10600      IF(KC.GT.1) GO TO 180
10700      CFC=CTRM*UEUF**2*V(1)
10800      CFN=0.
10900      DLSTS=S1SORX*(ETA(NP)-F(NP,2,2))
11000      THTAS=S1SORX*(F(NP,2,2)-CI4)
11100 180   IF(ABS(WE(2,2)).GT.1.E-8) GO TO 200
11200      DLSTN=0.
11300      THTAN=0.
11400      GO TO 210
11500 200   DLSTN=S1SORX*(ETA(NP)-UREF1/WE(2,2)*G(NP,2,2))
11600      THTAN=S1SORX*UREF1/WE(2,2)*(G(NP,2,2)-UREF1/WE(2,2)*CI3)
11700 210   IF(IPRINT.EQ.1) WRITE(TAPEOT,300) CFC,DLSTS,THTAS,BETA(1),CFN,
11800      *           DLSTN,THTAN,RX,USE,DUSED5
11900      SUMTBL(1,NZ)=V(1)
12000      SUMTBL(2,NZ)=T(1)

```

```

12100      SUMTBL(3,NZ)=CFC
12200      SUMTBL(4,NZ)=CFN
12300      SUMTBL(5,NZ)=BETA(1)
12400      SUMTBL(6,NZ)=DLSTS
12500      SUMTBL(7,NZ)=DLSTN
12600      SUMTBL(8,NZ)=THTAS
12700      SUMTBL(9,NZ)=THTAN
12800      SUMTBL(10,NZ)=RX
12900      RETURN
13000  220  WRITE(TAPEOT,310) X(NX)
13100      DO 240 I=NZSTRT,NZT
13200      IF(NSEP(I).EQ.0) GO TO 230
13300      WRITE(TAPEOT,340) I,Z(I)
13400      GO TO 240
13500  230  WRITE(TAPEOT,320) I,Z(I),(SUMTBL(J,I),J=1,10)
13600  240  CONTINUE
13700      WRITE(TAPEOT,330)
13800      RETURN
13900  C  -----
14000  280  FORMAT(1H0,2X1HJ,4X1HY,10X4HU/UR,9X4HW/UR,9X4HQ/QE,9X4HBETA,
14100      *      BX7HE/UR**2,6X5HOMEGA,9X2HEL,11X3HEPS)
14200  290  FORMAT(1H ,I3,1PE10.3,8E13.4)
14300  300  FORMAT(2X6HCFC  =,1PE14.6,3X6HDLSTS=,E14.6,3X6HTHTAS=,E14.6,
14400      *      3X6HBETA1=,E14.6/2X6HCFN  =,E14.6,3X6HDLSTN=,E14.6,
14500      *      3X6HTHTAN=,E14.6,3X6HRX   =,E14.6/2X6HQE   =,E14.6,
14600      *      3X6HDQEDX=,E14.6/1H0,62(2H*-))
14700  310  FORMAT(1H0,35X28H****-- SUMMARY TABLE FOR X =,1PE10.2,
14800      *      1X7H -- ****/1H0,3H NZ,5X1HZ,
14900      *      8X5HVWALL,6X5HTWALL,7X3HCFC,8X3HCFN,7X4HBETA,
15000      *      7X5HDLSTS,6X5HDLSTN,6X5HTHTAS,6X5HTHTAN,7X2HRX)
15100  320  FORMAT(1H ,I3,1PE10.3,10E11.3)
15200  330  FORMAT(1H0,62(2H*-)/)
15300  340  FORMAT(1H ,I3,1PE10.3,3X17H*** SEPARATED ***)
15400      END

```

```

00100      SUBROUTINE PRESUR
00200      INTEGER TAPEIN, TAPEOT, TAPEGP, TAPEPF, TAPEDT, TAPEVL
00300      DIMENSION APC(10)
00400      COMMON/BLCO/  NX,NZ,NP,NXT,NZT,NTR,NPT,NXTL,NZTL,NXSTRT,
00500      *              NZSTRT,KC,IT,IFLOW,ICHORD,ISPAN,INTDIR
00600      COMMON/BLC3/  X(81),Z(41),DETA(101),ETA(101),ETAE,VGP,CEL,BEL1,BEL2
00700      COMMON/CHRD/  WNP1(81),WNPN(81)
00800      COMMON/PRSC/  PS1(2,2),H1(2,2),H2(2,2),UE(2,2),
00900      *              WE(2,2),CK1(2,2),CK2(2,2),CK12(2,2),CK21(2,2),
01000      *              THETA(2,2),PFRS,UFRS,CNUFRS,UREF1,WNP
01100      COMMON/PRES/  P1(2,2),P2(2,2),P3(2,2),P4(2,2),P5(2,2),
01200      *              P6(2,2),P7(2,2),P8(2,2),P9(2,2),P10(2,2),P11(2,2),
01300      *              P12(2,2),P13(2,2),P14(2,2)
01400      COMMON/DERV/  DUDX,DWDX,DUDZ,DWDZ
01500      COMMON/TAPE/  TAPEIN,TAPEOT,TAPEGP,TAPEPF,TAPEDT,TAPEVL
01600      COMMON/IPRT/  IPZ,IPX,IPRINT,EPSV,EPST
01700      COMMON/SVNM/  IPCF,IPRF
01800  C  - - - - -
01900      IF(NZ.EQ.NZT) GO TO 20
02000      IF(NZ.GT.NZSTRT) GO TO 10
02100      ICNT=(NZ-1)*NXTL+NX
02200      GO TO 30
02300  10  ICNT=(NZ-2)*NXTL+NX
02400      GO TO 30
02500  20  ICNT=(NZ-3)*NXTL+NX
02600  30  IPNT2=ICNT
02700      CALL STORIT(TAPEGP,APC,IPCF,IPNT2,1)
02800      H11=APC(1)
02900      H21=APC(2)
03000      THETA1=APC(7)
03100      S11=APC(8)
03200      UE1=APC(9)
03300      WE1=APC(10)
03400      ICNT=ICNT+NXTL
03500      IPNT2=ICNT
03600      CALL STORIT(TAPEGP,APC,IPCF,IPNT2,1)
03700      H12=APC(1)
03800      H22=APC(2)
03900      THETA2=APC(7)
04000      S12=APC(8)
04100      UE2=APC(9)
04200      WE2=APC(10)
04300      ICNT=ICNT+NXTL
04400      IPNT2=ICNT
04500      CALL STORIT(TAPEGP,APC,IPCF,IPNT2,1)
04600      H13=APC(1)
04700      H23=APC(2)
04800      THETA3=APC(7)
04900      S13=APC(8)
05000      UE3=APC(9)
05100      WE3=APC(10)
05200      IF(NZ.EQ.NZT) GO TO 60
05300      IF(NZ.GT.NZSTRT) GO TO 50
05400  40  Z1=Z(NZ)
05500      Z2=Z(NZ+1)
05600      Z3=Z(NZ+2)
05700      A1=(Z1-Z2)*(Z1-Z3)
05800      A2=(Z2-Z1)*(Z2-Z3)
05900      A3=(Z3-Z1)*(Z3-Z2)
06000      D1=(2.*Z1-Z3-Z2)/A1

```

```

06100      D2=(Z1-Z3)/A2
06200      D3=(Z1-Z2)/A3
06300      DWDZ=UREF1*WNP1(NX)
06400      GO TO 70
06500  50   Z1=Z(NZ-1)
06600      Z2=Z(NZ)
06700      Z3=Z(NZ+1)
06800      A1=(Z2-Z1)*(Z3-Z1)
06900      A2=(Z2-Z1)*(Z3-Z2)
07000      A3=(Z3-Z2)*(Z3-Z1)
07100      D1=-(Z3-Z2)/A1
07200      D2=(Z3-2.*Z2+Z1)/A2
07300      D3=(Z2-Z1)/A3
07400      DWDZ=D1*WE1+D2*WE2+D3*WE3
07500      GO TO 70
07600  60   Z1=Z(NZ-2)
07700      Z2=Z(NZ-1)
07800      Z3=Z(NZ)
07900      A1=(Z2-Z1)*(Z3-Z1)
08000      A2=(Z2-Z1)*(Z3-Z2)
08100      A3=(Z3-Z2)*(Z3-Z1)
08200      D1=(Z3-Z2)/A1
08300      D2=-(Z3-Z1)/A2
08400      D3=(2.*Z3-Z1-Z2)/A3
08500      DWDZ=UREF1*WNP1(NX)
08600  70   DUDZ=D1*UE1+D2*UE2+D3*UE3
08700      WNP=DWDZ/UREF1
08800      DH1DZ=D1*H11+D2*H12+D3*H13
08900      DTHDZ=D1*THETA1+D2*THETA2+D3*THETA3
09000      DS1DZ=D1*S11+D2*S12+D3*S13
09100      CPR1=SQRT(CNUFRS*UE1*S11)*H21*SIN(THETA1)*H11/H21
09200      *      *UREF1/UE1
09300      CPR2=SQRT(CNUFRS*UE2*S12)*H22*SIN(THETA2)*H12/H22
09400      *      *UREF1/UE2
09500      CPR3=SQRT(CNUFRS*UE3*S13)*H23*SIN(THETA3)*H13/H23
09600      *      *UREF1/UE3
09700      DCPDZ=D1*CPR1+D2*CPR2+D3*CPR3
09800      IF(NX.EQ.NXT) GO TO 120
09900      IF(NX.GT.NXSTRT) GO TO 110
10000  100  ICNT=(NZ-1)*NXTL+NX
10100      INX=1
10200      GO TO 130
10300  110  ICNT=(NZ-1)*NXTL+NX-1
10400      INX=2
10500      GO TO 130
10600  120  ICNT=(NZ-1)*NXTL+NX-2
10700      INX=3
10800  130  IPNT2=ICNT
10900      CALL STORIT(TAPEGP,APC,IPCF,IPNT2,1)
11000      H21=APC(2)
11100      THETA1=APC(7)
11200      S11=APC(8)
11300      UE1=APC(9)
11400      WE1=APC(10)
11500      ICNT=ICNT+1
11600      IPNT2=ICNT
11700      CALL STORIT(TAPEGP,APC,IPCF,IPNT2,1)
11800      H22=APC(2)
11900      THETA2=APC(7)
12000      S12=APC(8)

```

```

12100      UE2=APC(9)
12200      WE2=APC(10)
12300      ICNT=ICNT+1
12400      IPNT2=ICNT
12500      CALL STORIT(TAPEGP,APC,IPCF,IPNT2,1)
12600      H23=APC(2)
12700      THETA3=APC(7)
12800      S13=APC(8)
12900      UE3=APC(9)
13000      WE3=APC(10)
13100      IF(INX.EQ.3) GO TO 160
13200      IF(INX.EQ.2) GO TO 150
13300  140  X1=X(NX)
13400      X2=X(NX+1)
13500      X3=X(NX+2)
13600      A1=(X1-X2)*(X1-X3)
13700      A2=(X2-X1)*(X2-X3)
13800      A3=(X3-X1)*(X3-X2)
13900      D1=(2.*X1-X3-X2)/A1
14000      D2=(X1-X3)/A2
14100      D3=(X1-X2)/A3
14200      GO TO 170
14300  150  X1=X(NX-1)
14400      X2=X(NX)
14500      X3=X(NX+1)
14600      A1=(X2-X1)*(X3-X1)
14700      A2=(X2-X1)*(X3-X2)
14800      A3=(X3-X2)*(X3-X1)
14900      D1=-(X3-X2)/A1
15000      D2=(X3-2.*X2+X1)/A2
15100      D3=(X2-X1)/A3
15200      GO TO 170
15300  160  X1=X(NX-2)
15400      X2=X(NX-1)
15500      X3=X(NX)
15600      A1=(X2-X1)*(X3-X1)
15700      A2=(X2-X1)*(X3-X2)
15800      A3=(X3-X2)*(X3-X1)
15900      D1=(X3-X2)/A1
16000      D2=-(X3-X1)/A2
16100      D3=(2.*X3-X1-X2)/A3
16200  170  DUDX=D1*UE1+D2*UE2+D3*UE3
16300      IF(NX.EQ.NXSTRT) DUDX=(UE2-UE1)/(X2-X1)
16400      DBPDX=D1*SQR(CNUFRS*UE1*S11)*H21*SIN(THETA1)+
16500      *      D2*SQR(CNUFRS*UE2*S12)*H22*SIN(THETA2)+
16600      *      D3*SQR(CNUFRS*UE3*S13)*H23*SIN(THETA3)
16700  190  DWDX=D1*WE1+D2*WE2+D3*WE3
16800      HTRM1=H21*SIN(THETA1)
16900      HTRM2=H22*SIN(THETA2)
17000      HTRM3=H23*SIN(THETA3)
17100      DHTRMX=D1*HTRM1+D2*HTRM2+D3*HTRM3
17200      COTTH=COTAN(THETA(2,2))
17300      SINTH=SIN(THETA(2,2))
17400      COSTH=COS(THETA(2,2))
17500      BPR=SQR(CNUFRS*UE(2,2)*PS1(2,2))*H2(2,2)*SINTH
17600      H1H2=H1(2,2)/H2(2,2)
17700      IF(KC.EQ.4) GO TO 460
17800      DUDZ=0.
17900      DWDZ=0.
18000      DS1DZ=0.

```



```

18100      DCPDZ=0.
18200      460  DLOGS1=DS1DZ/PS1(2,2)
18300      P1(2,2)=.5*(PS1(2,2)/(H1(2,2)*UE(2,2))*DUDX+1.)+PS1(2,2)/
18400      *      (H1(2,2)*H2(2,2)*SINTH)*DHTRMX
18500      P2(2,2)=PS1(2,2)/(UE(2,2)*H1(2,2))*DUDX-PS1(2,2)*CK1(2,2)*COTTH
18600      P3(2,2)=-PS1(2,2)*COTTH*CK2(2,2)*UREF1/UE(2,2)
18700      P4(2,2)=PS1(2,2)*CK21(2,2)
18800      P5(2,2)=PS1(2,2)/H2(2,2)*UREF1/UE(2,2)**2*DUDZ+CK12(2,2)*
18900      *      PS1(2,2)*UREF1/UE(2,2)
19000      P6(2,2)=PS1(2,2)/(H1(2,2)*BPR)*DCPDZ
19100      P7(2,2)=PS1(2,2)/H2(2,2)*UREF1/UE(2,2)
19200      P8(2,2)=PS1(2,2)*CK2(2,2)/SINTH*(UREF1/UE(2,2))**2
19300      P9(2,2)=PS1(2,2)*CK1(2,2)/SINTH*UE(2,2)/UREF1
19400      P10(2,2)=PS1(2,2)/H1(2,2)
19500      P11(2,2)=PS1(2,2)*(1./(UE(2,2)*H1(2,2))*DUDX+WE(2,2)/(UE(2,2)
19600      *      **2*H2(2,2))*DUDZ-COTTH*CK1(2,2)+CK2(2,2)/SINTH*
19700      *      (WE(2,2)/UE(2,2))**2+CK12(2,2)*WE(2,2)/UE(2,2))
19800      P12(2,2)=PS1(2,2)/(UE(2,2)*UREF1)*(UE(2,2)/H1(2,2)*DWDZ+
19900      *      WE(2,2)/H2(2,2)*DWDZ-COTTH*CK2(2,2)*WE(2,2)**2+
20000      *      CK1(2,2)/SINTH*UE(2,2)**2+CK21(2,2)*WE(2,2)*UE(2,2))
20100      P13(2,2)=1.-PS1(2,2)/(H1(2,2)*UE(2,2))*DUDX
20200      P14(2,2)=P7(2,2)*(DLOGS1-DUDZ/UE(2,2))
20300      IF(KC.NE.1) RETURN
20400      DWDZ=UREF1*WNP
20500      NX1=NX+1-INX
20600      NX2=NX1+1
20700      NX3=NX2+1
20800      IF(NZ.EQ.NZSTRT) D2WDZX=D1*WNP1(NX1)+D2*WNP1(NX2)+D3*WNP1(NX3)
20900      IF(NZ.EQ.NZT) D2WDZY=D1*WNP1(NX1)+D2*WNP1(NX2)+D3*WNP1(NX3)
21000      P3(2,2)=P7(2,2)
21100      P5(2,2)=0.
21200      P6(2,2)=P7(2,2)
21300      P8(2,2)=0.
21400      P9(2,2)=0.
21500      P12(2,2)=PS1(2,2)/UREF1*(UREF1*D2WDZX/H1(2,2)
21600      *      +DWDZ**2/(UE(2,2)*H2(2,2))+CK21(2,2)*DWDZ)
21700      RETURN
21800      END

```

```

00100      SUBROUTINE PROFIL(LL)
00200      COMMON/BLC0/  NX,NZ,NP,NXT,NZT,NTR,NPT,NXTL,NZTL,NXSTRT,
00300      *              NZSTRT,KC,IT,IFLOW,ICHORD,ISPAN,INTDIR
00400      COMMON/BLC3/  X(81),Z(41),DETA(101),ETA(101),ETAE,VGP,CEL,BEL1,BEL2
00500      COMMON/BLC5/  F(101,2,2),U(101,2,2),V(101),G(101,2,2),W(101,2,2),
00600      *              T(101),TPROF(620),TPCF(10)
00700      COMMON/BLCE/  EDV(101)
00800      COMMON/PRSC/  PS1(2,2),H1(2,2),H2(2,2),UE(2,2),
00900      *              WE(2,2),CK1(2,2),CK2(2,2),CK12(2,2),CK21(2,2),
01000      *              THETA(2,2),PFRS,UFRS,CNUFRS,UREF1,WNP
01100      COMMON/TURB/  E(101,2,2),WT(101),WT2(101,2,2),ELT(101)
01200      COMMON/STOR/  CON,EPS,ICONVE,ITM,X,WTEDG
01300      COMMON/CONS/  SALFA,SALFAS,SBETA,SBETAS,SLAMDA,SSIGMA,SSGMAS,USTOP,
01400      *              ZIOTAE,ZIOTA,RW2
01500 C  -----
01600      GO TO (10,40,70), LL
01700  10  IF(NTR.EQ.0) GO TO 15
01800      CALL START
01900      GO TO 40
02000  15  V(1)=.332
02100      T(1)=.332
02200      WCON=20./SBETA
02300      REY=UE(2,2)*PS1(2,2)/CNUFRS
02400      WTEDG=UREF1/UE(2,2)*SQRT(REY*ZIOTAE/SALFAS)/(ZIOTAL*ETAE)
02500      C1=ETA(NP)*V(1)
02600      C2=3.-2.*C1
02700      C3=-2.+C1
02800      C1D2=C1/2.
02900      C3D4=C3/4.
03000      DO 20 J=1,NP
03100      ERAT=ETA(J)/ETA(NP)
03200      ERATC2=ERAT*C2
03300      ERATSQ=ERAT**2
03400      ERTSQ3=ERATSQ*C3
03500      F(J,2,2)=(C1D2+ERATC2/3.+C3D4*ERATSQ)*ERATSQ*ETA(NP)
03600      U(J,2,2)=(C1+ERATC2+ERTSQ3)*ERAT
03700      G(J,2,2)=F(J,2,2)
03800      W(J,2,2)=U(J,2,2)
03900      E(J,2,2)=0.
04000      IF(J.EQ.1) GO TO 20
04100      WT(J)=WCON/ETA(J)**2
04200      IF(WT(J).LT.WTEDG) WT(J)=WTEDG
04300      WT2(J,2,2)=WT(J)**2
04400  20  CONTINUE
04500      E(NP,2,2)=ZIOTAE
04600      WT2(1,2,2)=WT2(2,2,2)
04700  40  K=NP+1
04800      L=NPT
04900  50  DO 60 J=K,L
05000      F(J,2,2)=ETA(J)+F(K-1,2,2)-ETA(K-1)
05100      U(J,2,2)=1.
05200      W(J,2,2)=W(K-1,2,2)
05300      G(J,2,2)=W(K-1,2,2)*(ETA(J)-ETA(NP))+G(K-1,2,2)
05400      E(J,2,2)=E(K-1,2,2)
05500      WT(J)=WTEDG
05600  60  WT2(J,2,2)=WTEDG**2
05700      IF(LL.NE.1) RETURN
05800      CALL ARRYMV(4,1,2)
05900      RETURN
06000  70  K=NP+1

```

```
06100      NP=NP+2
06200      IF(NP.LE.NPT) GO TO 80
06300      NP=NPT
06400  80   L=NP
06500      GO TO 50
06600      END
```

```

00100      SUBROUTINE SHIFT(NZ1,NZ2)
00200      INTEGER TAPEIN,TAPEOT,TAPEGP,TAPEPF,TAPEDT,TAPEVL
00300      COMMON/BLC0/ NX,NZ,NP,NXT,NZT,NTR,NPT,NXTL,NZTL,NXSTRT,
00400      *          NZSTRT,KC,IT,IFLOW,ICHORD,ISPAN,INTDIR
00500      COMMON/BLC5/ F(101,2,2),U(101,2,2),V(101),G(101,2,2),W(101,2,2),
00600      *          T(101),TPROF(620),TPCF(10)
00700      COMMON/BLC9/ NSEP(41),ICASE(41)
00800      COMMON/TAPE/ TAPEIN,TAPEOT,TAPEGP,TAPEPF,TAPEDT,TAPEVL
00900      COMMON/SVNM/ IPCF,IPRF
01000 C -----
01100 C  WRITE (2,2) FLOW ARRAYS ON DISK
01200      IPNT=NX+(NZ-1)*NXTL
01300      CALL ARRYMV(1,2,2)
01400      CALL STORIT(TAPEPF,TPROF,IPRF,IPNT,0)
01500      INC=3-2*IFLOW
01600      IGO=ICASE(NZ)
01700      GO TO (100,200,300,300,400,500), IGO
01800 C  SET POINTERS FOR NX=NXSTRT EXCEPT AT END OF INITIAL LINE SWEEP
01900 100  IPNTG=NX+(NZ+INC-1)*NXTL
02000      GO TO 650
02100 C  SET POINTERS FOR NX=NXSTRT AND END OF INITIAL LINE SWEEP
02200 200  IF(IFLOW.EQ.2) GO TO 210
02300      IPNT=NX+(NZ1-1)*NXTL
02400      IPNT11=NX+NZ1*NXTL
02500      IPNTG=NX+1+(NZ1-1)*NXTL
02600      GO TO 220
02700 210  IPNT=NX+(NZ2-1)*NXTL
02800      IPNT11=NX+(NZ2-2)*NXTL
02900      IPNTG=NX+1+(NZ2-1)*NXTL
03000 220  CALL STORIT(TAPEPF,TPROF,IPRF,IPNT,1)
03100      CALL ARRYMV(2,1,2)
03200      CALL ARRYMV(5,1,2)
03300      GO TO 600
03400 C  SET POINTERS FOR GENERAL CASE AND BEGINNING OF LINE SWEEP
03500 300  CALL ARRYMV(6,1,2)
03600      IPNTG=NX+(NZ+INC-1)*NXTL
03700      NZNEXT=NZ+INC
03800      IF(ICASE(NZNEXT).NE.4) GO TO 650
03900      CALL ARRYMV(6,2,1)
04000      IPNT11=NX-1+(NZ+2*INC-1)*NXTL
04100      GO TO 600
04200 C  GENERAL CASE...END OF LINE SWEEP
04300 400  IF(NX.EQ.NXT) RETURN
04400      IF(IFLOW.EQ.2) GO TO 410
04500      IPNT=NX+(NZ1-1)*NXTL
04600      IPNT11=NX+NZ1*NXTL
04700      IPNTG=NX+1+(NZ1-1)*NXTL
04800      GO TO 420
04900 410  IPNT=NX+(NZ2-1)*NXTL
05000      IPNT11=NX+(NZ2-2)*NXTL
05100      IPNTG=NX+1+(NZ2-1)*NXTL
05200 420  CALL STORIT(TAPEPF,TPROF,IPRF,IPNT,1)
05300      CALL ARRYMV(2,1,2)
05400      CALL ARRYMV(5,1,2)
05500      GO TO 600
05600 C  END OF ATTACHED REGION
05700 500  NZN=NZ
05800 510  NZN=NZN+INC
05900      NZNP=NZN+INC
06000      IF(NSEP(NZNP).EQ.1) GO TO 510

```

```
06100      IPNT=NX-1+NZN*NXTL
06200      CALL STORIT(TAPEPF,TPROF,IPRF,IPNT,1)
06300      CALL ARRYMV(2,1,2)
06400      CALL ARRYMV(5,1,2)
06500      IPNT11=NX-1+(NZN+INC)*NXTL
06600      IPNTG=NX+NZN*NXTL
06700 C   READ IN (1,1) FLOW ARRAYS AND (2,2) GEOMETRY PARAMETERS
06800 600  CALL STORIT(TAPEPF,TPROF,IPRF,IPNT11,1)
06900      CALL ARRYMV(2,1,1)
07000 650  CALL STORIT(TAPEGP,TPCF,IPCF,IPNTG,1)
07100      CALL ARRYMV(3,2,2)
07200      RETURN
07300      END
```

```

00100      SUBROUTINE SHIFT2(NZ1,NZ2)
00200      INTEGER TAPEIN,TAPEOT,TAPEGP,TAPEPF,TAPEDT,TAPEVL
00300      COMMON/BLC0/ NX,NZ,NP,NXT,NZT,NTR,NPT,NXTL,NZTL,NXSTRT,
00400      *          NZSTRT,KC,IT,IFLOW,ICHORD,ISPAN,INTDIR
00500      COMMON/BLC5/ F(101,2,2),U(101,2,2),V(101),G(101,2,2),W(101,2,2),
00600      *          T(101),TPROF(620),TPCF(10)
00700      COMMON/BLC9/ NSEP(41),ICASE(41)
00800      COMMON/TAPE/ TAPEIN,TAPEOT,TAPEGP,TAPEPF,TAPEDT,TAPEVL
00900      COMMON/SVNM/ IPCF,IPRF
01000 C  -----
01100      IF(IFLOW.EQ.2) GO TO 50
01200 C  SEPARATION ON LOWER BOUNDARY...IFLOW=1
01300      IF(NZ.GT.NZ1) GO TO 10
01400  5    NZ1=NZ1+1
01500      NSEP(NZ1)=1
01600      IF(NSEP(NZ1+1).EQ.1) GO TO 5
01700      IPNT=NX-1+(NZ1-1)*NXTL
01800      CALL STORIT(TAPEPF,TPROF,IPRF,IPNT,1)
01900      CALL ARRYMV(2,1,2)
02000      IPNT11=NX-1+NZ1*NXTL
02100      IPNTG=NX+(NZ1-1)*NXTL
02200      ICASE(NZ1)=3
02300      GO TO 80
02400 C  SEPARARION ON UPPER BOUNDARY...IFLOW=1
02500  10   IF(NZ.LT.NZ2) GO TO 20
02600  15   NZ2=NZ2-1
02700      NSEP(NZ2)=1
02800      IF(NSEP(NZ2-1).EQ.1) GO TO 15
02900      IF(NX.EQ.NXT) RETURN
03000      IPNT=NX+(NZ1-1)*NXTL
03100      CALL STORIT(TAPEPF,TPROF,IPRF,IPNT,1)
03200      CALL ARRYMV(2,1,2)
03300      IPNT11=NX+NZ1*NXTL
03400      IPNTG=NX+1+(NZ1-1)*NXTL
03500      GO TO 80
03600 C  SEPARATION AT INTERIOR POINT...IFLOW=1
03700  20   NZN=NZ+1
03800      NSEP(NZN-1)=1
03900      IF(NSEP(NZN+1).EQ.1) GO TO 20
04000      IPNT=NX-1+(NZN-1)*NXTL
04100      CALL STORIT(TAPEPF,TPROF,IPRF,IPNT,1)
04200      CALL ARRYMV(2,1,2)
04300      IPNT11=NX-1+NZN*NXTL
04400      IPNTG=NX+(NZN-1)*NXTL
04500      ICASE(NZN)=3
04600      GO TO 80
04700 C  SEPARARION ON UPPER BOUNDARY...IFLOW=2
04800  50   IF(NZ.LT.NZ2) GO TO 60
04900  55   NZ2=NZ2-1
05000      NSEP(NZ2)=1
05100      IF(NSEP(NZ2-1).EQ.1) GO TO 55
05200      IPNT=NX-1+(NZ2-1)*NXTL
05300      CALL STORIT(TAPEPF,TPROF,IPRF,IPNT,1)
05400      CALL ARRYMV(2,1,2)
05500      IPNT11=NX-1+(NZ2-2)*NXTL
05600      IPNTG=NX+(NZ2-1)*NXTL
05700      ICASE(NZ2)=3
05800      GO TO 80
05900 C  SEPARATION ON LOWER BOUNDARY...IFLOW=2
06000  60   IF(NZ.GT.NZ1) GO TO 70

```

```

06100 65 NZ1=NZ1+1
06200 NSEP(NZ1)=1
06300 IF(NSEP(NZ1+1).EQ.1) GO TO 65
06400 IF(NX.EQ.NXT) RETURN
06500 IPNT=NX+(NZ2-1)*NXTL
06600 CALL STORIT(TAPEPF,TPROF,IPRF,IPNT,1)
06700 CALL ARRYMV(2,1,2)
06800 IPNT11=NX+(NZ2-2)*NXTL
06900 IPNTG=NX+1+(NZ2-1)*NXTL
07000 GO TO 80
07100 C SEPARATION AT INTERIOR POINT...IFLOW=2
07200 70 NZN=NZ-1
07300 NSEP(NZN+1)=1
07400 IF(NSEP(NZN-1).EQ.1) GO TO 70
07500 IPNT=NX-1+(NZN-1)*NXTL
07600 CALL STORIT(TAPEPF,TPROF,IPRF,IPNT,1)
07700 CALL ARRYMV(2,1,2)
07800 IPNT11=NX-1+(NZN-2)*NXTL
07900 IPNTG=NX+(NZN-1)*NXTL
08000 ICASE(NZN)=3
08100 80 CALL ARRYMV(5,1,2)
08200 CALL STORIT(TAPEPF,TPROF,IPRF,IPNT11,1)
08300 CALL ARRYMV(2,1,1)
08400 CALL STORIT(TAPEGP,TPCF,IPCF,IPNTG,1)
08500 CALL ARRYMV(3,2,2)
08600 RETURN
08700 END

```

```

00100      SUBROUTINE SOLVEV
00200      INTEGER TAPEIN,TAPEOT,TAPEGP,TAPEPF,TAPEDT,TAPEVL
00300      DIMENSION      ESAVE(101),WSAVE(101)
00400      COMMON/BLC0/  NX,NZ,NP,NXT,NZT,NTR,NPT,NXTL,NZTL,NXSTRT,
00500      *              NZSTRT,KC,IT,IFLOW,ICHORD,ISPAN,INTDIR
00600      COMMON/BLC3/  X(81),Z(41),DETA(101),ETA(101),ETAE,VGP,CEL,BEL1,BEL2
00700      COMMON/BLC5/  F(101,2,2),U(101,2,2),V(101),G(101,2,2),W(101,2,2),
00800      *              T(101),TPROF(620),TPCF(10)
00900      COMMON/PRES/  P1(2,2),P2(2,2),P3(2,2),P4(2,2),P5(2,2),
01000      *              P6(2,2),P7(2,2),P8(2,2),P9(2,2),P10(2,2),P11(2,2),
01100      *              P12(2,2),P13(2,2),P14(2,2)
01200      COMMON/PRSC/  PS1(2,2),H1(2,2),H2(2,2),UE(2,2),
01300      *              WE(2,2),CK1(2,2),CK2(2,2),CK12(2,2),CK21(2,2),
01400      *              THETA(2,2),PFRS,UFRS,CNUFRS,UREF1,WNP
01500      COMMON/TURB/  E(101,2,2),WT(101),WT2(101,2,2),ELT(101)
01600      COMMON/STOR/  CON,EPS,ICONVE,ITMAX,WTEDG
01700      COMMON/CONV/  ICOUNE,ICOUNW,INEGE,INEGW,DELV,DELT
01800      COMMON/CONS/  SALFA,SALFAS,SBETA,SBETAS,SLAMDA,SSIGMA,SSGMAS,USTOP,
01900      *              ZIOTAE,ZIOTAL,RW2
02000      COMMON/BLCE/  EDV(101)
02100      COMMON/BLC7/  PU(101),PW(101),QU(101),QW(101)
02200      COMMON/BLCY/  Y1(101),Y3(101),W1,W2,W3,W4
02300      COMMON/IPRT/  IPZ,IPX,IPRINT,EPSV,EPST
02400      COMMON/SHRT/  ISHORT
02500      COMMON/TAPE/  TAPEIN,TAPEOT,TAPEGP,TAPEPF,TAPEDT,TAPEVL
02600      COMMON/RELX/  RFTRB,RFVEL
02700  C - - - - -
02800      DO 1 J=2,NP
02900      IF(U(J,2,2).GE..999) GO TO 2
03000  1      CONTINUE
03100  2      NPL=J
03200      NPM=NPL-1
03300      U2=U(NPL,2,2)
03400      U1=U(NPM,2,2)
03500      ETADEL=ETA(NPM)+DETA(NPM)*(.999-U1)/(U2-U1)
03600      URUE=UREF1/UE(2,2)
03700      URUECT=2.*UREF1/UE(2,2)*COS(THETA(2,2))
03800      REY=UE(2,2)*PS1(2,2)/CNUFRS
03900      UTAU4=V(1)**2
04000      IF(KC.NE.1) UTAU4=UTAU4+(URUE*T(1))**2+URUECT*V(1)*T(1)
04100      WTEDG=SQRT(REY*ZIOTAE/SALFAS)/(ZIOTAL*ETADEL)
04200      UHAT=(UTAU4/REY)**.25
04300      JSTOP=1
04400      WCOEF=20./SBETA
04500      DO 25 J=2,NP
04600      UTOT=U(J,2,2)
04700      IF(KC.NE.1) UTOT=SQRT(UTOT**2+(URUE*W(J,2,2))**2
04800      *              +URUECT*UTOT*W(J,2,2))
04900      UPLUS=UTOT/UHAT
05000      IF(UPLUS.GT.USTOP) GO TO 26
05100      WT(J)=WCOEF/ETA(J)**2
05200      WT2(J,2,2)=WT(J)**2
05300  25      JSTOP=JSTOP+1
05400  26      WT(1)=WT(2)
05500      WT2(1,2,2)=WT(2)**2
05600      E(1,2,2)=0.
05700      ICONVE=0
05800      PU(1)=0.
05900      QU(1)=0.
06000      PW(JSTOP)=WT2(JSTOP,2,2)

```



```

06100      QW(JSTOP)=0.
06200      E4M=1.+5*SSGMAS*(EDV(1)+EDV(2))
06300      E5M=1.+5*SSIGMA*(EDV(1)+EDV(2))
06400      DO 150 J=2,NPM
06500      ESAVE(J)=E(J,2,2)
06600      WSAVE(J)=WT2(J,2,2)
06700      E4P=1.+5*SSGMAS*(EDV(J)+EDV(J+1))
06800      E5P=1.+5*SSIGMA*(EDV(J)+EDV(J+1))
06900      VB=V(J)
07000      TB=T(J)
07100      FB=F(J,2,2)
07200      UB=U(J,2,2)
07300      GB=G(J,2,2)
07400      WB=W(J,2,2)
07500      EB=E(J,2,2)
07600      WTB=WT(J)
07700      E2=P1(2,2)*FB+P6(2,2)*GB
07800      DFB=CEL*(F(J,2,2)-F(J,1,2))
07900      DGB=BEL1*(G(J,2,2)-G(J,2,1))+BEL2*(G(J,1,1)-G(J,1,2))
08000      DVB=(E2+DFB+DGB)/(DETA(J)+DETA(J-1))
08100      RETB=URUE**2*REY*EB/WTB
08200      GAMS=1.-CON*EXP(-RETB)
08300      SHEAR2=VB*VB
08400      IF(KC.NE.1) SHEAR2=SHEAR2+(URUE*TB)**2+URUECT*TB*VB
08500      PRODE=GAMS*REY*SHEAR2/WTB
08600      DISSE=SBETAS*WTB
08700      ABE=.3
08800      PSIFIX=PRODE/DISSE-.7
08900      IF(PSIFIX.GT.ABE) ABE=PSIFIX
09000      X3=PRODE-(1.+ABE)*DISSE
09100      X7=- (CEL*UB+BEL2*WB)*E(J,1,2)-BEL1*WB*E(J,2,1)
09200      *                               +BEL2*WB*E(J,1,1)
09300      A1= E4M*Y3(J)-DVB
09400      B1=-E4P*Y1(J)-E4M*Y3(J)+X3-(CEL*UB+BEL1*WB)
09500      C1= E4P*Y1(J)+DVB
09600      D1=X7-ABE*DISSE*EB
09700      B1S=B1+A1*QU(J-1)
09800      D1S=D1-A1*PU(J-1)
09900      PU(J)=D1S/B1S
10000      QU(J)=-C1/B1S
10100      IF(J.LE.JSTOP) GO TO 145
10200      WT2B=WT2(J,2,2)
10300      GAM=SALFA*(1.-CON*EXP(-RETB/RW2))/GAMS
10400      DLDY=(ELT(J+1)-ELT(J-1))/(DETA(J)+DETA(J-1))
10500      DLDY2=DLDY*DLDY
10600      PRODW=GAM*PRODE+X9
10700      SBETAT=SBETA+2.*SSIGMA*REY*DLDY2*URUE**2
10800      DISSW=SBETAT*WTB
10900      ABW=.3
11000      PSIFIX=PRODW/DISSW-.7
11100      IF(PSIFIX.GT.ABW) ABW=PSIFIX
11200      PSISTP=.5*PRODW/DISSW-.25
11300      IF(PSISTP.GT.ABW) ABW=PSISTP
11400      X6=PRODW-(1.+ABW)*DISSW
11500      X8=- (CEL*UB+BEL2*WB)*WT2(J,1,2)-BEL1*WB*WT2(J,2,1)
11600      *                               +BEL2*WB*WT2(J,1,1)
11700      X9=2.*(P13(2,2)*UB+P14(2,2)*WB)
11800      A2= E5M*Y3(J)-DVB
11900      B2=-E5P*Y1(J)-E5M*Y3(J)+X6-(CEL*UB+BEL1*WB)
12000      C2= E5P*Y1(J)+DVB

```

```

12100      D2=X8-ABW*DISSW*WT2B
12200      B2S=B2+A2*QW(J-1)
12300      D2S=D2-A2*PW(J-1)
12400      PW(J)=D2S/B2S
12500      QW(J)=-C2/B2S
12600  145  E4M=E4P
12700  150  E5M=E5P
12800      WT2EDG=WTEDG*WTEDG
12900      E(NPL,2,2)=ZIoTAE
13000      WT2(NPL,2,2)=WT2EDG
13100      KON=NPM
13200      DO 170 J=2,NPM
13300      E(KON,2,2)=PU(KON)+QU(KON)*E(KON+1,2,2)
13400      IF(KON.LE.JSTOP) GO TO 170
13500      WT2(KON,2,2)=PW(KON)+QW(KON)*WT2(KON+1,2,2)
13600  170  KON=KON-1
13700      ICOUNE=0
13800      INEGE=0
13900      DO 175 J=2,NPM
14000      IF(E(J,2,2).GE.0.) GO TO 176
14100      INEGE=INEGE+1
14200      E(J,2,2)=.5*ESAVE(J)
14300      IF(J.GE.(NPL-3)) E(J,2,2)=ZIoTAE
14400      GO TO 175
14500  176  IF(ABS(E(J,2,2)-ESAVE(J)).GT.EPS*ESAVE(J)) ICOUNE=ICOUNE+1
14600  175  E(J,2,2)=ESAVE(J)+RFTRB*(E(J,2,2)-ESAVE(J))
14700      ICOUNW=0
14800      INEGW=0
14900      DO 195 J=JSTOP,NPM
15000      IF(WT2(J,2,2).GT.0.) GO TO 196
15100      WT2(J,2,2)=.5*WSAVE(J)
15200      INEGW=INEGW+1
15300      GO TO 197
15400  196  IF(ABS(WT2(J,2,2)-WSAVE(J)).GT.EPS*WSAVE(J)) ICOUNW=ICOUNW+1
15500      WT2(J,2,2)=WSAVE(J)+RFTRB*(WT2(J,2,2)-WSAVE(J))
15600  197  WT(J)=SQRT(WT2(J,2,2))
15700  195  CONTINUE
15800      ICOUNT=ICOUNW+ICOUNE+INEGW+INEGE
15900      IF(ICOUNT.LE.2) ICONVE=1
16000      NPP=NPL+1
16100      DO 200 J=NPP,NPT
16200      E(J,2,2)=E(NPL,2,2)
16300      WT(J)=WTEDG
16400  200  WT2(J,2,2)=WT2EDG
16500      IF(ISHORT.EQ.1) RETURN
16600      IF(IPRINT.EQ.0) RETURN
16700      IF(IT.EQ.1) WRITE(TAPEOT,40)
16800      WRITE(TAPEOT,50) IT,V(1),DELV,T(1),DELT,INEGE,
16900      *           INEGW,ICOUNE,ICOUNW
17000      RETURN
17100  40  FORMAT(1H0,1X2HIT,5X7HV(WALL),7X7HDELV(1),7X7HDELT,
17200      *           7X5HINEGE,3X5HINEGW,3X6HICOUNE,1X4HICOUNT)
17300  50  FORMAT(1H ,I3,1P4E14.5,4I8)
17400      END

```

AD-A123 285

A MICROCOMPUTER-BASED METHOD FOR PREDICTING VISCOUS
FLOW ABOUT A SHIP HULL(U) DCW INDUSTRIES INC STUDIO
CITY CA D C WILCOX DEC 82 DCW-R-28-81 ONR-CR062-735-1F
UNCLASSIFIED N00014-82-C-0432

2/2

NL

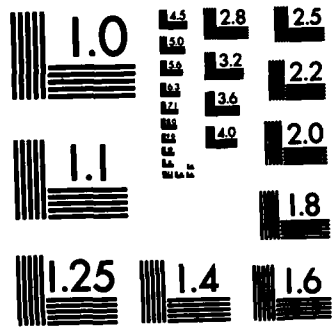
F/G 9/2



END

FILMED

1 811



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```

00100      SUBROUTINE SOLV6
00200      DIMENSION USAVE(101), WSAVE(101)
00300      COMMON/BLC0/  NX,NZ,NP,NXT,NZT,NTR,NPT,NXTL,NZTL,NXSTRT,
00400      *              NZSTRT,KC,IT,IFLOW,ICHORD,ISPAN,INTDIR
00500      COMMON/BLC3/  X(81),Z(41),DETA(101),ETA(101),ETAE,VGP,CEL,BEL1,BEL
00600      COMMON/BLC5/  F(101,2,2),U(101,2,2),V(101),G(101,2,2),W(101,2,2),
00700      *              T(101),TPROF(620),TPCF(10)
00800      COMMON/BLC7/  PU(101),PW(101),QU(101),QW(101)
00900      COMMON/BLCY/  Y1(101),Y3(101),W1,W2,W3,W4
01000      COMMON/IPRT/  IPZ,IPX,IPRINT,EPSV,EPST
01100      COMMON/CONV/  ICOUNE,ICOUNW,INEGE,INEGW,DELV,DELT
01200      COMMON/RELX/  RFTRB,RFVEL
01300  C  - - - - -
01400      VWSAVE=V(1)
01500      TWSAVE=T(1)
01600      NPM=NP-1
01700      KON=NPM
01800      DO 10 J=2,NPM
01900      USAVE(KON)=U(KON,2,2)
02000      WSAVE(KON)=W(KON,2,2)
02100      U(KON,2,2)=PU(KON)+QU(KON)*U(KON+1,2,2)
02200      W(KON,2,2)=PW(KON)+QW(KON)*W(KON+1,2,2)
02300 10  KON=KON-1
02400      U(1,2,2)=0.
02500      W(1,2,2)=0.
02600      DO 20 J=2,NPM
02700      U(J,2,2)=USAVE(J)+RFVEL*(U(J,2,2)-USAVE(J))
02800 20  W(J,2,2)=WSAVE(J)+RFVEL*(W(J,2,2)-WSAVE(J))
02900      F(1,2,2)=0.
03000      G(1,2,2)=0.
03100      CF1=0.
03200      CG1=0.
03300      DO 30 J=2,NP
03400      CF=U(J,2,2)
03500      CG=W(J,2,2)
03600      F(J,2,2)=F(J-1,2,2)+.5*(CF+CF1)*DETA(J-1)
03700      G(J,2,2)=G(J-1,2,2)+.5*(CG+CG1)*DETA(J-1)
03800      CF1=CF
03900 30  CG1=CG
04000      DO 35 J=2,NPM
04100      V(J)=(U(J+1,2,2)-U(J-1,2,2))/(DETA(J)+DETA(J-1))
04200 35  T(J)=(W(J+1,2,2)-W(J-1,2,2))/(DETA(J)+DETA(J-1))
04300      V(NP)=(1.-U(NPM,2,2))/DETA(NPM)
04400      T(NP)=(W(NP,2,2)-W(NPM,2,2))/DETA(NPM)
04500      V(1)=-W1*U(1,2,2)+W2*U(2,2,2)-W3*U(3,2,2)+W4*U(4,2,2)
04600      T(1)=-W1*W(1,2,2)+W2*W(2,2,2)-W3*W(3,2,2)+W4*W(4,2,2)
04700      DELV=V(1)-VWSAVE
04800      DELT=T(1)-TWSAVE
04900      RETURN
05000      END

```

```

00100 SUBROUTINE START
00200 INTEGER TAPEIN, TAPEOT, TAPEGP, TAPEPF, TAPEDT, TAPEVL
00300 DIMENSION YPLUS(101)
00400 COMMON/BLC0/ NX,NZ,NP,NXT,NZT,NTR,NPT,NXTL,NZTL,NXSTRT,
00500 * NZSTRT,KC,IT,IFLOW,ICHORD,ISPAN,INTDIR
00600 COMMON/BLC3/ X(01),Z(41),DETA(101),ETA(101),ETAE,VGP,CEL,BEL1,BEL2
00700 COMMON/BLC5/ F(101,2,2),U(101,2,2),V(101),G(101,2,2),W(101,2,2),
00800 * T(101),TPROF(620),TPCF(10)
00900 COMMON/PRBC/ PS1(2,2),H1(2,2),H2(2,2),UE(2,2),
01000 * WE(2,2),CK1(2,2),CK2(2,2),CK12(2,2),CK21(2,2),
01100 * THETA(2,2),PFRS,UFRS,CNUFRS,UREF1,WNP
01200 COMMON/BLCE/ EDV(101)
01300 COMMON/TURB/ E(101,2,2),WT(101),WT2(101,2,2),ELT(101)
01400 COMMON/STOR/ CON,EPS,ICONVE,ITMAX,WTEDG
01500 COMMON/TAPE/ TAPEIN,TAPEOT,TAPEGP,TAPEPF,TAPEDT,TAPEVL
01600 COMMON/CONS/ SALFA,SALFAS,SBETA,SBETAS,SLAMDA,SSIGMA,SSGMAS,USTOP,
01700 * ZIOTAE,ZIOTAL,RW2
01800 COMMON/INTG/ ALAMI(41),CFXI(41),CFZI(41),DELTAI(41),
01900 * THETXI(41),THETZI(41)
02000 DATA PI/3.14159265/,AKAPPA/.41/
02100 C - - - - -
02200 PS1(1,2)=PS1(2,2)-H1(2,2)*(X(NXSTRT+1)-X(NXSTRT))
02300 ALAMM=ALAMI(NZ)
02400 CFX=CFXI(NZ)
02500 CFZ=CFZI(NZ)
02600 DELTA=DELTAI(NZ)
02700 THETAX=THETXI(NZ)
02800 THETAZ=THETZI(NZ)
02900 RES=UE(2,2)*PS1(1,2)/CNUFRS
03000 REDEL=UE(2,2)*DELTA/CNUFRS
03100 ROOTRS=SQRT(RES)
03200 URUE=UREF1/UE(2,2)
03300 URUE2=URUE**2
03400 UTAU2=.5*(CFX+ABS(CFZ))
03500 UTAU=SQRT(UTAU2)
03600 DELTAP=UREF1*UTAU*DELTA/CNUFRS
03700 YSCALP=UTAU*URUE*ROOTRS
03800 COSTH=COS(THETA(2,2))
03900 SINTH=SIN(THETA(2,2))
04000 COTTH=COSTH/SINTH
04100 DO 10 J=1,NP
04200 10 YPLUS(J)=YSCALP*ETA(J)
04300 C VELOCITY PROFILES
04400 DOTXM3=DELTA/THETAX-3.
04500 DOTZM3=3.
04600 IF(ABS(THETAZ).GT.1.E-8) DOTZM3=DELTA/THETAZ-3.
04700 ENX=.5*(DOTXM3+SQRT(DOTXM3*DOTXM3-8.))
04800 ENZ=.5*(DOTZM3+SQRT(DOTZM3*DOTZM3-8.))
04900 SUBSCL=.5*ROOTRS*URUE2
05000 OTSCLX=(1.+WE(2,2)*COSTH/UE(2,2))*(ROOTRS/REDEL)**(1./ENX)
05100 OTSCLZ=(WE(2,2)*SINTH/UE(2,2))*(ROOTRS/REDEL)**(1./ENZ)
05200 V(1)=SUBSCL*CFX
05300 T(1)=SUBSCL*CFZ
05400 U(1,2,2)=0.
05500 W(1,2,2)=0.
05600 DO 20 J=2,NP
05700 IF(YPLUS(J).GE.DELTAP) GO TO 30
05800 U(J,2,2)=CFX*SUBSCL*ETA(J)
05900 W(J,2,2)=CFZ*SUBSCL*ETA(J)
06000 UOUT=OTSCLX*ETA(J)**(1./ENX)

```

```

06100      WOUT=OTSC LZ*ETA(J)**(1./ENZ)
06200      IF(U(J,2,2).GT.UOUT) U(J,2,2)=UOUT
06300  20    IF(ABS(W(J,2,2)).GT.ABS(WOUT)) W(J,2,2)=WOUT
06400  30    WSCALE=1./(URUE*SINTH)
06500      DO 40 J=2,NP
06600      IF(YPLUS(J).GE.DELTAP) GO TO 35
06700      U(J,2,2)=U(J,2,2)-W(J,2,2)*COTTH
06800      W(J,2,2)=WSCALE*W(J,2,2)
06900      GO TO 40
07000  35    U(J,2,2)=1.
07100      W(J,2,2)=WE(2,2)/UREF1
07200  40    CONTINUE
07300      V(1)=V(1)-T(1)*COTTH
07400      T(1)=WSCALE*T(1)
07500  C    STREAMFUNCTION COMPONENTS
07600      F(1,2,2)=0.
07700      G(1,2,2)=0.
07800      CF1=0.
07900      CG1=0.
08000      DO 50 J=2,NP
08100      CF=U(J,2,2)
08200      CG=W(J,2,2)
08300      F(J,2,2)=F(J-1,2,2)+.5*(CF+CF1)*DETA(J-1)
08400      G(J,2,2)=G(J-1,2,2)+.5*(CG+CG1)*DETA(J-1)
08500      CF1=CF
08600  50    CG1=CG
08700  C    TURBULENT MIXING ENERGY
08800      ESCALE=ALAMM*UTAU2/SALFAS
08900      EMATCH=ESCALE*(COS(.5*PI/DELTAP))**2
09000      DO 70 J=1,NP
09100      IF(YPLUS(J).GT.10.) GO TO 65
09200      E(J,2,2)=EMATCH*(.1*YPLUS(J))**4
09300      GO TO 70
09400  65    IF(YPLUS(J).GE.DELTAP) GO TO 68
09500      E(J,2,2)=ESCALE*(COS(.5*PI*YPLUS(J)/DELTAP))**2
09600      IF(E(J,2,2).GT.ZIOTAE) GO TO 70
09700  68    E(J,2,2)=ZIOTAE
09800  70    CONTINUE
09900  C    TURBULENT DISSIPATION RATE
10000      YPLSUB=20.*SALFAS*AKAPPA/SBETA
10100      WSUB=(20./SBETA)**2
10200      WWAL=(ALAMM*YSCALP/(SALFAS*AKAPPA))**2
10300      WWAK=(PS1(1,2)*URUE/(ZITOTAL*DELTA))**2/SALFAS
10400      DO 80 J=2,NP
10500      IF(YPLUS(J).GT.YPLSUB) GO TO 75
10600      WT2(J,2,2)=WSUB/ETA(J)**4
10700      GO TO 80
10800  75    WTEST1=WWAL/ETA(J)**2
10900      WTEST2=WWAK*E(J,2,2)
11000      IF(WTEST1.LT.WTEST2.OR.YPLUS(J).GE.DELTAP) GO TO 78
11100      WT2(J,2,2)=WTEST1
11200      GO TO 80
11300  78    WT2(J,2,2)=WTEST2
11400  80    WT(J)=SQRT(WT2(J,2,2))
11500      WTEDG=WT(NP)
11600      WT(1)=WT(2)
11700      WT2(1,2,2)=WT(2)**2
11800      RETURN
11900      END

```

```

00100      SUBROUTINE STORIT(IUNIT,A,NDIM,IPNT,N)
00200 C *****
00300 C THIS SUBROUTINE COORDINATES TEMPORARY STORAGE AND RETREIVAL
00400 C OF DATA FROM DISK. DATA ARE STORED AS A SERIES OF BLOCK
00500 C DATA IN THE A ARRAY WHOSE DIMENSION IS NDIM. DURING A READ
00600 C FROM THE SUBROUTINE (N=1), THE FIRST (IPNT-1) BLOCKS ARE
00700 C SKIPPED AND THE A ARRAY CORRESPONDING TO IPNT IS READ FROM
00800 C DISK. DATA ARE WRITTEN ON DISK WHEN N=0.
00900 C *****
01000      INTEGER TAPEIN,TAPEOT,TAPEGP,TAPEPF,TAPEDT,TAPEVL
01100      DIMENSION A(620)
01200      COMMON/TAPE/ TAPEIN,TAPEOT,TAPEGP,TAPEPF,TAPEDT,TAPEVL
01300 C *****
01400 C WRITE DATA ON DISK
01500 C *****
01600 C
01700      IF(N.NE.0) GO TO 10
01800 C
01900 C*****IBM OPTION*****
02000 C      WRITE(IUNIT,IPNT) A
02100 C*****CDC 7600 OPTION*****
02200 C      CALL WRITMS(IUNIT,A,NDIM,IPNT)
02300 C*****UNIVAC 1108 OPTION*****
02400      INDEX=NDIM*(IPNT-1)
02500      CALL NTRAN(IUNIT,10)
02600      CALL NTRAN(IUNIT,6,INDEX)
02700      CALL NTRAN(IUNIT,1,NDIM,A,LFLAG)
02800 1      IF(LFLAG.EQ.-1) GO TO 1
02900      IF(LFLAG.LT.-1) GO TO 20
03000      RETURN
03100 C
03200 C *****
03300 C READ DATA FROM DISK
03400 C *****
03500 C
03600 10      CONTINUE
03700 C
03800 C*****IBM OPTION*****
03900 C      READ (IUNIT,IPNT) A
04000 C*****CDC 7600 OPTION*****
04100 C      CALL READMS(IUNIT,A,NDIM,IPNT)
04200 C*****UNIVAC 1108 OPTION*****
04300      INDEX=NDIM*(IPNT-1)
04400      CALL NTRAN(IUNIT,10)
04500      CALL NTRAN(IUNIT,6,INDEX)
04600      CALL NTRAN(IUNIT,2,NDIM,A,LFLAG)
04700 11      IF(LFLAG.EQ.-1) GO TO 11
04800      IF(LFLAG.LT.-1) GO TO 20
04900      RETURN
05000 C
05100 20      IF(N.EQ.0) WRITE(TAPEOT,5)
05200      IF(N.EQ.1) WRITE(TAPEOT,6)
05300      IF(LFLAG.EQ.-2) WRITE(TAPEOT,2) LFLAG
05400      IF(LFLAG.EQ.-3) WRITE(TAPEOT,3) LFLAG
05500      IF(LFLAG.EQ.-4) WRITE(TAPEOT,4) LFLAG
05600      STOP
05700 C *****
05800 2      FORMAT(1H ,1X7HLFLAG =,I3,5X23HEND OF FILE ENCOUNTERED)
05900 3      FORMAT(1H ,1X7HLFLAG =,I3,5X12HDEVICE ERROR)
06000 4      FORMAT(1H ,1X7HLFLAG =,I3,5X20HTRANSMISSION ABORTED)

```



```
06100 5   FORMAT(1H1,29HWRITE ERROR DETECTED BY NTRAN)
06200 6   FORMAT(1H1,28HREAD ERROR DETECTED BY NTRAN)
06300 C *****
06400     END
```

```

10 REM *****
20 REM ***** INTERPOLATION ROUTINE TO ACCOUNT *****
30 REM ***** FOR EFFECTS OF *****
40 REM ***** DISPLACEMENT THICKNESS *****
50 REM *****
60 DEFUSR1=&H7D00
70 CLEAR 1000
80 DIM X(41),Z(21),DISP(41,21)
90 DIM ZB(41,21)
100 DIM XR(41),YR(41,11),ZR(11),YN(41,11)
110 OPEN "I",1,"DISPL"
120 INPUT "NXTL =" ;NX
130 INPUT "NZTL =" ;NZ
140 FOR I=1 TO NX
150 FOR J=1 TO NZ
160 INPUT#1, DISP(I,J)
170 NEXT J
180 NEXT I
190 CLOSE
200 INPUT "FILE NAME FOR NONORTHOGONAL MESH" ;F$
210 OPEN "I",1,F$
220 FOR J=1 TO NZ
230 INPUT#1, H$,Z(J)
240 FOR I=1 TO NX
250 INPUT#1, D,X(I),D,ZB(I,J),D,D,D,D,D,D,D,D
260 NEXT I
270 NEXT J
280 CLOSE
290 OPEN "I",1,"MESHPR"
300 INPUT "IMAX =" ;IM
310 INPUT "JMAX =" ;JM
320 FOR I=1 TO IM
330 INPUT#1, XR(I)
340 NEXT I
350 FOR J=1 TO JM
360 INPUT#1, ZR(J)
370 NEXT J
380 FOR I=1 TO IM
390 FOR J=1 TO JM
400 INPUT#1, YR(I,J)
410 NEXT J
420 NEXT I
430 CLOSE
440 NAME "MESHPR" AS "MESHPR/OLD"
450 YN=USR1(X,Z,DISP,ZB,XR,YR,ZR,NX,NZ,IM,JM)
460 OPEN "O",1,"MESHPR"
470 FOR I=1 TO IM
480 PRINT#1 USING "##.#####↑↑↑↑",XR(I)
490 NEXT I
500 FOR J=1 TO JM
510 PRINT#1 USING "##.#####↑↑↑↑",ZR(J)
520 NEXT J
530 FOR I=1 TO IM
540 FOR J=1 TO JM
550 PRINT#1 USING "##.#####↑↑↑↑",YN(I,J)
560 NEXT J
570 NEXT I
580 CLOSE
590 END

```

2-8

DT