

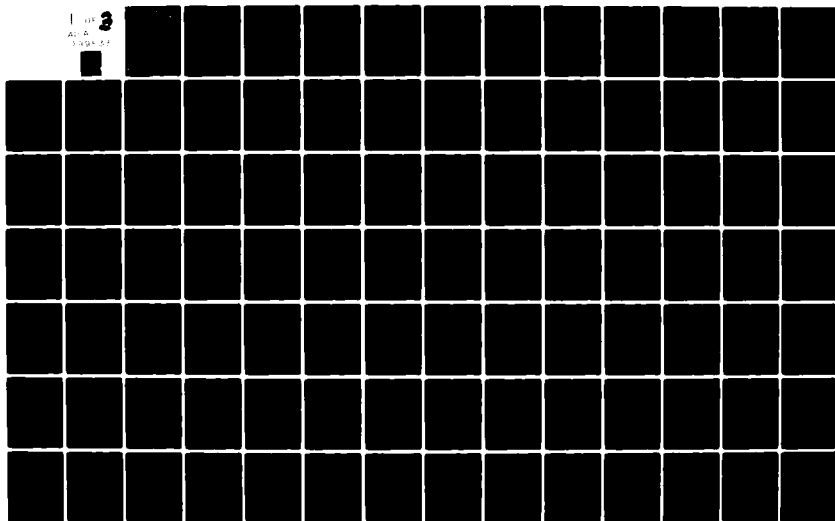
AD-A099 533

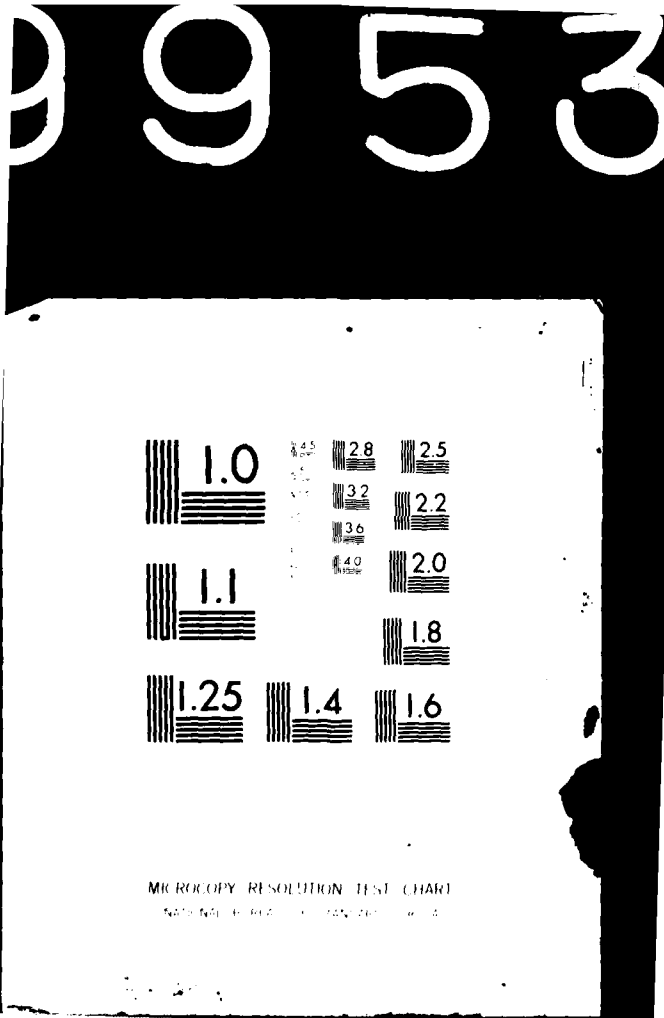
DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CE--ETC F/G 20/4
DOCUMENTATION FOR SWATH SHIP RESISTANCE AND PROPULSION PREDICTI--ETC(U)
APR 81 A M REED

UNCLASSIFIED DTNSRDC/SPD-0927-02

NL

1 of 3
A-1
1000-10





LEVEL II

12

DTNSRDC/SPD-0927-02

DOCUMENTATION FOR SWATH SHIP RESISTANCE AND PROPULSION PREDICTION PROGRAMS
(CLOSEFIT AND SYNTHESIS): MAINTENANCE MANUAL

AD A 099 533

**DAVID W. TAYLOR NAVAL SHIP
RESEARCH AND DEVELOPMENT CENTER**

Bethesda, Maryland 20084



DOCUMENTATION FOR SWATH SHIP
RESISTANCE AND PROPULSION PREDICTION PROGRAMS

(CLOSEFIT AND SYNTHESIS):

MAINTENANCE MANUAL

Arthur M. Reed

Approved for Public Release: Distribution Unlimited

DTIC
ELECTE
JUN 0 1 1981
E

DTIC FILE COPY

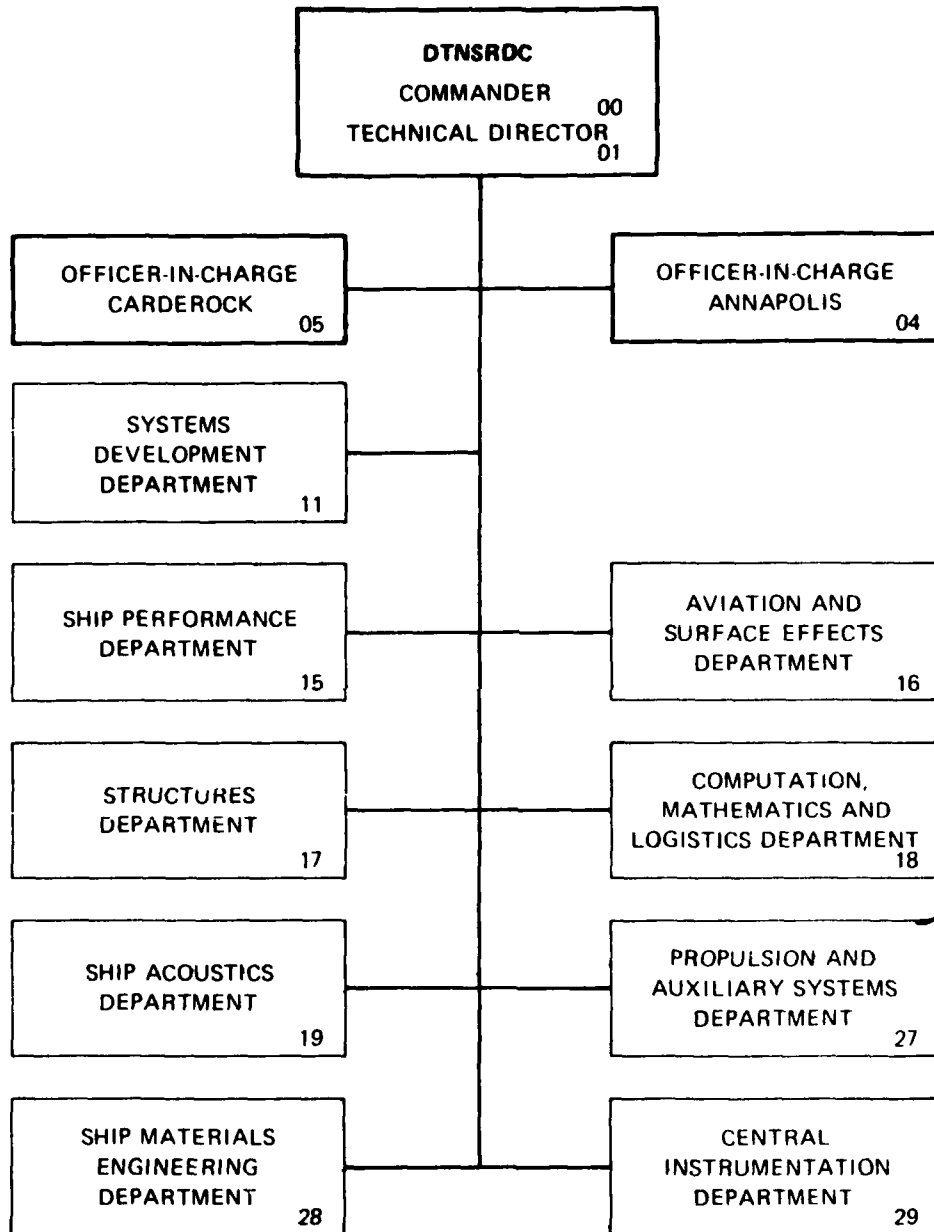
Ship Performance Department

APRIL 1981

DTNSRDC/SPD-0927-02

81 6 01 107

MAJOR DTNSRDC ORGANIZATIONAL COMPONENTS



UNCLASSIFIED 387 612

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

14 REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM	
1. REPORT NUMBER DTNSRDC/SPD-0927-02	2. GOVT ACCESSION NO. ADA099533	3. RECIPIENT'S CATALOG NUMBER 9	
4. TITLE (and Subtitle) DOCUMENTATION FOR SWATH SHIP RESISTANCE AND PROPULSION PREDICTION PROGRAMS (CLOSEFIT AND SYNTHESIS): MAINTENANCE MANUAL		5. TYPE OF REPORT & PERIOD COVERED FINAL rept.	
7. AUTHOR(s) Arthur M./Reed		6. CONTRACT OR GRANT NUMBER(s) 17	
9. PERFORMING ORGANIZATION NAME AND ADDRESS David W. Taylor Naval Ship Research and Development Center, Bethesda, Maryland 20084		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62543N, ZF43 421 001 1-1500-103	
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE 11 APR 1981	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) (16) F43421		13. NUMBER OF PAGES	
		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) Small-Waterplane-Area Twin-Hull (SWATH) ships, resistance and propulsion prediction			
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report documents two computer programs which determine the wave resistance and propulsive performance of Small-Waterplane-Area Twin-Hull (SWATH) ships. These programs are SYNTHESIS and CLOSEFIT, and they use as input data the moments and offsets of ship geometry, respectively. While they both employ linearized ship wave theory and thin-ship approximation and share many sub-routines, CLOSEFIT is a more refined program capable of furnishing superior results. On the other hand, SYNTHESIS is simpler and includes several additional subroutines which can determine the characteristics and performance			

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

i

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

389694

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

of a SWATH ship's propulsive system at both design and off-design conditions.

61

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

TABLE OF CONTENTS

	Page
LIST OF FIGURES	iv
NOMENCLATURE	v
ABSTRACT.	1
ADMINISTRATIVE INFORMATION	1
BACKGROUND	2
OVERVIEW OF CLOSEFIT.	7
OVERVIEW OF SYNTHESIS.	13
PROGRAM DOCUMENTATION OF CLOSEFIT.	15
COMMON BLOCK DEFINITIONS FOR CLOSEFIT.	16
SUBROUTINE DOCUMENTATION FOR CLOSEFIT.	27
PROGRAM DOCUMENTATION OF SYNTHESIS	61
COMMON BLOCK DEFINITIONS FOR SYNTHESIS	63
SUBROUTINE DOCUMENTATION FOR SYNTHESIS	79
REFERENCES.	117
APPENDICES.	118
PROGRAM LISTING OF CLOSEFIT.	119
PROGRAM LISTING OF SYNTHESIS	171

LIST OF FIGURES

	Page
1 - Profile of SWATH Ship with Single Strut	4
2 - Profile of SWATH Ship with Tandem Struts	5
3 - Tree Diagram of CLOSEFIT SWATH Program	8
4 - Tree Diagram of SYNTHESIS SWATH Program	14

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A	

NOMENCLATURE

<u>Symbol</u>	<u>Description</u>
$A(x)$	Body sectional area function
A_o	Maximum section area of body
A_{Bm}	Symmetric coefficients of Chebychev sine series for body
A_{Sm}	Symmetric coefficients of Chebychev sine series for strut
A_{wp}	Area of the waterplane
b	Half of the separation distance of the hulls
B_{Bm}	Anti-symmetric coefficients of Chebychev series for body
B_{Sm}	Anti-symmetric coefficients of chebychev sine series for strut
C_A	Correlation Allowance
C_F	Frictional resistance coefficient
C_{Fm}	Form drag coefficient
C_P	Body Prismatic coefficient
C_{WP}	Waterplane coefficient
C_{IW}	Waterplane inertia coefficient
g	Acceleration due to gravity
h_B	Maximum depth of submergence of axis of body
h_S	Maximum draft of strut
$J_n(a)$	Bessel functions
L_B	Maximum length of body
L_S	Maximum length of strut
P_E	Effective power

NOMENCLATURE (CONT)

<u>Symbol</u>	<u>Description</u>
R_n	Reynolds number
R	Total ship resistance
R_{WB}	Wave resistance due to one main body
R_F	Frictional resistance
R_{Fm}	Eddy resistance
R_{WS}	Wave resistance due to one strut
R_{WSB}	Wave resistance due to the interaction of strut and main body
R_W	Wave resistance
S	Wetted surface
T	Thickness of mid length of strut
$t(x)$	Strut half thickness function
T_{Bmn}	Auxiliary function used in calculation of R_B
T (T_{max} also used)	Thickness of mid length of strut
T_{SBmn}	Auxiliary wave resistance function used in calculation of R_{SB}
T_{Smn}	Auxiliary wave resistance function used in calculation of R_S
$U_m(x)$	Cheby chev cosine series
V	Velocity of ship
$V_m(x)$	Chebychev sin series
W_{Bmn}	Auxiliary wave resistance function used in calculation of R_B
W_{Smn}	Auxiliary wave resistance used in calculation of R_S

NONMENCLATURE (CONT)

<u>Symbol</u>	<u>Description</u>
W_{SBmn}	Auxiliary wave resistance function used in calculation of R_{SB} Variable of integration
ρ	Density of water
γ_{OB}	Dimensionless number related to body length and ship speed, used to remove singularities
γ_{OS}	Dimensionless number related to strut length and ship speed, used to remove singularities
ζ	Variable of integration.

ENGLISH/SI EQUIVALENTS

1 degree (angle)	= 0.01745 rad (radians)
1 foot	= 0.3048 m (meters)
1 foot per second (fps)	= 0.3048 m/sec (meters per second)
1 inch	= 25.40 mm (millimeters)
1 knot	= 0.5144 m/s (meters per second)
1 lb (force)	= 4.448 N (Newtons)
1 lb (force - inch lbs)	= 0.1130 N·m (Newton-meter)
1 long ton (2240 pounds)	= 1.016 metric tons; or 1016 kilograms
1 power	= 0.746 kW (kilowatts)

ABSTRACT

This report documents two computer programs which determine the wave resistance and propulsive performance of Small-Waterplane-Area Twin-Hull (SWATH) ships. These programs are SYNTHESIS and CLOSEFIT, and they use as input data the moments and offsets of ship geometry, respectively. While they both employ linearized ship wave theory and thin-ship approximation and share many subroutines, CLOSEFIT is a more refined program capable of furnishing superior results. On the other hand, SYNTHESIS is simpler and includes several additional subroutines which can determine the characteristics and performance of a SWATH ship's propulsive system at both design and off-design conditions.

ADMINISTRATIVE INFORMATION

This investigation was authorized under a direct funded block from the Naval Material Command (NAVMAT 08T23) under Program Element 62543N, Task Area ZF43-421-001, and funded through the High Performance Vehicles Office of the David W. Taylor Naval Ship Research and Development Center (DTNSRDC), Work Unit 1-1500-103.

BACKGROUND

This report (Maintenance Manual) contains documentation for two computer programs, CLOSEFIT and SYNTHESIS, which have been developed at the David W. Taylor Naval Ship R&D Center (DTNSRDC) to predict the effective and delivered power for Small-Waterplane-Area Twin-Hull (SWATH) ships. This maintenance manual contains documentation for each subroutine in the programs, definitions of all of the variables in the COMMON BLOCKS, and a listing of the computer programs. This documentation is intended for the individual who must modify the computer programs, and must therefore be able to understand the functioning of the programs, and should be contrasted with the Users Manual, which serves to define the input to and output from the programs. The ability to prepare proper input, and to use the output does not require an understanding of the workings of the program.

SWATH ships are catamarans which combine a thin strut with a submerged elongated slender body. Each demi-hull of the SWATH has either one (single) or two (tandem) struts which support the lower hull. Figures 1 and 2 show schematic diagrams illustrating the geometry of single and tandem strut SWATH configurations, respectively.

Programs CLOSEFIT and SYNTHESIS have a similar theoretical foundation in that they are both based on linearized ship wave theory and the thin ship approximation in the framework of the potential flow of an incompressible, inviscid fluid. In general, thin-ship theory can be applied to any conventional catamaran with arbitrary camber. However, as this theory has been specialized to SWATH configurations, the ship geometry is

restricted to ships with "no-cross flow" cambers. In practice, this means SWATHS with very small strut thicknesses compared to the transverse strut separation.

In the theory of ship resistance extrapolation, the resistance of a ship is typically decomposed into residuary and frictional resistance components. The frictional resistance component is generally taken as equal to the resistance of a flat plate with area of the ship, and with length equal to the length of the ship. The three dimensional viscous effects are then included in the residuary resistance along with other resistance components such as wave making resistance. For a SWATH ship, the calculation of the frictional resistance is not as simple as stated above, however, the same principle applies if one calculated the frictional resistance of the bodies and struts separately.

The wave making resistance component of the residuary resistance can be computed using inviscid fluid flow theory. However, the other components of the residuary resistance are not easily calculated theoretically. Because of this, the remainder of the residuary resistance is computed empirically. The difference between the experimentally determined residuary resistance and the theoretically determined wave resistance is calculated for those cases where adequate experimental data exists. These differences which are denoted as "form drag", are then plotted as a function of strut speed-length ratio. Under the assumption that the form drag remains unchanged for reasonable changes of SWATH proportions, a curve can be faired through the individual form drag curves, and applied to new hull forms.

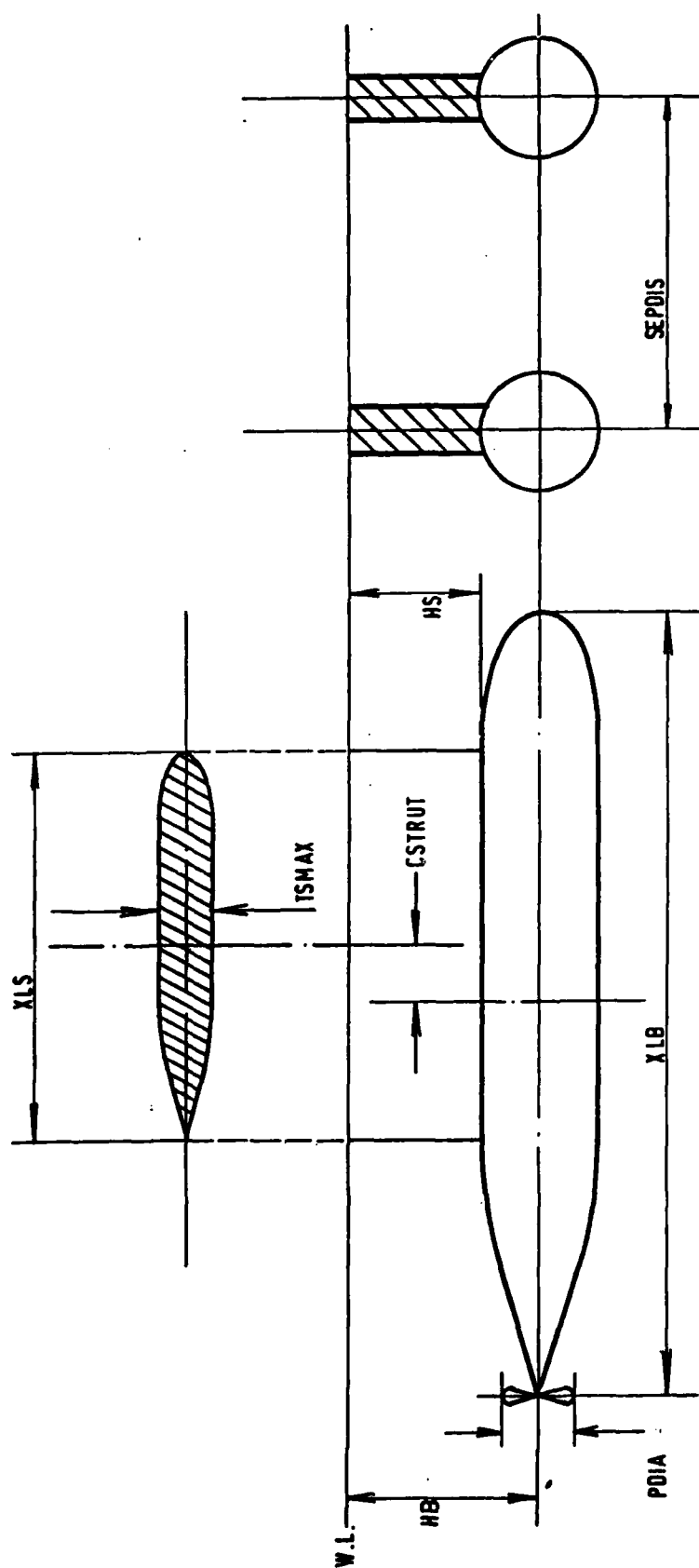


Figure 1 - Profile of SWATH Ship with Single Strut

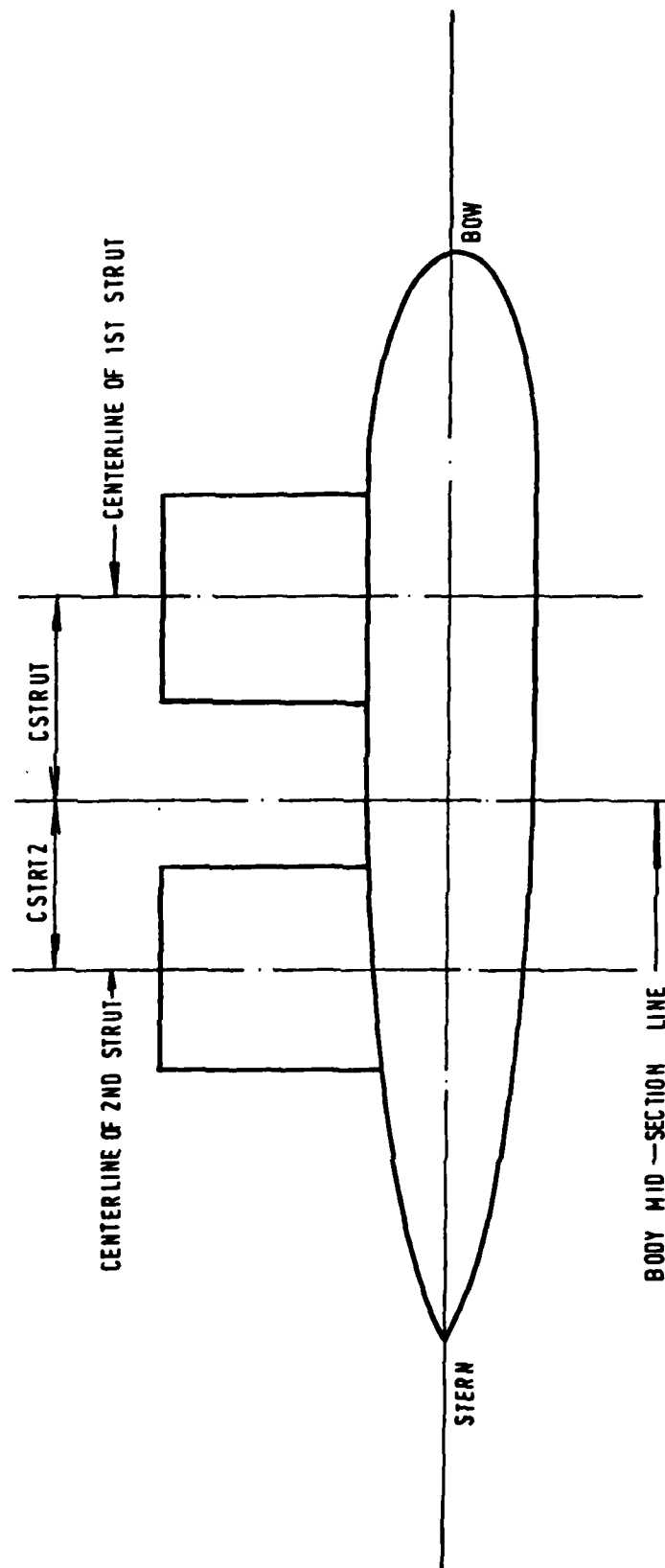


Figure 2 - Profile of SWATH Ship with Tandem Struts

For the wave resistance predictions, the geometries of the body and strut(s) are represented by modified Chebychev series. The methods used to determine these series are the principal differences between the programs CLOSEFIT and SYNTHESIS. The program CLOSEFIT uses an integration technique, similar to that used to determine the coefficients of a Fourier series, to determine the coefficients of the Chebychev series based on the actual hull offsets. The program SYNTHESIS uses statical moments of the strut waterplane and body to define the coefficients of the Chebychev series. Typically, the CLOSEFIT program uses a series of 10 to 20 pairs of terms to represent a body or strut, where each pair of terms contains an even and an odd Chebychev function, in the same fashion that a Fourier series consists of pairs of sine and cosine functions. The program SYNTHESIS uses a series consisting of only three pairs of terms to represent the body and strut. This reduction in the number of terms results in savings of ten to forty times in the computation of the wave resistance in SYNTHESIS as compared to CLOSEFIT.

In addition to the effective power computations, the program SYNTHESIS also computes the delivered power which would be required to propel the SWATH ship. The propulsion calculations are performed by a subprogram SHPCMP and its attendant subprograms. SHPCMP determines the wake and thrust deduction for the design based on empirical relations derived from earlier SWATH propulsion experiments. The program then calculated the optimum four bladed Troost series propeller at the design speed, along with the off design performance of the propeller. This data along with the effective power computations is used to calculate the delivered power.

OVERVIEW OF CLOSEFIT

The CLOSEFIT SWATH program is composed of several modules, each of which consists of a number of subroutines and performs a specific function in the SWATH ship resistance calculation. Figure 1 presents a tree diagram which reflects the relationship among all subroutines.

Subroutine CHEB evaluates the coefficients of the Chebychev series, which represents the strut half thickness function and the body sectional area function in wave resistance equations. This is accomplished by calling READ, SCALE, COMPUT, CHECK and SURFACE. The offset tables are read in by Subroutine READ, then normalized by Subroutine SCALE. The driver Subroutine COMPUT calls Subroutine CHEV, which in turn calls SPLINE and SUMSPL to compute the coefficients of the Chebychev series. The integrals used to evaluate the strut Chebychev coefficients are as follows:

$$A_{sm} = \frac{2}{\pi} \int_{-\pi/2}^{\pi/2} t(\sin \theta) \cos(2m-1)\theta d\theta$$

$$B_{sm} = \frac{2}{\pi} \int_{-\pi/2}^{\pi/2} t(\sin \theta) \sin 2m\theta d\theta$$

where $t(\sin \theta)$ is a function adopted to describe the offsets of the strut or the body. The Cubic Spline Function Technique is used to fit an analytic curve to the discrete offset points. Subroutine SPLINE solves

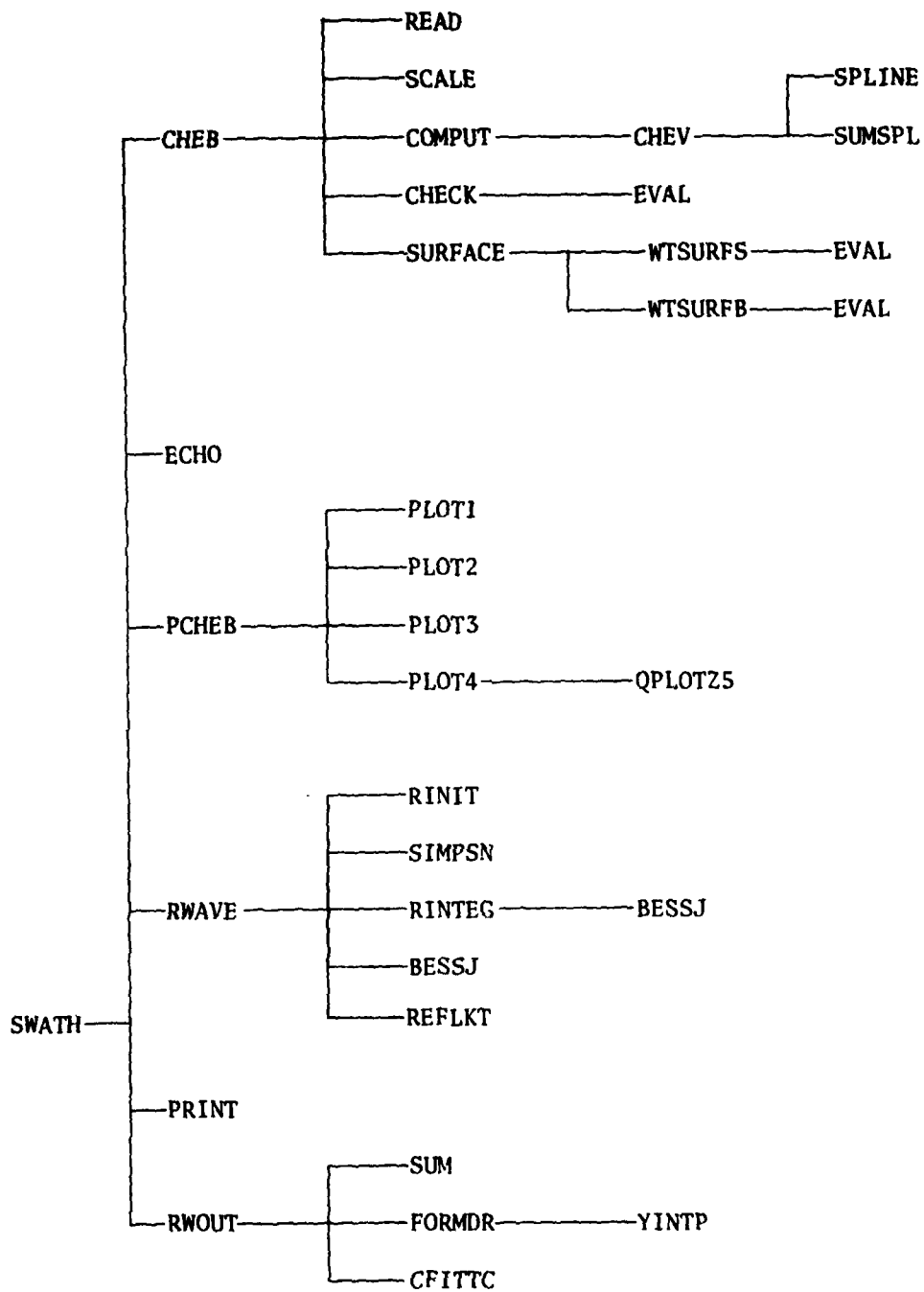


Figure 3 - Tree Diagram of CLOSEFIT SWATH Program

the set of simultaneous linear algebraic equations, which results from the spline fitting procedure and obtains the second derivatives of the analytical functions $t(\sin \theta)$ at each offset point. Next, Subroutine SUMSPL uses these second derivatives in a set of integration formulas based on the Filon Quadrature Technique to evaluate the above trigonometric integrals. Subroutine CHECK evaluates the offsets approximated from the Chebychev series in Subroutine COMPUT and prints out the predicted offsets along with the true offsets. This permits a check on the accuracy of the Chebychev series approximations.

Subroutine SURFACE computes the total SWATH ship wetted surface by calling WTSURFS and WTSURFB. Subroutine ECHO echoes the input data of the SWATH program on the printer as a check of its accuracy. Subroutine PCHEB plots the strut half thickness curve and body sectional area curve on the printer in order to validate the Chebychev series approximation.

A large amount of the execution time of this program is spent in the evaluation of the auxiliary functions, T and W, given by:

$$\left\{ \begin{array}{c} \frac{T_{Smn}}{(2m-1)(2n-1)} \\ \frac{W_{Smn}}{(2m)(2n)} \end{array} \right\} = \int_{\gamma_{0S}}^{\infty} \frac{d\alpha}{\alpha^2 \sqrt{\alpha^2 - \gamma_{0S}^2}} D \left(\alpha; \frac{2b}{L_S}, \gamma_{0S} \right) \\ \times F_S^2(\alpha) \left\{ \begin{array}{c} J_{2m-1}(\alpha) J_{2n-1}(\alpha) \\ J_{2m}(\alpha) J_{2n}(\alpha) \end{array} \right\},$$

$$\left\{ \begin{array}{c} \frac{T_{Bmn}}{(2m-1)(2n-1)} \\ \frac{W_{Bmn}}{(2m)(2n)} \end{array} \right\} = \int_{\gamma_{0S}}^{\infty} d\alpha \frac{\alpha^2}{\sqrt{\alpha^2 - \gamma_{0S}^2}} D \left(\alpha; \frac{2b}{L_S}, \gamma_{0S} \right) \\ \times F_B^2(\beta) \left\{ \begin{array}{c} J_{2m-1}(\beta) J_{2n-1}(\beta) \\ J_{2m}(\beta) J_{2n}(\beta) \end{array} \right\}$$

$$\left\{ \frac{T_{SBmn}}{(2m-1)(2n-1)} \right\} = \int_{\gamma_{0S}}^{\infty} \frac{d\alpha}{\sqrt{\alpha^2 - \gamma_{0S}}} D\left(\alpha; \frac{2b}{L_S}, \gamma_{0S}\right) \\ \times E_S(\alpha) E_B(\beta) \left\{ \begin{matrix} J_{2m-1}(\alpha) J_{2n-1}(\beta) \\ J_{2m}(\alpha) J_{2n}(\beta) \end{matrix} \right\}.$$

Due to the highly oscillatory nature of the integrands of these auxiliary functions, the integration range is divided into three separate intervals. For each interval, a particular numerical integration method is used to improve the accuracy of the calculation. To avoid the singularity in the denominator of the integrand, the integration variable (α) is replaced by $Z = \sqrt{\alpha - \gamma_{0S}}$ in the interval from γ_{0S} to $\gamma_{0S} + 1$. The numerical method used in this interval is Simpson's Rule, which uses 30 points. The central region of the integration process goes from $\alpha = \gamma_{0S} + 1$ to $\alpha = \alpha_{\max}$. The choice of $\alpha_{\max}$ is based on the requirement for accuracy in the numerical integration method used in this region, where a step size is computed at a given α , and a three-point Simpson's Rule is used until the upper integration limit of the region, $\alpha_{\max}$, is reached.

An analytical integration formula was developed to determine the effects of the final region, from $\alpha = \alpha_{\max}$ to $\alpha = \alpha_{l_{\max}}$, where $\alpha_{l_{\max}}$ is an empirical constant which defines the upper limit of the tail region. Beyond this point, the integration area is assumed to be negligible.

Subroutine PRINT will print the table of the T and W arrays, provided the flag JTEST is set other than zero.

Subroutine RWOUT calls SUM to evaluate the components of the wave

resistance, based on Lin and Day's equations¹. Form drag is evaluated by Subroutine FORMDR, which is based on experimental data. Frictional drag is estimated by calling Subroutine CFITTC, which is based on the C_F curve adopted by the International Towing Tank Conference (ITTC)². For each test speed, a table is printed showing all the components of total ship resistance and the effective power required to tow the SWATH ship at a constant speed.

¹References are listed on page 112.

OVERVIEW OF SYNTHESIS

The procedure of the SYNTHESIS SWATH program parallels the CLOSEFIT program in calculating total ship resistance, except in its approach to obtaining the coefficients of the Chebychev series. In SYNTHESIS, the waterplane area coefficients, waterplane moment, waterplane inertia, body prismatic and body moment are used to evaluate the Chebychev coefficients. The only other major difference between the two programs is that SYNTHESIS includes a propeller design and optimization program.

Figure 2 presents the tree diagram of all subroutines called by the SYNTHESIS program. The module CHEB is much simpler in SYNTHESIS than its CLOSEFIT counterpart due to its approach in evaluating the Chebychev coefficients through geometric identities as opposed to numerical integrations.

Modules RWAVE and ROUT are identical for both programs, and in both require a great deal of execution time.

Module FINDRG evaluates the fin drag of a SWATH ship. It is directly excerpted from coding documented in reference (3).

Module SHPCMP can determine propeller design and off-design performance if the propeller diameter is specified. The subroutines used in this module are excerpted from coding documented in reference (4).

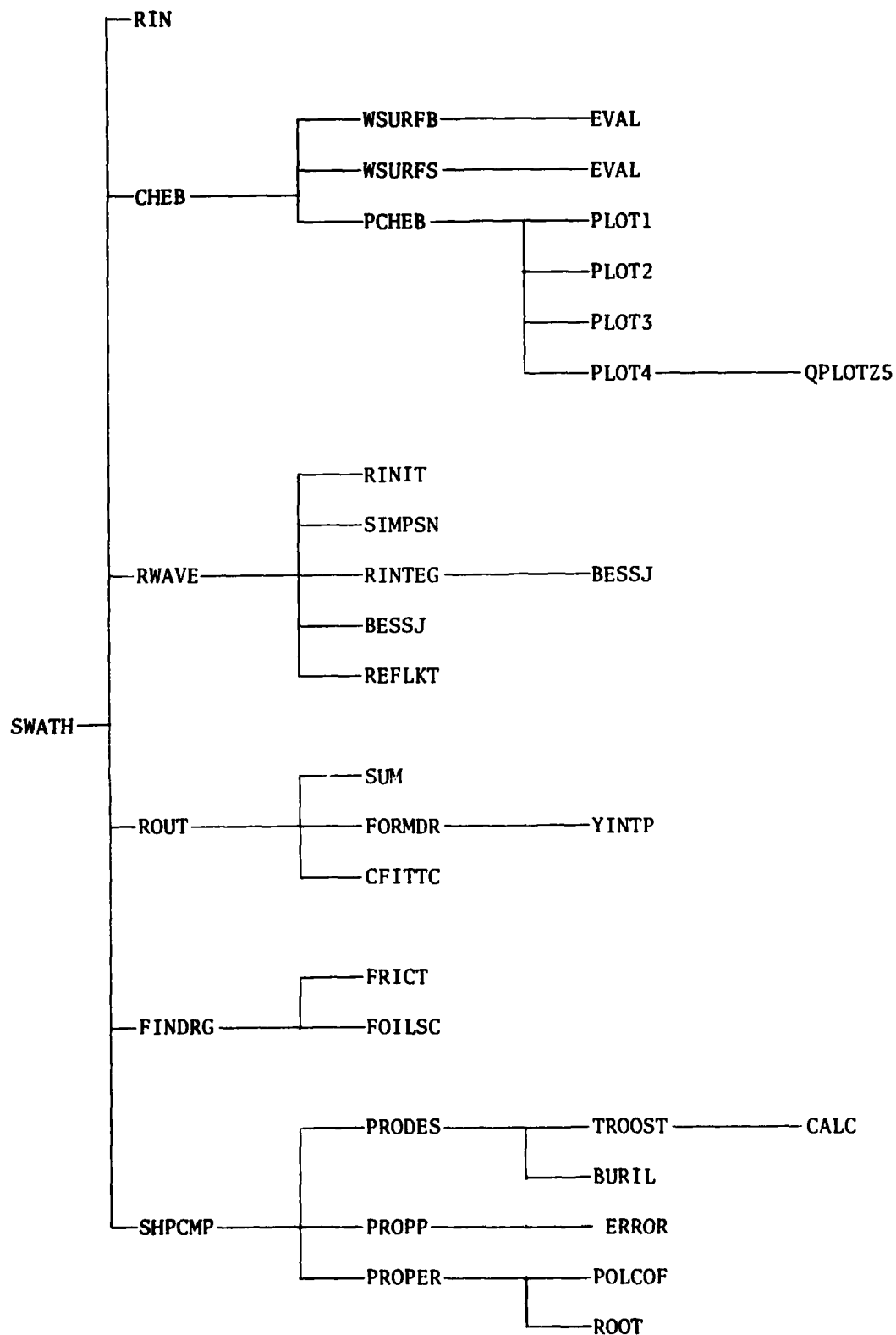


FIGURE 4 - Tree Diagram of SYNTHESIS SWATH Program

PROGRAM DOCUMENTATION OF CLOSEFIT

COMMON BLOCK DEFINITIONS FOR CLOSEFIT

COMMON/AUX/

PURPOSE: AUX stores the auxiliary function table.

NAME	TYPE	LENGTH	DEFINITION
TS	R	(10,10)	Auxiliary function T for strut
WS	R	(10,10)	Auxiliary function W for strut
TB	R	(10,10)	Auxiliary function T for body
WB	R	(10,10)	Auxiliary function W for body
TSB	R	(10,10)	Auxiliary function T for strut to body
WSB	R	(10,10)	Auxiliary function W for strut to body
TS12	R	(10,10)	Auxiliary function T for strut 1 to strut 2
WS12	R	(8,10,10)	Auxiliary function W for strut 1 to strut 2
TB12	R	(8,10,10)	Auxiliary function T for body 1 to body 2
WB12	R	(8,10,10)	Auxiliary function W for body 1 to body 2
TSB12	R	(8,10,10)	Auxiliary function T for strut 1 to body 2
WSB12	R	(8,10,10)	Auxiliary function W for strut 1 to body 2
TSBP	R	(10,10)	Auxiliary function T
WSBP	R	(10,10)	Auxiliary function W
TSB12P	R	(10,10)	Auxiliary function T
WSB12P	R	(10,10)	Auxiliary function W

COMMON/NAME/

PURPOSE: NAME stores a flag indicating whether the offset data are for the strut or the body and providing the alphanumeric information for the strut and the body.

NAME	TYPE	LENGTH	DEFINITION
INCHAR	I	8	INCHAR (1) = STRUT. Strut offset data will follow. INCHAR (1) = BODY. Body offset data will follow. INCHAR (2) to INCHAR (8). Hold the alphanumeric information for the strut or the body.
NAME	I	7	Temporary storage location for alphanumeric strut or body information.

COMMON/NORM/

PURPOSE: NORM stores the scaling factors for normalizing X and Y.

NAME	TYPE	LENGTH	DEFINITION
YMAX	R		Maximum value of Y
XDIVER	R		Divisor for normalizing X
YDIVER	R		Divisor for normalizing Y
XADDER	R		Number to be added to normalize X

COMMON/OMEGA/

PURPOSE: OMEGA stores the variables used in evaluating the auxiliary function.

NAME	TYPE	LENGTH	DEFINITION
NMAX	I		Maximum order of Chebychev Series
HSOLS	R		Ratio of draft of strut to length of strut (h_s/L_s)
HBOLB	R		Ratio of depth of submergence to length of body (h_b/L_b)
GAMAOS	R		$\gamma_{os} = \frac{1}{2} g L_s / v^2$
GAMAOB	R		$\gamma_{ob} = \frac{1}{2} g L_b / v^2$
NOSEPS	I		Number of different hull separation distances
SEPDIS	R	(8)	Hull separation distance b (ft)
SEP	R	(8)	$2b/(\gamma_{os} \cdot L_s)$
CSTRUT	R	(8)	Distance of CL of strut to CL of body
CSTRT2	R	(8)	Distance of CL of 2nd strut (if present) from CL of body
GOSQ	R		γ_{os}^2
PHIS	R		$2(\frac{h_s}{L_s})/\gamma_{os}$
PHIB	R		$2(\frac{h_b}{L_b})/\gamma_{ob}$
RATIOI	R		γ_{ob}/γ_{os}

COMMON/OUT/

PURPOSE: OUT defines and stores physical constants and geometric parameters.

NAME	TYPE	LENGTH	DEFINITION	SET VALUES
HS	R		Draft of strut h_s (ft)	
HB	R		Depth of submergence from surface to body centerline h_b (ft)	
XLS	R		Length of strut L_s (ft)	
XLB	R		Length of body L_b (ft)	
TSMAX	R		Maximum thickness of strut t_{max} (ft)	
AX	R		Maximum cross-sectional area of body A_x (ft ²)	
PI	R		3.1415926535897	
G	R		Acceleration due to gravity g (ft/sec ²)	32.155
RHO	R		Density of water ρ (lb x sec ² /ft ⁴)	1.9367
GNU	R		kinematic viscosity ν (ft ² /sec)	1.2970×10^5
WETS	R		Wetted surface area of strut S_s (ft ²)	
WETB	R		Wetted surface area of body S_b (ft ²)	
WTSURF	R		Total wetted surface area S_t (ft ²)	
VMFPS	R		Test velocity V (fps)	
DELCF	R		Correction allowance ΔC_F	0.0005
TITLE	I	(8)	Array of characters containing the title of the test	

COMMON/PLOT/

PURPOSE: PLOT stores the constants and characters for the plotting routine

NAME	TYPE	LENGTH	DEFINITION	SET VALUES
NFIRST	I		Position in the arrays of the first ordered pair to be plotted	1
NLAST	I		Position in the arrays of the last ordered pair to be plotted	101
NPOINT	I		NPOINT equals "1" each point from NFIRST to NLAST is to be plotted, "2" if every other point is to be plotted, etc.	1
NMAX	R		Value of abscissa at right-most grid line	1.0
XMIN	R		Value of abscissa at left-most grid line	-1.0
NSCLI	L		Logical value (Should be false if PLOTI has not been called and standard grid is desired)	
NCHAR	I		Number of valid characters in label	
NSCALE	I	(4)	Printing scale factor of ordinate	
PCHAR	I	(2)	Plotting characters	

COMMON/PSI/

PURPOSE: PSI stores the parameters for the integration procedure

NAME	TYPE	LENGTH	DEFINITION	SET VALUES
NPTSZ	I		Number of integration steps from γ to $\gamma+1$ (odd number) os os	31
PTSFA	R		Scaling factor of step size in integrating from $\alpha_{\max}$ to $\alpha_{\ell\max}$	10.0
EXPN	R		Empirical constant for intergration to stop	7.0
NALMAX	I		Maximum number of integration steps from $\gamma+1$ to $\alpha_{\max}$ os	300
NAL	I		Counter of integration steps	
TAIL	R		Integration made from α to $\alpha_{\ell\max}$	
ALFA	R		Integrating variable (α)	
ALMAX	R		Upper limit of α for integration	
NSTEPS	I		Number of integration steps from $\alpha_{\max}$ to $\alpha_{\ell\max}$	

COMMON/XRPLTF/

PURPOSE: XRPLTF stores the values of variables for the plotting routine

NAME	TYPE	LENGTH	DEFINITION
XL	R		Value of abscissa at left-most grid line
XH	R		Value of abscissa at right-most grid line
YL	R		Value of ordinate at bottom grid line
YH	R		Value of ordinate at top grid line
XMOV	R		Abscissa index increment number for array GRAF
YMOV	R		Ordinate index increment number for array GRAF

COMMON/XRPLTG/

PURPOSE: XEPLTG stores the values and characters of variables for the plotting routine

NAME	TYPE	LENGTH	DEFINITION
GRAF	I	(11,204)	Array containing the image to be plotted

COMMON/XRPLQTQ/

PURPOSE: XRPLQTQ stores the constants and characters for the plotting routine

NAME	TYPE	LENGTH	DEFINITION	SET VALUES
II	I		Ordinate scale factor	
JJ	I		Ordinate scale factor	
KK	I		Ordinate scale factor	
LL	I		Ordinate scale factor	
NHL	I		Number of horizontal grid lines	6
NSBH	I		Number of spaces between adjacent horizontal grid lines	10
NVL	I		Number of vertical grid lines	11
NSBV	I		Number of spaces between adjacent vertical grid lines	10
NCHAR	I		Plotting character of horizontal grid lines	1H.
VCHAR	I		Plotting character of vertical grid lines	1H.
IY	I		Number of vertical spaces	
IX	I		Number of horizontal spaces	
V	L		Logical variable, = . TRUE . when the maximum and minimum values of the ordinate are determined	
H	L		Logical variable, = . TRUE . when the maximum and minimum values of the abscissa are determined	

SUBROUTINE DOCUMENTATION FOR CLOSEFIT

NAME: PROGRAM SWATH

PURPOSE: Program SWATH is the main program for the
CLOSEFIT resistance prediction of Small-Water-
plane-Area Twin-Hull Ships

CALLING SERVICE: Program SWATH (INPUT, OUTPUT, TAPE5 = INPUT
TAPE6 = OUTPUT, TAPE8)

ARGUMENTS: NONE

COMMON BLOCKS: OUT, AUX, OMEGA, PSI, COEFS

SUBROUTINES CALLED: CHEB, ECHO, PCHEB, PRINT, RWAVE, RWOUT

COMMENTS:

NAME:	SUBROUTINE BESSJ
PURPOSE:	SUBROUTINE BESSJ evaluates the Bessel function from order 0 to order N.
CALLING SEQUENCE:	CALL BESSJ (X, N, VJ)
ARGUMENTS:	X Argument of the Bessel function N Maximum order of the Bessel function VJ Array holding (N+1) values of the Bessel function of order zero up to N, where $VJ(0) = J_0(X)$. . . $VJ(N) = J_N(X)$
COMMON BLOCKS:	NONE
SUBROUTINE CALLED:	NONE
CALLED BY:	RINTEG, RWAVE
COMMENTS:	

NAME: FUNCTION CFITTC

PURPOSE: Function CFITTC determines the ITTC
frictional resistance coefficient C_f

CALLING SEQUENCE: CF = CFITTC (RN)

ARGUMENT: RN Reynolds number R_n at test condition

COMMON BLOCKS: NONE

SUBROUTINE CALLED: NONE

CALLED BY: RWOUT

COMMENTS:

$$C_f = \frac{0.075}{[\log_{10}(R_n) - 2]^2}$$

NAME:	SUBROUTINE CHEB	
PURPOSE:	Subroutine CHEB computes from offset tables the coefficients of the Chebychev Series, which is an approximation of the outline of the waterplane area or body cross-sectional area curve.	
CALLING SEQUENCE:	CALL CHEB (AMC, BMC, MMAX)	
ARGUMENTS:	AMC	Coefficients of the Chebychev Cosine Series
	BMC	Coefficients of the Chebychev Sine Series
	MMAX	Maximum order of the Chebychev Series
COMMON BLOCKS:	NAVME, OFFSET	
SUBROUTINES CALLED:	READ, SCALE, COMPUT, CHECK, SURFACE	
CALLED BY:	SWATH	
COMMENTS:		

NAME:	SUBROUTINE CHEB	
PURPOSE:	Subroutine CHEB computes from offset tables the coefficients of the Chebychev Series, which is an approximation of the outline of the waterplane area or body cross-sectional area curve.	
CALLING SEQUENCE:	CALL CHEB (AMC, BMC, MMAX)	
ARGUMENTS:	AMC	Coefficients of the Chebychev Cosine Series
	BMC	Coefficients of the Chebychev Sine Series
	MMAX	Maximum order of the Chebychev Series
COMMON BLOCKS:	NAME, OFFSET	
SUBROUTINES CALLED:	READ, SCALE, COMPUT, CHECK, SURFACE	
CALLED BY:	SWATH	
COMMENTS:		

NAME:	SUBROUTINE CHECK	
PURPOSE:	Subroutine CHECK evaluates the Chebychev Series from the Chebychev coefficients obtained from Subroutine COMPUT and prints out the evaluated offset table for manual checking.	
CALLING SEQUENCE:	CALL CHECK (AMC, BMC, MMAX, ISBODY)	
ARGUMENTS:	AMC	Coefficients of the Chebychev Cosine Series
	BMC	Coefficients of the Chebychev Sine Series
	MMAX	Maximum order of the Chebychev polynomial
	ISBODY	Logical flag, indicating "Is this for body?"
	ISBODY =	$\left\{ \begin{array}{l} \text{. TRUE . for body} \\ \text{. FALSE . for strut} \end{array} \right.$
COMMON BLOCKS:	OFFSET, NAME	
SUBROUTINE CALLED:	EVAL	
CALLED BY:	CHEB	
COMMENTS:		

NAME:	SUBROUTINE CHEV	
PURPOSE:	Subroutine CHEV uses the offsets to compute the Chebychev coefficients via spline approximation and numerical integration.	
CALLING SEQUENCE:	AMC	Coefficients of the Chebychev Cosine Series
	BMC	Coefficients of the Chebychev Sine Series
	MMA	Maximum order of the Chebychev polynomial
	ISBODY	Logical flag, indicating "Is this for body?"
	ISBODY =	$\begin{cases} \text{. TRUE . for body} \\ \text{. FALSE . for strut} \end{cases}$
COMMON BLOCKS:	OFFSET, NAME	
SUBROUTINES CALLED:	SPLINE, FUNCTION SUMPSPL	
CALLED BY:	COMPUT	
COMMENTS:		

NAME:	SUBROUTINE COMPUT	
PURPOSE:	Subroutine COMPUT initiates the computation of the Chebychev coefficients.	
CALLING SEQUENCE:	CALL COMPUT (AMC, BMC, MMAX, ISBODY)	
ARGUMENTS:	AMC	Coefficients of the Chebychev Cosine Series
	BMC	Coefficients of the Chebychev Sine Series
	MMAX	Maximum order of the Chebychev polynomials
	ISBODY	Logical flag, indicating "Is this for body?"
	ISBODY = {	. TRUE . for body
		. FALSE . for strut
COMMON BLOCKS:	OFFSET, NAME, NORM	
SUBROUTINE CALLED:	CHEV	
CALLED BY:	CHEB	
COMMENTS:		

NAME:	SUBROUTINE ECHO
PURPOSE:	Subroutine ECHO prints a summary of input data and the Chebychev coefficients
CALLING SEQUENCE:	CALL ECHO (NLOC)
ARGUMENT:	NLOC Number of strut locations to be tested
COMMON BLOCKS:	AUX, OMEGA, OUT
SUBROUTINE CALLED:	NONE
CALLED BY:	SWATH
COMMENTS:	

NAME:	SUBROUTINE EVAL	
PURPOSE:	Subroutine EVAL evaluates the Chebychev Series at a given station.	
CALLING SEQUENCE:	CALL EVAL (XX, MMAX, FFSET, AMC, BMC)	
ARGUMENTS:	XX	Value of X station on a range of -1 to +1
	MMAX	Maximum order of the Chebychev polynomial
	FFSET	Offset approximated by the Chebychev Series
	AMC	Coefficients of the Chebychev Cosine Series
	BMC	Coefficients of the Chebychev Sine Series
COMMON BLOCKS:	NONE	
SUBROUTINE CALLED:	NONE	
CALLED BY:	CHECK	
COMMENTS:		

NAME: FUNCTION FORMDR

PURPOSE: Function FORMDR evaluates the form drag coefficient, based on experimental data

CALLING SEQUENCE: FOR = FORMDR (VL)

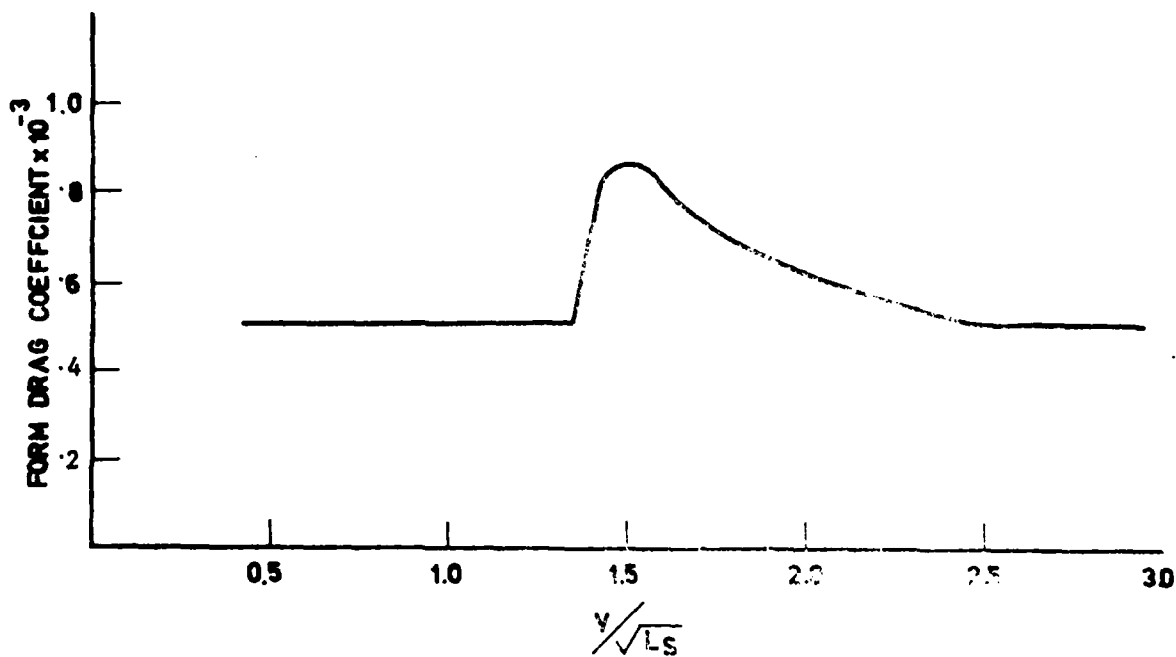
ARGUMENT: VL Ratio of ship speed to length of strut $V/\sqrt{L_s}$

COMMON BLOCKS: NONE

SUBROUTINE CALLED: FUNCTION YINTP

CALLED BY: ROUT

COMMENTS: Form drag is assumed to be 0.5×10^{-3} for $V/\sqrt{L_s}$ less than 1.325 and greater than 2.5



NAME:	SUBROUTINE PCHEB
PURPOSE:	Subroutine PCHEB plots by line printer the body sectional area curve and waterplane outline curve from the given Chebychev coefficients
CALLING SEQUENCE:	CALL PCHEB (AS, BS, AB, BB, NN, TITLE)
ARGUMENTS:	AS Coefficients of Chebychev Sine Series for strut BS Coefficients of Chebychev Cosine Series for strut AB Coefficients of Chebychev Sine Series for body BB Coefficient of Chebychev Cosine Series for body NN Maximum order of Chebychev Series TITLE Array containing the alphanumeric characters of the title of the experiment
COMMON BLOCKS:	PLOT, XRPLQT
SUBROUTINES CALLED:	PLOT1, PLOT2, PLOT3, PLOT4
CALLED BY:	CHEB
COMMENTS:	

NAME: SUBROUTINE PLOT1

PURPOSE: Subroutine PLOT1 sets up spacing and determines the values of the axes.

CALLING SEQUENCE: CALL PLOT 1 (NSCALE, A, B, C, D, E, F)

ARGUMENTS:

NSCALE	Integer array defined as follows:
NSCALE (1) =	I, if printed values of the ordinate are $10^{**} I$ times the actual value
NSCALE (2) =	J, if printed values of the ordinate are $10^{**} J$ times the actual value
NSCALE (3) =	K, if printed values of the abscissa are $10^{**} K$ times the actual values
NSCALE (4) =	L, if printed values of the abscissa are $10^{**} L$ times the actual value
A	Integer number of horizontal grid lines
B	Integer number of spaces beyond each horizontal grid line to the next grid line
C	Integer number of vertical grid lines
D	Integer number of spaces beyond each vertical grid line to the next grid line
E	Horizontal grid character
F	Vertical grid character

COMMON BLOCK: XRPLQT

SUBROUTINE CALLED: NONE

CALLED BY PCHEB

COMMENTS:

NAME:	SUBROUTINE PLOT2	
PURPOSE:	Subroutine PLOT2 examines the minimum and maximum values of the abscissa and the ordinate and establishes an internal formula for computing location in the image region corresponding to the point to be plotted.	
CALLING SEQUENCE:	CALL PLOT2 (XMAX, XMIN, YMAX, YMIN, NSCLI)	
ARGUMENTS:	XMAX	Value of abscissa at right-most grid line
	XMIN	Value of abscissa at left-most grid line
	YMAX	Value of ordinate at top grid line
	YMIN	Value of ordinate at bottom grid line
	NSCLI	Logical flat (should be .FALSE., if PLOT1 has not been called and standard grid is desired)
COMMON BLOCKS:	XRPLTF, SRPLQT, XRPLTG	
SUBROUTINE CALLED:	NONE	
CALLED BY:	PCHEB	
COMMENTS:		

NAME: SUBROUTINE PLOT3

PURPOSE: Subroutine PLOT3 assigns an alpha-character to each point to be plotted.

CALLING SEQUENCE: CALL PLOT3 (PCHAR, X, Y, SDATA, FDATA, DDATA)

ARGUMENTS:

PCHAR	Plotting character
X	Array containing the X coordinates to be plotted
Y	Array containing the Y coordinates to be plotted
SDATA	Integer position in the arrays of the first ordered pair to be plotted
FDATA	{ 1 if each point from SDATA to DDATA is to be plotted 2 if every other point is to be plotted 3 if every third point is to be plotted
DDATA	Integer position in the array of the last ordered pair to be plotted

COMMON BLOCKS: XRPLTF, XRPLTG

SUBROUTINE CALLED: NONE

CALLED BY: PCHEB

COMMENTS:

NAME:	SUBROUTINE PLOT4				
PURPOSE:	Subroutine PLOT4 prints the image of the completed graph on the printer, including the values of the abscissa and the ordinate at the grid lines outside the bottom and left edge of the graph.				
CALLING SEQUENCE:	CALL PLOT4 (MCHAR, NCHAR)				
ARGUMENTS:	<table border="0"> <tr> <td>MCHAR</td> <td>Single dimension array containing alpha-characters to be plotted at the left of the graph</td> </tr> <tr> <td>NCHAR</td> <td>Number of valid characters in MCHAR</td> </tr> </table>	MCHAR	Single dimension array containing alpha-characters to be plotted at the left of the graph	NCHAR	Number of valid characters in MCHAR
MCHAR	Single dimension array containing alpha-characters to be plotted at the left of the graph				
NCHAR	Number of valid characters in MCHAR				
COMMON BLOCKS:	XRPLOTF, XRPLOTG, XRPLOTQ				
SUBROUTINE CALLED:	QPLOTZ5				
CALLED BY:	PCHEB				
COMMENTS:					

NAME:	SUBROUTINE PRINT								
PURPOSE:	Subroutine PRINT prints the auxiliary function tables of T and W.								
CALLING SEQUENCE:	CALL PRINT (I, NLOC2)								
ARGUMENTS:	<table border="0"> <tr> <td>I</td> <td>Index indicating strut location</td> </tr> <tr> <td>NLOC2</td> <td>Flag indicating the presence of a second strut</td> </tr> <tr> <td>NLOC2</td> <td>= { 0 if single strut</td> </tr> <tr> <td></td> <td>1 if tandem struts</td> </tr> </table>	I	Index indicating strut location	NLOC2	Flag indicating the presence of a second strut	NLOC2	= { 0 if single strut		1 if tandem struts
I	Index indicating strut location								
NLOC2	Flag indicating the presence of a second strut								
NLOC2	= { 0 if single strut								
	1 if tandem struts								
COMMON BLOCKS:	OUT, AUX, OMEGA								
SUBROUTINE CALLED:	NONE								
CALLED BY:	SWATH								
COMMENTS:									

NAME: SUBROUTINE QPLOTZ5

PURPOSE: Subroutine QPLOTZ5 calculates the scaling information needed to generate the format to label the left-hand side of the program.

CALLING SEQUENCE: CALL QPLOTZ5 (PDQ)

ARGUMENT: PDQ Scaling factor for ordinate plot

COMMON BLOCKS: XRPLOTF, XRPLOTQ

SUBROUTINES CALLED: NONE

COMMENTS:

NAME: SUBROUTINE READ

PURPOSE: Subroutine READ reads the offset data of the strut or body and checks for error.

CALLING SEQUENCE: CALL READ

ARGUMENTS: NONE

COMMON BLOCKS: OFFSET, NAME, NORM

SUBROUTINE CALLED: NONE

CALLED BY: CHEB

COMMENTS:

NAME:	SUBROUTINE REFLKT
PURPOSE:	Subroutine REFLKT reflects all the symmetrical matrices of T and W.
CALLING SEQUENCE:	CALL REFLKT
ARGUMENTS:	NONE
COMMON BLOCKS:	AUX, OMEGA
SUBROUTINE CALLED:	NONE
CALLED BY:	RWAVE
COMMENTS:	

NAME:	SUBROUTINE RINIT
PURPOSE:	Subroutine RINIT initializes the T and W arrays to zero.
CALLING SEQUENCE:	CALL RINIT
ARGUMENT:	NONE
COMMON BLOCKS:	AUX, OMEGA
SUBROUTINE CALLED:	NONE
CALLED BY:	RWAVE
COMMENTS:	

NAME: SUBROUTINE RINTEG

PURPOSE: Subroutine RINTEG evaluates the integrand for T and W functions.

CALLING SEQUENCE: CALL RINTEG (ALFA, B, D, NLOC2, WTINT, SEPCOS, SQ)

ARGUMENTS:

ALFA	Integrating variable α
B	CSTRUT(I)/XLS
D	CSTRT2(I)/XLS
NLOC2	Flag indicating the presence of a second strut
NLOC2 =	$\begin{cases} 0 & \text{if single strut} \\ 1 & \text{if tandem struts} \end{cases}$
WTINT	Weighting constant for the integrand
SEPCOS	Value of the cosine function in the integrand
SQ	Value of part of the integrand
	$\frac{1}{(\alpha^2 - \gamma^2)^{\frac{1}{2}}}$

COMMON BLOCKS: AUX, OMEGA

SUBROUTINE CALLED: BESSJ

CALLED BY: RWAVE

COMMENTS:

NAME: SUBROUTINE RWAVE

PURPOSE: Subroutine RWAVE computes the auxiliary function of T and W.

CALLING SEQUENCE: CALL RWAVE (B, D, NLOC2)

ARGUMENTS:

B	CSTRUT (I)/XLS
D	CSTRT2 (I)/XLS
NLOC2	Flag indicating the presence of a second strut
NLOC2 =	$\begin{cases} 0 & \text{if single strut} \\ 1 & \text{if tandem struts} \end{cases}$

COMMON BLOCKS: OUT, AUX, OMEGA, PSI

SUBROUTINES CALLED: RINIT, SIMPSN, RINTEG, BESSJ, REFLKT

CALLED BY: SWATH

COMMENTS:

Typical Auxiliary Functions

$$\left\{ \frac{T_{smn}}{(2m-1)(2n-1)} \right\} = \int_{\gamma_{0s}}^{\infty} \frac{d\alpha}{\alpha^2 \sqrt{\alpha^2 - \gamma_{0s}^2}} D \left(\alpha, \frac{2b}{L_s}, \gamma_{0s} \right) \times E_s^2(\alpha) \left\{ \frac{J_{2m-1}(\alpha) J_{2n-1}(\alpha)}{J_{2m}(\alpha) J_{2n}(\alpha)} \right\},$$

where

$$D = 1 + \cos \left[\left(\frac{2b}{L_s} \right) \left(\frac{2}{\gamma_{0s}} \right) \alpha \sqrt{\alpha^2 - \gamma_{0s}^2} \right],$$

$$E_s = 1 - e^{-2(m_s/L_s)(\alpha^2/\gamma_{0s})},$$

$$\gamma_{0s} = \frac{gL_s}{(2U^2)}.$$

NAME:	SUBROUTINE RWOUT						
PURPOSE:	Subroutine RWOUT computes and prints drag coefficients and total resistance of the test model.						
CALLING SEQUENCE:	CALL RWOUT (I, NLOC2)						
ARGUMENTS:	<table border="0"> <tr> <td>I</td> <td>Index indicating strut location</td> </tr> <tr> <td>NLOC2</td> <td>Flag indicating the presence of a second strut</td> </tr> <tr> <td>NLOC 2 =</td> <td> $\begin{cases} 0 & \text{if single strut} \\ 1 & \text{if tandem struts} \end{cases}$ </td> </tr> </table>	I	Index indicating strut location	NLOC2	Flag indicating the presence of a second strut	NLOC 2 =	$\begin{cases} 0 & \text{if single strut} \\ 1 & \text{if tandem struts} \end{cases}$
I	Index indicating strut location						
NLOC2	Flag indicating the presence of a second strut						
NLOC 2 =	$\begin{cases} 0 & \text{if single strut} \\ 1 & \text{if tandem struts} \end{cases}$						
COMMON BLOCKS:	OUT, AUX, OMEGA						
SUBROUTINES CALLED:	SUM, FUNCTION FORMDR, FUNCTION CFITTC						
CALLED BY:	SWATH						
COMMENTS:							

NAME:	SUBROUTINE SCALE
PURPOSE:	Subroutine SCALE computes scaling factors and scales the input for length between -1.0 and +1.0 maximum beam and sectional area to 1.0.
CALLING SEQUENCE:	CALL SCALE (ISBODY)
ARGUMENT:	ISBODY Logical flag indicating, "Is this for body?" ISBODY = { . TRUE . for body . FALSE . for strut
COMMON BLOCKS:	OFFSET, NAME, NORM
SUBROUTINE CALLED:	NONE
CALLED BY:	CHEB
COMMENTS:	

NAME: SUBROUTINE SIMPSN

PURPOSE: Subroutine SIMPSN sets up Simpson's multipliers for numerical integration by Simpson's rule.

CALLING SEQUENCE: CALL SIMPSN (NPTS, SIMP)

ARGUMENTS: NPTS Number of integration steps
SIMP Array containing Simpson's multipliers

COMMON BLOCKS: NONE

SUBROUTINE CALLED: NONE

CALLED BY: RWAVE

COMMENTS:

NAME:	SUBROUTINE SPLINE	
PURPOSE:	Subroutine SPLINE fits smooth spline segments through a given set of discrete data points.	
CALLING SEQUENCE:	CALL SPLINE (X, Y, D, N, BC, KODE, KCODE, ISBODY)	
ARGUMENTS:	X	Value of abscissas of offset data
	Y	Value of ordinates of offset data
	D	Value of second derivatives at a point
	N	Total number of offset data points
	BC	Value of boundary conditions 0 if user is specifying boundary conditions
	KODE =	{ 1 if user is taking an extrapolation of second derivatives as boundary conditions
	KCODE =	{ 0 if there is no printout from spline routine 1 if spline prints abscissas, ordinates and second derivatives
	ISBODY	Logical flag indicating, "Is this for body?"
	ISBODY =	{ . TRUE . for body . FALSE . for strut
COMMON BLOCKS:	NONE	
SUBROUTINE CALLED:	NONE	
CALLED BY:	CHEV	
COMMENTS:		

NAME: SUBROUTINE SUM

PURPOSE: Subroutine SUM computes sums of the form:

$$\sum_{m=1}^M \sum_{n=1}^N \{A_{sm} A_{sn} T_{smn} + B_{sm} B_{sn} W_{smn}\}$$

CALLING SEQUENCE: CALL SUM (L, SUM1S, SUM1B, SUM1SB, SUM12S, SUM12B, SUM12SB)

ARGUMENTS:

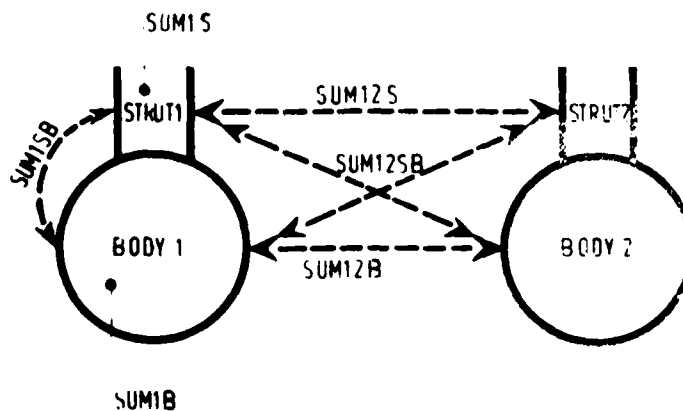
L	Number of different hull separation distances
SUM1S	Partial sum for strut 1
SUM1B	Partial sum for body 1
SUM1SB	Partial sum for interaction between strut 1 and body 1
SUM12S	Partial sum for interaction between strut 1 and strut 2
SUM12B	Partial sum for interaction between body 1 and body 2
SUM12SB	Partial sum for interactions between strut 1 and body 2 or strut 2 and body 1

COMMON BLOCKS: AUX, OMEGA, COEFS

SUBROUTINE CALLED: NONE

CALLED BY: RWOUT

COMMENTS:



NAME: FUNCTION SUMSPL

PURPOSE: Function SUMSPL uses the spline coefficients and integrates the trigonometric integrals $F(X)*\sin(AK*X)$ or $F(X)*\cos(AK*X)$.

CALLING SEQUENCE: SUM = SUMSPL (X, Y, S, N, AK, KODE)

ARGUMENTS:

X	Values of abscissas of offset data
Y	Values of ordinates of offset data
S	Values of second derivatives
N	Total number of data points
AK	$(2m-1)$ or $(2m)$, $m = 1, 2, 3 \dots MMAX$
KODE =	$\begin{cases} +1 & \text{if sine integration desired} \\ -1 & \text{if cosine integration desired} \end{cases}$

COMMON BLOCKS: NONE

SUBROUTINE CALLED: NONE

CALLED BY: CHEV

NAME:	SUBROUTINE SURFACE
PURPOSE:	Subroutine SURFACE finds the wetted surface area of the strut or body as well as the waterplane area or displaced volume.
CALLING SEQUENCE:	CALL SURFACE (AMC, BMC, MMAX, ISBODY)
ARGUMENTS:	AMC Coefficients of the Chebychev Cosine Series BMC Coefficients of the Chebychev Sine Series MMAX Maximum order of the Chebychev polynomials ISBODY Logical flag indicating "Is this for body?" ISBODY = { . TRUE . for body . FALSE . for strut
COMMON BLOCKS:	OUT, NAME
SUBROUTINES CALLED:	WTSURFS, WTSURFB
CALLED BY:	CHEB
COMMENTS:	

NAME: SUBROUTINE WTSURFB

PURPOSE: Subroutine WTSURFB determines the surface area of a body of revolution given by the Chebychev coefficients.

CALLING SEQUENCE: CALL WTSURFB (AMC, BMC, MMAX)

ARGUMENTS: AMC Coefficients of the Chebychev Cosine Series
 BMC Coefficients of the Chebychev Sine Series
 MMAX Maximum order of the Chebychev Series

COMMON BLOCK: OUT

SUBROUTINE CALLED: EVAL

CALLED BY: SURFACE

COMMENTS:

NAME:	SUBROUTINE WTSURFS	
PURPOSE:	Subroutine WTSURFS determines the wetted surface area of a strut whose thickness distribution is given by the Chebychev coefficients.	
CALLING SEQUENCE:	CALL WTSURFS (AMC, BMC, MMAX)	
ARGUMENTS:	AMC	Coefficients of Chebychev Cosine Series
	BMC	Coefficients of Chebychev Sine Series
	MMAX	Maximum order of Chebychev Series
COMMON BLOCK:	OUT	
SUBROUTINE CALLED:	EVAL	
CALLED BY:	SURFACE	
COMMENTS:		

NAME: FUNCTION YINTP

PURPOSE: Function YINTP interpolates through a set of discrete data.

CALLING SEQUENCE: FR = YINTP (XA, X, Y, N)

ARGUMENTS: XA Point to be interpolated
X Array of ordinate data
Y Array of abscissa data
N Number of data points

COMMON BLOCKS: NONE

SUBROUTINE CALLED: NONE

CALLED BY: FORMDR

COMMENTS: Uses linear interpolation

PROGRAM DOCUMENTATION OF SYNTHESIS

COMMON BLOCK DEFINITIONS FOR SYNTHESIS

COMMON/AUX/

PURPOSE: AUX stores the auxiliary wave resistance function table.

NAME	TYPE	LENGTH	DEFINITION
TS	R	(3,3)	Auxiliary function T for strut
WS	R	(3,3)	Auxiliary function W for strut
TB	R	(3,3)	Auxiliary function T for body
WB	R	(3,3)	Auxiliary function W for body
TSB	R	(3,3)	Auxiliary function T for strut to body
WSB	R	(3,3)	Auxiliary function W for strut to body
TS12	R	(3,3)	Auxiliary function T for strut 1 to strut 2
WS12	R	(3,3)	Auxiliary function W for strut 1 to strut 2
TB12	R	(3,3)	Auxiliary function T for body 1 to body 2
WB12	R	(3,3)	Auxiliary function W for body 1 to body 2
TSB12	R	(3,3)	Auxiliary function T for strut 1 to body 2
WSB12	R	(3,3)	Auxiliary function W for strut 1 to body 2
TSBP	R	(3,3)	Auxiliary function T
WSBP	R	(3,3)	Auxiliary function W
TSB12P	R	(3,3)	Auxiliary function T
WSB12P	R	(3,3)	Auxiliary function W

COMMON/BARCK/IE /

PURPOSE: BARCK stores the variables which define the propulsor design condition.

NAME	TYPE	LENGTH	DEFINITION
IE	I		Error message pointer

IE = {

- 0,if program ran in normal fashion
- 100,if limiting value of P/D has been used
- 200,if there is an iteration malfunction -- PD(2)
used for P/D
- 3,if maximum BAR, 0.95, used -- marine screw
- 4,if minimum BAR, .4, used -- marine screw
- 5,if maximum tip speed used -- air screw
- 6,if blade number is outside program limit --
closest number used

COMMON/CDESC/

PURPOSE: CDESC stores the values of constants which define water properties.

NAME	TYPE	LENGTH	DEFINITION
RHOD	R	(3)	Water density array
ANUD	R	(3)	Kinematic viscosity array
IP	I		Pointer indicating the particular water condition used. Set IP = 1 for this program to obtain sea water condition.

COMMON/COEFS/

PURPOSE: COEFS stores the values of the Chebychev coefficients for the strut and body.

NAME	TYPE	LENGTH	DEFINITION
ASM	R	(3)	Coefficients of Chebychev Sine Series for strut
BSM	R	(3)	Coefficients of Chebychev Cosine Series for strut
ABM	R	(3)	Coefficients of Chebychev Sine Series for body
BBM	R	(3)	Coefficients of Chebychev Cosine Series for body
MMAX	I		Maximum order of Chebychev Series

COMMON/CPROP/

PURPOSE: CPROP stores the values of the variables and constants used in the propulsor design routine.

NAME	TYPE	LENGTH	DEFINITION
IPTYP	I	(3)	Indicating propeller type
DIAM	R	(3)	Diameter of propeller (ft)
VADES	R	(3)	Speed of advance of propulsor at design condition (kts)
RPMDES	R	(3)	RPM of propeller at design point
TDES	R	(3)	Thrust of propulsor system (lbs)
QDES	R	(3)	Torque of propulsor system (ft-lbs)
EFFDES	R	(3)	Efficiency of propulsor at design condition
OMWTD	R	(3)	Thrust wake $(1-w_T)$
CAV	R	(3)	Percent back cavitation allowed on water screw
Z	I	(3)	Number of blades of propeller
H	R	(3)	Depth of water to centerline of propulsor shaft
BAR	R	(3)	Blade area ratio
PD	R	(3)	P/D of propulsor
J	R	(3)	J of propulsor
IERROR	I	(3)	Error message indicator
KT	R	(3)	Thrust coefficient (K_T)
KQ	R	(3)	Torque coefficient (K_Q)

COMMON/INPUT/

PURPOSE: INPUT stores input data.

NAME	TYPE	LENGTH	DEFINITION
XLS	R		Length of strut L_S (ft)
HS	R		Draft of strut H_S (ft)
TSMAX	R		Maximum thickness of strut t_{max} (ft)
CWP	R		Waterplane area coefficient $C_{WP} = A_W / (L_S t_{max})$ where A_W = waterplane area of one strut
CLCF	R		Waterplane moment coefficient $C_{LCF} = M_{x_S} / (A_W \cdot L_S)$ *
CIYY	R		Waterplane inertia coefficient $C_{IYY} = I_{wp} / (A_W \cdot L_S^2)$ *
XLB	R		Length of body L_B (ft)
HB	R		Draft of body H_B (ft)
AX	R		Maximum cross-sectional area of body A_X (ft ²)
CP	R		Body prismatic coefficient $C_P = \nabla_b / (A_X L_b)$ where ∇_b = displaced volume of one body
CLCB	R		Body moment coefficient $C_{LCB} = M_{x_B} / (\nabla_e \cdot L_e)$ *
SEPDIS	R		Separation distance between centerline of the two bodies (ft)
CSTRUT	R		Distance from centerline of strut to centerline of body (ft)
CSTR2	R		Distance from centerline of second strut to centerline of body (ft)
PDIA	R		Diameter of propeller D (ft)
SPAN	R		Span of a demi-fin (ft)
CHORD	R		Chord of a fin (ft)
TFINS	R		Maximum thickness of a demi-fin (ft)
NLOC	I		Switch indicating the presence of a second strut

*All moments and moments of inertia are taken about the mid-length of the respective strut or body.

COMMON/OFFDES/

PURPOSE: OFFDES stores the variables which define the coefficients of KT/KQ curves.

NAME	TYPE	LENGTH	DEFINITION
PJAY	R	(4)	Coefficients of polynomial representing KT or KQ.

COMMON/OMEGA/

PURPOSE: OMEGA stores the values of variables and constants for evaluation of auxiliary functions and ship resistance coefficients.

NAME	TYPE	LENGTH	DEFINITION
VMFPS	R		Speed of ship V_K (kts)
GAMAOS	R		$\gamma_{os} = \frac{1}{2} g L_s / V^2$ where U is speed in fps
GAMAOB	R		$\gamma_{ob} = \frac{1}{2} g L_b / V^2$
GOSQ	R		$(\gamma_{os})^2$
HSOLS	R		Ratio of draft to length of strut H_s / L_s
HBOLB	R		Ratio of draft to length of body H_b / L_b
WETS	R		Wetted surface area of strut S_s (ft ²)
WETB	R		Wetted surface area of body S_b (ft ²)
WETFIN	R		Wetted surface area of a fin S_f (ft ²)
WTSURF	R		Total wetted surface area
SEP	R		$2b / (\gamma_{os} \cdot L_s)$
PHIS	R		$2 \left(\frac{H_s}{L_s} \right) / \gamma_{os}$
PHIB	R		$2 \left(\frac{H_b}{L_b} \right) / \gamma_{ob}$
RATIOI	R		$\gamma_{ob} / \gamma_{os}$
CFS	R		Frictional drag coefficient of strut C_{F_s}
CFB	R		Frictional drag coefficient of body C_{F_b}
APPDRG	R		Appendage drag (lb)

COMMON/PCOF /

PURPOSE: PCOF stores coefficients for KQ/KT polynomial approximation.

NAME	TYPE	LENGTH	DEFINITION
CT	R	(4)	Coefficients of the polynomial which is an approximation of the KT curve
CQ	R	(4)	Coefficients of the polynomial which is an approximation of the KQ curve

COMMON/PHYSICO/

PURPOSE: PHYSICO stores physical constants.

NAME	TYPE	LENGTH	DEFINITION
RHO	R		Density of water
GNU	R		Kinematic viscosity of water
G	R		Acceleration due to gravity
PI	R		Ratio of circumference to diameter of a circle
DELCF	R		Correction allowance of friction drag from ITTC line.

COMMON/PLOT/

PURPOSE: PLOT stores the constants and characters for the plotting routine.

NAME	TYPE	LENGTH	DEFINITION
NFIRST	I		Position in the arrays of the first ordered pair to be plotted
NLAST	I		Position in the arrays of the last ordered pair to be plotted
NPOINT	I		NPOINT equals "1" if each point from NFIRST to NLAST is to be plotted, "2" if every other point is to be plotted, etc.
XMAX	R		Value of abscissa at right-most grid line
XMIN	R		Value of abscissa at left-most grid line
NSCLI	L		Logical value (Should be false if PLOT1 has not been called and standard grid is desired)
NCHAR	I		Number of valid characters in label
NSCALE	I	(4)	Printing scale factor of ordinate
PCHAR	I	(2)	Plotting characters

COMMON/PSI/

PURPOSE: PSI stores the values of the variables and constants used in integration routines.

NAME	TYPE	LENGTH	DEFINITION
NPTSZ	I		Number of integration steps from γ to $\gamma+1$
PTSAF	R		Scaling factor of step size in integrating from $\alpha_{\max}$ to $\alpha_{\ell\max}$
EXPN	R		Empirical constant for integration to stop
NALMAX	R		Maximum number of integration steps from $\gamma+1$ to $\alpha_{\max}$
NAL	I		Counter of integration steps
TAIL	R		Integration correction made from $\alpha_{\max}$ to $\alpha_{\ell\max}$
ALFA	R		Integrating variable (α)
ALSMAX	R		Maximum of α for integration
NSTEPS	I		Number of integration steps from $\alpha_{\max}$ to $\alpha_{\ell\max}$

COMMON/ROOTC/

PURPOSE: ROOTC stores the coefficients of a third order polynomial
which is the approximation of the KT curve of a propeller.

NAME	TYPE	LENGTH	DEFINITION
A	R	4	Array holding the coefficient of a third order polynomial which is the approximation of the KT curve of a propeller

COMMON/XRPLOTF/

PURPOSE: XRPLOTF stores the values of variables for the plotting routine.

NAME	TYPE	LENGTH	DEFINITION
XL	R		Value of abscissa at left-most grid line
XH	R		Value of abscissa at right-most grid line
YL	R		Value of ordinate at bottom grid line
YH	R		Value of ordinate at top grid line
XMOV	R		Abscissa index increment number for array GRAF
YMOV	R		Ordinate index increment number for array GRAF

COMMON/XRPLLOTG/

PURPOSE: XRPLLOTG stores the values and characters of variables for the plotting routine.

NAME	TYPE	LENGTH	DEFINITION
GRAF	I	(11,204)	Array containing the image to be plotted

COMMON/XRPLQTQ

PURPOSE: XRPLQTQ stores the constants and characters for the plotting routine.

NAME	TYPE	LENGTH	DEFINITION
I	I		Ordinate scale factor is 10^I
J	I		Number digits following ordinate decimal point
K	I		Abcissa scale factor is 10^K
L	I		Number digits following abscissa decimal point
NHL	I		Number of horizontal grid lines
NSBH	I		Number of spaces between adjacent horizontal grid lines
NVL	I		Number of vertical grid lines
NSBV	I		Number of spaces between adjacent vertical grid lines
HCHAR	I		Plotting character of horizontal grid lines
ISX	I		Number of horizontal spaces
ISY	I		Number of vertical spaces
V	L		Logical variable, = . TRUE . when the maximum and minimum values of the ordinate are determined
H	L		Logical variable, = . TRUE . when the maximum and minimum values of the abscissa are determined

SUBROUTINE DOCUMENTATION FOR SYNTHESIS

NAME: PROGRAM SYNTH

PURPOSE: Program SYNTH is the main program for calculating the propulsor resistance and design for SWATH ships.

CALLING SEQUENCE: PROGRAM SYNTH (INPUT, OUTPUT, TAPE 5 = INPUT, TAPE 6 = OUTPUT, TAPE 8)

ARGUMENTS: NONE

COMMON BLOCKS: COEFS, INPUT, OMEGA, PHYSCO, PSI

SUBROUTINES CALLED: RIN, CHEB, RWAVE, FINDRG, ROUT, SHPCMP, Function CFITTC

COMMENTS:

NAME:	SUBROUTINE BESSJ
PURPOSE:	Subroutine BESSJ evaluates the Bessel function from order 0 to order N.
CALLING SEQUENCE:	CALL BESSJ (X, N, VJ)
ARGUMENTS:	X Argument of the Bessel function N Maximum order of the Bessel function VJ Array holding (N+1) values of the Bessel function of order zero up to N, where $VJ(0) = J_0(X)$. . . $VJ(N) = J_N(X)$
COMMON BLOCKS:	NONE
SUBROUTINE CALLED:	NONE
CALLED BY:	RINTEG, RWAVE
COMMENTS:	

NAME: SUBROUTINE BURIL

PURPOSE: SUBROUTINE BURIL uses Burrill's cavitation criteria to determine if the blade area ratio (BAR) used is sufficient and, if it is not, to estimate what BAR is necessary.

CALLING SEQUENCE: CALL BURIL (T, D, H, VA, N, CAVI, BAR, IERROR, PD, NRET)

ARGUMENTS:

T	Thrust of propeller (lbs)
D	Diameter of propeller (ft)
H	Depth of propeller shaft (ft)
VA	Speed of advance (fps)
N	RPM of propeller
CAVI	Percent back cavitation allowed
BAR	Blade area ratio
IERROR	Error message indication as defined in Common Block BARCK
PD	P/D of propeller
NRET	Internal flag indicating BAR optimization

NRET = $\begin{cases} 0 & \text{if not optimized} \\ 1 & \text{if optimized} \end{cases}$

COMMON BLOCK: BARCK

SUBROUTINE CALLED: NONE

CALLED BY: PRODES

COMMENTS:

NAME:	FUNCTION CALC										
PURPOSE:	FUNCTION CALC calculates "KT" or "KQ" for two to seven-bladed propellers, given J, BAR, P/D, Z, and an indicator for "KT" or "KQ."										
CALLING SEQUENCE:	TQ = CALC (J, BAR, PD, Z, ITQ)										
ARGUMENTS:	<table border="0"> <tr> <td>J</td> <td>Advance coefficient of propeller</td> </tr> <tr> <td>BAR</td> <td>Blade area ratio</td> </tr> <tr> <td>PD</td> <td>P/D of propeller</td> </tr> <tr> <td>Z</td> <td>Number of blades</td> </tr> <tr> <td>ITQ</td> <td>Index defined by</td> </tr> </table> $ITQ = \begin{cases} 1 & \text{if thrust is input} \\ 2 & \text{if power is input} \end{cases}$	J	Advance coefficient of propeller	BAR	Blade area ratio	PD	P/D of propeller	Z	Number of blades	ITQ	Index defined by
J	Advance coefficient of propeller										
BAR	Blade area ratio										
PD	P/D of propeller										
Z	Number of blades										
ITQ	Index defined by										
COMMON BLOCK:	OFFDES										
SUBROUTINE CALLED:	NONE										
CALLED BY:	TROOST, ROOT, POLCOF										
COMMENTS:											

NAME: FUNCTION CFITTC

PURPOSE: Function CFITTC determines the ITTC frictional resistance coefficient C_F

CALLING SEQUENCE: CF = CFITTC (RN)

ARGUMENT: RN Reynolds number at R_n at test conditions

COMMON BLOCKS: NONE

SUBROUTINE CALLED: NONE

CALLED BY: SYNTH

COMMENTS:

$$C_F = \frac{0.075}{[\log_{10}(R_n) - 2]^2}$$

NAME:	SUBROUTINE CHEB
PURPOSE:	Subroutine CHEB determines the Chebychev coefficients and wetted surface area of the strut and body. It also plots the strut waterline and the body area distribution.
CALLING SEQUENCE:	CALL CHEB (TITLE)
ARGUMENT:	TITLE Array containing the alphanumeric characters of the title of the experiment
COMMON BLOCKS:	COEFS, INPUT, OMEGA, PHYSCO
SUBROUTINES CALLED:	WSURFB, WSURFS, PCHEB
CALLED BY:	SYNTH
COMMENTS:	

NAME:	SUBROUTINE ERROR
PURPOSE:	Subroutine ERROR prints the error message of the propulsor design.
CALLING SEQUENCE:	CALL ERROR (IE)
ARGUMENT:	IE Error message pointer as defined in Common Block BARCK
COMMON BLOCKS:	NONE
SUBROUTINE CALLED:	NONE
CALLED BY:	PROPP
COMMENTS:	

NAME: FUNCTION EVAL

PURPOSE: Function EVAL evaluates the Chebychev Series:

$$F(X) = \sum_{M=1}^{MMAX} A(M) * U(M,X) + B(M) * V(M,X)$$

CALLING SEQUENCE: EV = EVAL (X,A,B,MMAX)

ARGUMENTS:

X	Arguments of the Chebychev Series
A	Coefficients of the Chebychev Cosine Series
B	Coefficients of the Chebychev Sine Series
MMAX	Maximum order of the Chebychev Series

COMMON BLOCKS: NONE

SUBROUTINE CALLED: NONE

CALLED BY: WSURFS, WSURFB

COMMON BLOCKS: NONE

SUBROUTINE CALLED: NONE

CALLED BY: WSURFS, WSURFB

COMMENTS:

$$U_M(X) = \cos[(2M-1)(\theta)]$$

$$V_M(X) = \sin 2M \theta$$

$$\theta = \sin^{-1}(X)$$

NAME: SUBROUTINE FINDRG

PURPOSE: Subroutine FINDRG determines the fin drag.

CALLING SEQUENCE: CALL FINDRG (RHOS, XNUS, VSFW, SFINS, TFINS, CFINS, TBFINS, SBFINS, DS)

ARGUMENTS:

RHOS	Density of water
XNUS	Kinematic viscosity of water
VSFW	Ship speed (fps)
SFINS	Chord * span
TFINS	Thickness of a fin (ft)
CFINS	Chord of a fin (ft)
TBFINS	$0.01 * TFINS$
SBFINS	$TBFINS * span$
DS	Total fin drag

COMMON BLOCKS: NONE

SUBROUTINES CALLED: FRICT, FOILSC

CALLED BY: SYNTH

COMMENTS:

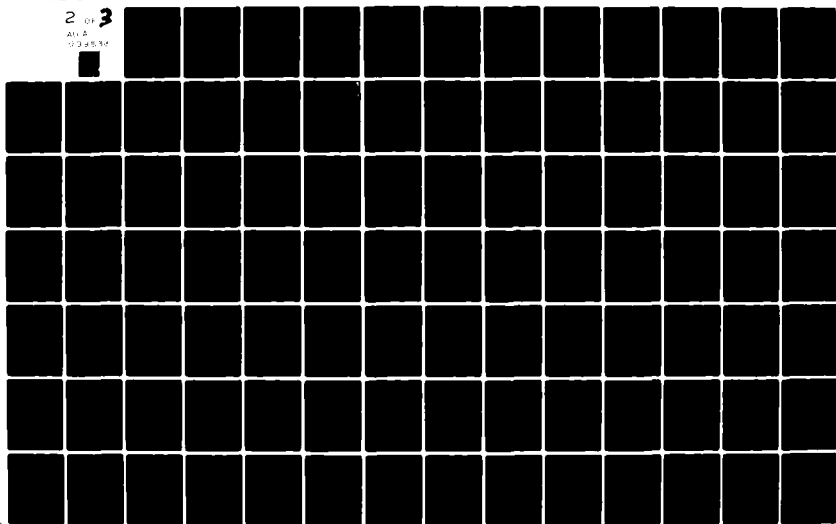
AD-A099 533

DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CE--ETC F/G 20/4
DOCUMENTATION FOR SWATH SHIP RESISTANCE AND PROPULSION PREDICTI--ETC(U)
APR 81 A M REED
DTNSRDC/SPD-0927-02

UNCLASSIFIED

NL

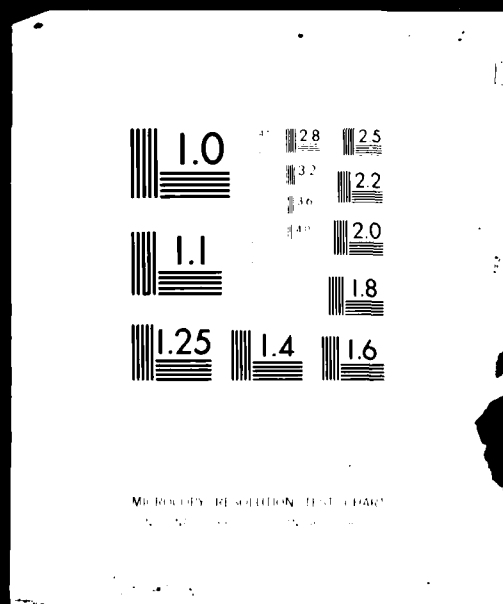
2 OF 3
AD-A
00 05 34



2 OF 3

AD A

099533



NAME: SUBROUTINE FOILSC

PURPOSE: Subroutine FOILSC calculates the drag components for a symmetrical foil section.

CALLING SEQUENCE: CALL FOILSC (XS, CFS, TOC, TOC2, TOC4, SS, TS2, SBS, CLOTB, RFS, RFVAS, RPS, RINTS, RBS)

ARGUMENTS:

XS	Ship constant = $\frac{1}{2} * \rho * SFINS$ VSFW ** 2
CFS	Flat plate friction coefficient
TOC	Thickness/chord ratio
TOC2	T/C**2
TOC4	T/C**4
SS	Chord * span $\approx$ area of foil
TS2	(SS) ** 2 $\approx$ area squared
SBS	TBFINS * span
CLOTB	Chord/0.01 * TFINS
RFS	Frictional resistance of a flat plate
RFUAS	Resistance due to velocity argumentation
RPS	Resistance due to pressure and separation
RINTS	Resistance due to intersection with hull
RBS	Base drag due to bluntness of trailing edge

COMMON BLOCKS: NONE

SUBROUTINE CALLED: NONE

CALLED BY: FINDRG

COMMENTS:

NAME: FUNCTION FORMDR

PURPOSE: Function FORMDR evaluates the form drag coefficient.

CALLING SEQUENCE: FDR = FORMDR (VL)

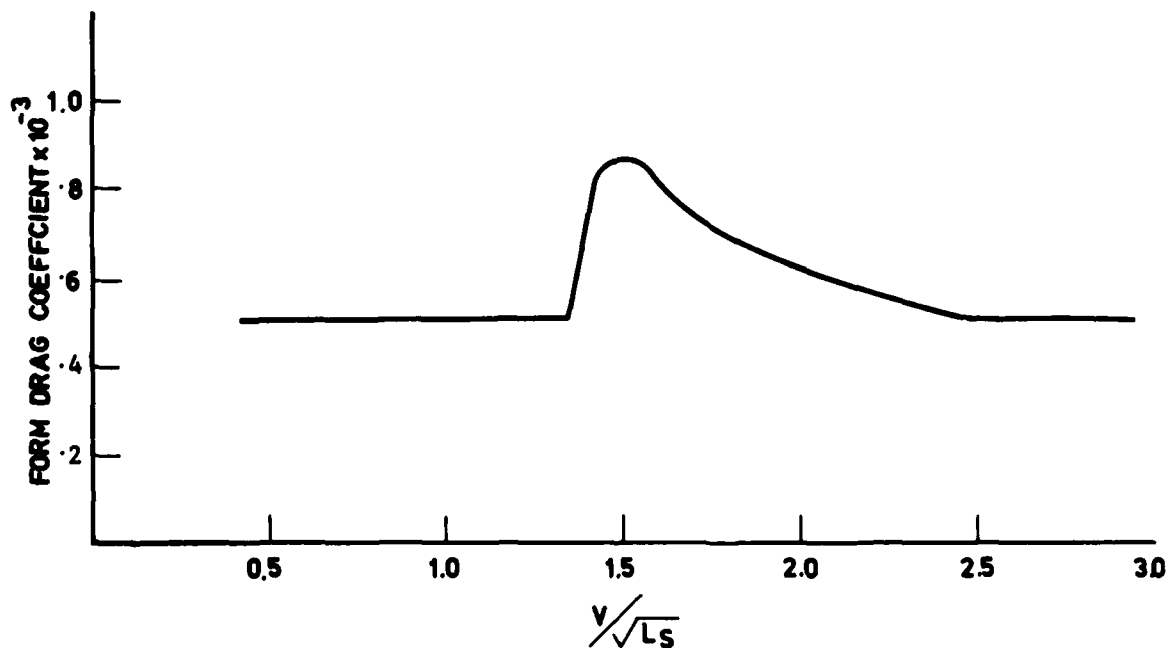
ARGUMENT: VL Ratio of ship speed to square root of length of strut $V/\sqrt{L_S}$

COMMON BLOCKS: NONE

SUBROUTINE CALLED: FUNCTION YINTP

CALLED BY: RWOUT

COMMENTS:



NAME:	SUBROUTINE FRICT	
PURPOSE:	Subroutine FRICT calculates the flat plate friction coefficient based on the local Reynolds number.	
CALLING SEQUENCE:	CALL FRICT (CF, VF, XLEN, XNU)	
ARGUMENTS:	CF	Flat plate friction coefficient
	VF	Speed of ship (fps)
	XLEN	Chord of fin (ft)
	XNU	Kinematic viscosity
COMMON BLOCKS:	NONE	
SUBROUTINE CALLED:	NONE	
CALLED BY:	FINDRG	
COMMENTS:		

NAME: SUBROUTINE PCHEB

PURPOSE: Subroutine PCHEB plots by line printer the body sectional area curve and waterplane outline curve from the given Chebychev coefficients.

CALLING SEQUENCE: CALL PCHEB (AS, BS, AB, BB, NN, TITLE)

ARGUMENTS:

AS	Coefficients of Chebychev Sine Series for strut
BS	Coefficients of Chebychev Cosine Series for strut
AB	Coefficients of Chebychev Sine Series for body
BB	Coefficients of Chebychev Cosine Series for body
NN	Maximum order of Chebychev Series
TITLE	Array containing the alphanumeric characters of the title of the experiment

COMMON BLOCKS: PLOT, XRPLQTQ

SUBROUTINES CALLED: PLOT1, PLOT2, PLOT3, PLOT4

CALLED BY: CHEB

COMMENTS:

NAME: SUBROUTINE PLOT1

PURPOSE: Subroutine PLOT1 sets up spacing and determines the values of the axes.

CALLING SEQUENCE: CALL PLOT 1 (NSCALE, A, B, C, D, E, F)

ARGUMENTS:

NSCALE	Integer array defined as follows:
NSCALE (1)	= I, if printed values of the ordinate are $10^{**} I$ times the actual value
NSCALE (2)	= J, if printed values of the ordinate are $10^{**} J$ times the actual value
NSCALE (3)	= K, if printed values of the abscissa are $10^{**} K$ times the actual values
NSCALE (4)	= L, if printed values of the abscissa are $10^{**} L$ times the actual value
A	Integer number of horizontal grid lines
B	Integer number of spaces beyond each horizontal grid line to the next grid line
C	Integer number of vertical grid lines
D	Integer number of spaces beyond each vertical grid line to the next grid line
E	Horizontal grid character
F	Vertical grid character

COMMON BLOCK: XRPLQTQ

SUBROUTINE CALLED: NONE

CALLED BY: PCHEB

COMMENTS:

NAME: SUBROUTINE PLOT2

PURPOSE: Subroutine PLOT2 examines the minimum and maximum values of the abscissa and the ordinate and establishes an internal formula for computing location in the image region corresponding to the point to be plotted.

CALLING SEQUENCE: CALL PLOT2 (XMAX, XMIN, YMAX, YMIN, NSCLI)

ARGUMENTS:

XMAX	Value of abscissa at right-most grid line
XMIN	Value of abscissa at left-most grid line
YMAX	Value of ordinate at top grid line
YMIN	Value of ordinate at bottom grid line
NSCLI	Logical flag (should be false, if PLOT1 has not been called and standard grid is desired)

COMMON BLOCKS: XRPLTF, XRPLQT, XRPLTG

SUBROUTINE CALLED: NONE

CALLED BY: PCHEB

COMMENTS:

NAME:	SUBROUTINE PLOT3	
PURPOSE:	Subroutine PLOT3 assigns an alpha-character to each point to be plotted.	
CALLING SEQUENCE:	CALL PLOT3 (PCHAR, X, Y, SDATA, FDATA, DDATA)	
ARGUMENTS:	PCHAR	Plotting character
	X	Array containing the X coordinates to be plotted
	Y	Array containing the Y coordinates to be plotted
	SDATA	Integer position in the arrays of the first ordered pair to be plotted
	FDATA	{ 1 if each point from SDATA to DDATA is to be plotted { 2 if every other point is to be plotted { 3 if every third point is to be plotted
	DDATA	Integer position in the array of the last ordered pair to be plotted
COMMON BLOCKS:	XRPLTF, XRPLTG	
SUBROUTINE CALLED:	NONE	
CALLED BY:	PCHEB	
COMMENTS:		

NAME: SUBROUTINE PLOT4

PURPOSE: Subroutine PLOT4 prints the image of the completed graph on the printer, including the values of the abscissa and the ordinate at the grid lines outside the bottom and left edge of the graph.

CALLING SEQUENCE: CALL PLOT4 (MCHAR, NCHAR)

ARGUMENTS: MCHAR Single dimension array containing alpha characters to be plotted at the left of the graph
NCHAR Number of valid characters in MCHAR

COMMON BLOCKS: XRPLTF, XRPLTG, XRPLTQ

SUBROUTINE CALLED: QPLOTZ5

CALLED BY: PCHEB

COMMENTS:

NAME: SUBROUTINE POLCOF

PURPOSE: Subroutine POLCOF determines the coefficients of the polynomials which approximate the KT and KQ curves.

CALLING SEQUENCE: CALL POLCOF (PD, BAR, Z)

ARGUMENTS: PD P/D of propeller
BAR Blade area ratio
Z Number of blades

COMMON BLOCKS: OFFDES, PCOF

SUBROUTINE CALLED: Function CALC

CALLED BY: PROPER

COMMENTS:

NAME: SUBROUTINE PRODES

PURPOSE: Subroutine PRODES is used in the design of a propulsion system to determine characteristics and performance at the design condition.

CALLING SEQUENCE: CALL PRODES (IS)

ARGUMENT: IS Type of propulsor (in this program set equal to 1 to indicate conventional fixed blade propellers)

COMMON BLOCKS: BARCK, CDESC, CPROP

SUBROUTINES CALLED: TROOST, BURIL

CALLED BY: SHPCMP

COMMENTS:

NAME: SUBROUTINE PROPER

PURPOSE: Subroutine PROPER calculates the open water curves' table and off-design performance of propulsor system "IS."

CALLING SEQUENCE: CALL PROPER (IS, VOF, TOF, PWR, TITLE)

ARGUMENTS:

IS	Type of propulsor system (in this program, set to 1 to indicate conventional fixed blade propellers)
VOF	Off design speed (ft)
TOF	Off design thrust (lbs)
PWR	SHP at off design condition
TITLE	Array containing the alphanumeric characters of the title of the experiment

COMMON BLOCKS: CDESC, CPROPR, PCOF, ROOTC

SUBROUTINES CALLED: POLCOF, ROOTP

CALLED BY: SHPCMP

COMMENTS:

NAME: SUBROUTINE PROPP

PURPOSE: Subroutine PROPP prints all data in
common block "CPROP"

CALLING SEQUENCE: CALL PROPP (IS, TITLE)

ARGUMENTS: IS Type of propulsor system (in this
program, set to 1 to indicate
conventional fixed blade propellers)
TITLE Array containing the alphanumeric
characters of the title of the
experiment

COMMON BLOCK: CPROP

SUBROUTINE CALLED: ERROR

CALLED BY: SHPCMP

COMMENTS:

NAME: SUBROUTINE QPLOTZ5

PURPOSE: Subroutine QPLOTZ5 calculates the scaling information needed to generate the format to label the left-hand side of the program.

CALLING SEQUENCE: CALL QPLOTZ5 (PDQ)

ARGUMENT: PDQ Scaling factor for ordinate plot

COMMON BLOCKS: XRPLOTF, XRPLOTQ

SUBROUTINES CALLED: NONE

COMMENTS:

NAME: SUBROUTINE REFLKT

PURPOSE: Subroutine REFLKT reflects the symmetrical
matrices to T and W.

CALLING SEQUENCE: CALL REFLKT

ARGUMENTS: NONE

COMMON BLOCKS: AUX, COEFS

SUBROUTINE CALLED: NONE

CALLED BY: RWAVE

COMMENTS:

NAME:	SUBROUTINE RIN
PURPOSE:	Subroutine RIN reads input data for the program.
CALLING SEQUENCE:	CALL RIN (TITLE)
ARGUMENT:	TITLE Array containing the alphanumeric characters of the title of the experiment
COMMON BLOCK:	INPUT
SUBROUTINE CALLED:	NONE
CALLED BY:	SYNTH
COMMENTS:	

NAME:	SUBROUTINE RINIT
PURPOSE:	Subroutine RINIT initializes the T and W arrays.
CALLING SEQUENCE:	CALL RINIT
ARGUMENTS:	NONE
COMMON BLOCKS:	AUX, COEFS
SUBROUTINE CALLED:	NONE
CALLED BY:	RWAVE
COMMENTS:	

NAME: SUBROUTINE RINTEG

PURPOSE: Subroutine RINTEG provides numerical integration for the auxiliary function of T and W.

CALLING SEQUENCE: CALL RINTEG (ALFA, B, D, NLOC2, WTINT, SEPCOS, SQ)

ARGUMENTS:

ALFA	Integrating variable (α)
B	Ratio of distance between centerlines of strut and body to the length of strut
D	Ratio of distance between centerlines of second strut and body to the length of strut
NLOC2 =	$\begin{cases} 0 & \text{if single strut} \\ 1 & \text{if tandem struts} \end{cases}$
WTINT	Weighting constant for numerical integration
SEPCOS	Value of the Cosine function in the integrand
SQ	Value of $\frac{1}{(\gamma^2 - \gamma'^2)^{1/2}}$ in the integrand

COMMON BLOCKS: AUX, COEFS, INPUT, OMEGA

SUBROUTINE CALLED: BESSJ

CALLED BY: RWAVE

COMMENTS:

NAME: SUBROUTINE ROOT

PURPOSE: Subroutine ROOT uses the Regula Falsi method to find the intersection of the "K" curve and the "K/J**2" curve.

CALLING SEQUENCE: CALL ROOT (PD, Z, BAR, COEFF, IPJ, ITQ, KT, KQ, EFFIN, J2)

ARGUMENTS:

PD	P/D of propeller
Z	Number of blades
BAR	Blade area ratio
COEFF	Optimization coefficients as defined in TROOST
IPJ	Optimization index defined by:
	1, no optimization
	2, optimize N with KT/J^{**2}
IPJ =	3, optimize N with KQ/J^{**3}
	4, optimize D with KT/J^{**4}
	5, optimize D with KQ/J^{**5}
ITQ	Index indicating the following:
	1, if thrust is input
ITQ =	2, if power is input
KT	Thrust coefficient
KQ	Torque coefficient
EFFIN	Open water efficiency
J2	Advance coefficient

COMMON BLOCK: ROOTC

SUBROUTINE CALLED: FUNCTION CALC

CALLED BY: TROOST

COMMENTS:

NAME: SUBROUTINE ROUT

PURPOSE: Subroutine ROUT outputs the computed results for all components of drag coefficients and effective horsepower.

CALLING SEQUENCE: CALL ROUT (TITLE, EHP)

ARGUMENTS: TITLE Array containing the alphanumeric characters of the title of the experiment
EHP Effective horsepower of the ship

COMMON BLOCKS: INPUT, OMEGA, PHYSCO

SUBROUTINES CALLED: SUM, Function FORMDR

CALLED BY: SYNTH

COMMENTS:

NAME: SUBROUTINE RWAVE

PURPOSE: Subroutine RWAVE evaluates the auxiliary functions of T and W.

CALLING SEQUENCE: CALL RWAVE (B, D, NLOC2)

ARGUMENTS:

B Ratio of distance between center-lines of strut and body to the length of strut

D Ratio of distance between center-lines of second strut and body to the length of strut

NLOC2 Switch indicating the presence of a second strut

NLOC2 = $\begin{cases} 0 & \text{if single strut} \\ 1 & \text{if tandem struts} \end{cases}$

COMMON BLOCKS: AUX, COEFS, INPUT, OMEGA, PHYSCO, PSI

SUBROUTINES CALLED: RINIT, SIMPSN, RINTEG, BESSJ, REFLKT

CALLED BY: SWATH

COMMENTS:

Typical Auxiliary Functions

$$\left\{ \begin{array}{l} \frac{T_{smn}}{(2m-1)(2n-1)} \\ \frac{W_{smn}}{(2m)(2n)} \end{array} \right\} = \int_{\gamma_{0s}}^{\infty} \frac{d\alpha}{\alpha^2 \sqrt{\alpha^2 - \gamma_{0s}^2}} D \left(\alpha, \frac{2b}{L_s}, \gamma_{0s} \right) \times E_s^2(\alpha) \left\{ \begin{array}{l} J_{2m-1}(\alpha) J_{2n-1}(\alpha) \\ J_{2m}(\alpha) J_{2n}(\alpha) \end{array} \right\},$$

where

$$D = 1 + \cos \left[\left(\frac{2b}{L_s} \right) \left(\frac{2}{\gamma_{0s}} \right) \alpha \sqrt{\alpha^2 - \gamma_{0s}^2} \right],$$

$$E_s = 1 - e^{-2(n_s/L_s)(\alpha^2/\gamma_{0s})},$$

$$\gamma_{0s} = \frac{gL_s}{(2U^2)},$$

NAME: SUBROUTINE SHPCMP

PURPOSE: Subroutine SHPCMP determines a propeller's design and off-design performance and uses this to determine shaft horsepower.

CALLING SEQUENCE: CALL SHPCMP (VDES, EHPDES, PDIA, BDIA, HB, VOFF, EHP, SHP, NV, TITLE)

ARGUMENTS:

VDES	Ship speed at design point (fps)
EHPDES	Effective horsepower at design point
PDIA	Diameter of propeller (ft)
BDIA	Diameter of hull body (ft)
HB	Draft of body (ft)
VOFF	Off-design speed of ship (fps)
EHP	Effective horsepower at off design speeds
SHP	Shaft horsepower at off design speeds
NV	Number of test speeds
TITLE	Array containing the alphanumeric characters of the title of the experiment

COMMON BLOCKS: CDESC, CPROP

SUBROUTINES CALLED: PRODES, PROPP, PROPER, PROPERO

CALLED BY: SYNTH

COMMENTS:

NAME: SUBROUTINE SIMPSN

PURPOSE: Subroutine SIMPSN sets up Simpson's multiplier table for numerical integration using Simpson's rule.

CALLING SEQUENCE: CALL SIMPSN (NPTS, SIMP)

ARGUMENTS: NPTS Number of integration steps (odd and greater than one)
SIMP Array containing Simpson's coefficients

COMMON BLOCKS: NONE

SUBROUTINE CALLED: NONE

CALLED BY: RWAVE

COMMENTS:

NAME: SUBROUTINE SUM

PURPOSE: Subroutine SUM computes the matrix products as follows:

$$\text{SUM} = \sum_{m=1}^n \sum_{n=1}^m \{A_{sm} A_{sn} T_{smn} + B_{sm} B_{sn} W_{smn}\}$$

CALLING SEQUENCE: CALL SUM (SUM1S, SUM1B, SUM1SB, SUM12S, SUM12B, SUM12SB)

ARGUMENTS:

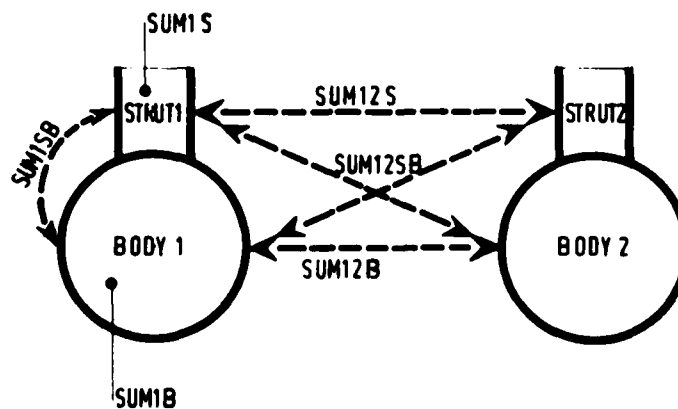
SUM1S	Partial sum for strut 1
SUM1B	Partial sum for body 1
SUM1SB	Partial sum for interaction between strut 1 and body 1
SUM12S	Partial sum for interaction between strut 1 and strut 2
SUM12B	Partial sum for interaction between body 1 and body 2
SUM12SB	Partial sum for interactions between body 1 and strut 2 or body 2 and strut 1

COMMON BLOCKS: AUX, COEFS

SUBROUTINE CALLED: NONE

CALLED BY: ROUT

COMMENTS:



NAME: SUBROUTINE TROOST

PURPOSE: Subroutine TROOST is used with Subroutine ROOT and Function CALC. Either rpm or diameter can be optimized given the other.

CALLING SEQUENCE: CALL TROOST (VAL, P, T, N1, D, Z, PDNEW, BAR, KT, KQ, EFFNEW, J, ITQ, IPJ, RHO, IERROR)

ARGUMENTS:

VAL	Waked ship velocity (kts)
P	Torque per shaft (ft-lb)
T	Propeller thrust (lbs per shaft)
N1	Propeller rpm
D	Propeller diameter (ft)
Z	Number of blades of propeller
PDNEW	P/D of propeller
BAR	Blade area ratio
KT	Thrust coefficient
KQ	Torque coefficient
EFFNEW	Open water efficiency
J	Advance coefficient
ITQ	Index defined by:
	1 if thrust is input
ITQ =	2 if power is input
IPJ	Optimization index defined by:
	1 - no optimization, J known, calculates KT, KQ, EFFNEW only
	2 - optimize N with KT/J^{**2}
IPJ =	3 - optimize N with KQ/J^{**3}
	4 - optimize D with KT/J^{**4}
	5 - optimize D with KQ/J^{**5}
RHO	Water density
IERROR	Error message indicator

COMMON BLOCKS: NONE

SUBROUTINES CALLED: ROOT, TROST2, FUNCTION CALC

CALLED BY: PRODES

COMMENTS:

NAME: SUBROUTINE TROST2

PURPOSE: Subroutine TROST2 is used to calculate a single KT, KQ and efficiency value for a given propeller.

CALLING SEQUENCE: CALL TROST2 (KT, KQ, J, PDTR2, BAR, Z, EFFTR2)

ARGUMENTS:

KT	Thrust coefficient
KQ	Torque coefficient
J	Advance coefficient
PDTR2	Pitch to diameter ratio
BAR	Blade area ratio
Z	Number of blades
EFFTR2	Open water efficiency

COMMON BLOCKS: NONE

SUBROUTINE CALLED: Function CALC

CALLED BY: TROOST

COMMENTS:

NAME: SUBROUTINE WSURFB

PURPOSE: Subroutine WSURFB determines the surface area of a body of revolution given by Chebychev coefficients.

CALLING SEQUENCE: CALL WFSURFB (AREA)

ARGUMENT: AREA Surface area of a body (ft²)

COMMON BLOCKS: COEFS, INPUT, PHYSCO

SUBROUTINE CALLED: FUNCTION EVAL

CALLED BY: CHEB

COMMENTS:

NAME: SUBROUTINE WSURFS

PURPOSE: Subroutine WSURFS determines the wetted surface area of a strut.

CALLING SEQUENCE: CALL WSURFS (AREA)

ARGUMENT: AREA Wetted surface area of a strut (ft²)

COMMON BLOCKS: COEFS, INPUT

SUBROUTINE CALLED: FUNCTION EVAL

CALLED BY: CHEB

COMMENTS:

NAME:	FUNCTION YINTP								
PURPOSE:	Function YINTP interpolates through a set of discrete data.								
CALLING SEQUENCE:	YTP = YINTP (XA, X, Y, N)								
ARGUMENTS:	<table border="0"> <tr> <td>XA</td> <td>Point to be interpolated</td> </tr> <tr> <td>X</td> <td>Array of ordinate data</td> </tr> <tr> <td>Y</td> <td>Array of abscissa data</td> </tr> <tr> <td>N</td> <td>Number of data points</td> </tr> </table>	XA	Point to be interpolated	X	Array of ordinate data	Y	Array of abscissa data	N	Number of data points
XA	Point to be interpolated								
X	Array of ordinate data								
Y	Array of abscissa data								
N	Number of data points								
COMMON BLOCKS:	NONE								
SUBROUTINE CALLED:	NONE								
CALLED BY:	FORMDR								
COMMENTS:									

REFERENCES

- (1) Lin, L.W. and W.G Day, "The Still-Water Resistance and Propulsion Characteristics of Small-Waterplane-Area Twin-Hull (SWATH) Ships." AIAA/SNAME Advanced Marine Vehicles Conference. San Diego, California, 1974.
- (2) Proceedings of the 10th International Towing Tank Conference, London, England, 1963. Published by the National Physical Laboratory, England.
- (3) Lasky, M.P., "An Investigation of Appendage Drag." NSRDC/SPD Report 458-H-01.
- (4) Roddy, R.F. and J. Strom-Tejsen, "Study of Air Cushion Vehicle Propulsion System - Propulsor Design and Performance." NSRDC/SPD Report 378-09.

APPENDICES

PRECEDING PAGE BLANK-NOT FILMED

PROGRAM LISTING OF CLOSEFIT

```

1      PROGRAM SWATH(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE8)
2      SWATH
3      SWATH
4      SWATH
5      C      MAIN PROGRAM FOR RESISTANCE OF SWATH SHIP USING CHEBYCHEV
6      C      REPRESENTATION OF LINES- REVISED FALL 1974 WINTER 1975 - A.M. REED
7      C      2-D STRUTS PLUS SUBMERGED ELONGATED BODIES
8      C      INCLUDES 1-2 INTERFERENCE AND MULTIPLE HULL SPACINGS
9      C      MERGED WITH CHEB AND REVISED FOR TOTAL RESISTANCE--SPRING 1978
10     C
11     COMMON/OUT/HS,HB,XLS,XLB,TSMAX,AX,PI,G,RHO,GNU,WETS,WETB,WTSURF,
12     VMFPS,DELCF,TITLE(8),SPACE
13     C
14     COMMON/AUX/TS(10,10),WS(10,10),TB(10,10),WB(10,10),TSB(10,10),
15     WSB(10,10),TS12(8,10,10),WS12(8,10,10),TB12(8,10,10),
16     WB12(8,10,10),TSB12(8,10,10),WSB12(8,10,10),
17     TSBP(10,10),WSBP(10,10),TSB12P(8,10,10),WSB12P(8,10,10)
18     C
19     COMMON/OMEGA/MMAX,HSOLS,HBOILB,GAMAOS,GAMA0B,NOSEPS,
20     SEPDIS(8),SEP(8),CSTRUT(8),CSTR12(8),GOSO,PHIS,PHIB,
21     RATIOI
22     C
23     COMMON/PSI/NPTSZ,PTSAF,EXPN,NALMAX,NAL,TAIL,ALFA,ALSMAX,NSTEPS
24     C
25     COMMON / COEFS / ASM(10),BSM(10),ABM(10),BBM(10)
26     C
27     DATA D,MMAX / 0.0, 10 /
28     C
29     XLS = LENGTH OF STRUT (FT)
30     HS = DRAFT OF STRUT (FT)
31     TSMAX = MAXIMUM THICKNESS OF STRUT (FT)
32     ASM(1) = STRUT COSINE COEFFICIENTS
33     BSM(1) = STRUT SINE COEFFICIENTS
34     XLB = LENGTH OF BODY (FT)
35     HB = DEPTH OF SUBMERGENCE TO BODY CENTERLINE (FT)
36     AX = MAXIMUM BODY CROSS-SECTIONAL AREA (FT**2)
37     ABM(1) = BODY COSINE COEFFICIENTS
38     BBM(1) = BODY SINE COEFFICIENTS
39     SEPDIS = SEPARATION DISTANCE OF HULL CENTERLINES (FT)
40     NOSEPS = NUMBER OF SEPARATION DISTANCES TO BE TESTED (FT)
41     CSTRUT = DISTANCE OF CL OF STRUT FROM CL OF BODY (FT)
42     CSTR12 = DISTANCE OF CL OF SECOND STRUT( IF PRESENTED) FROM CL OF
43     BODY (FT)
44     (+ FORWARD OF THE BODY CL, - AFT OF THE BODY CL )
45     NLOC = NUMBER OF DIFFERENT STRUT LOCATIONS TO BE TESTED
46     NLOC2 = FLAG INDICATES 2-D STRUTS
47     NLOC2=0, SINGLE STRUT
48     NLOC2 .NE. 0, 2-D STRUTS
49     C      MMAX = MAXIMUM NUMBER OF TERMS USED IN CHEBYCHEV APPROXIMATION
50     C      WETS = WETTED SURFACE AREA OF STRUT
51     C      WETB = WETTED SURFACE AREA OF BODY
52     C      WTSURF = TOTAL WETTED SURFACE AREA OF STRUT AND BODY
53     C      SPACE = DECIMAL FRACTION OF SPACE BETWEEN TANDEM STRUTS TO STRUT
54     C      --FOR TANDEM STRUTS WHICH ARE NOT EQUAL
55     C
56     C
57     C
58     C      INPUT

```

```

60      C      699 CONTINUE
        READ ( 5,210 ) ( TITLE(K),K=1,8)
        READ (5,215) XLS,HS,TSMAX,SPACE
        C
        C CALL CHEB TO INPUT STRUT DATA AND COMPUTE CHEBYCHEV
        C COEFFICIENTS
        C
        C CALL CHEB (ASM,BSM,MMAX)
        C
        C READ(5,215) XLB,HB,AX
        C
        C CALL CHEB TO INPUT BODY DATA AND COMPUTE CHEBYCHEV
        C COEFFICIENTS
        C
        C CALL CHEB (ASM,BBM,MMAX)
        C
        C COMPUTE TOTAL WETTED SURFACE AREA
        C
        C WTSURF = WETS + WETB
        C
        C READ(5,220) NOSEPS
        C READ(5,215) (SEPDIS(I),I=1,NOSEPS)
        C READ(5,220) NLOC,NLOC2
        C READ(5,215) (CSTRUT(I),I=1,NLOC)
        C IF(NLOC2.NE.0) READ(5,215) (CSTRUT2(I),I=1,NLOC)
        C IF(NLOC2.NE.0) WTSURF=WTSURF+WETS-XLS-TSMAX*ASM(1)*PI/4.0
        C
        C ECHO INPUT
        C
        C CALL ECHO (NLOC)
        C
        C PLOT BODY SECTIONAL AREA CURVE AND STRUT WATERPLANE OUTLINE
        C CURVE
        C
        C CALL PCHEB (ASM,BSM,ABM,BBM,MMAX,TITLE)
        C
        C HSOLS = HS/XLS
        C HBOLB = HB/XLB
        C RATIOL = XLB/XLS
        C
        C INPUT TEST VELOCITIES
        C
        C VMFPS=SHIP SPEED IN FEET PER SECOND
        C JTEST = SWITCH SET TO ZERO IF NO PRINTING OF T AND W
        C MATRICES IS NEEDED
        C
        C 710 READ (5,603) VMFPS, JTEST
        C IF (VMFPS.EQ.0.) GO TO 699
        C IF(VMFPS.LT.0.0) STOP
        C GAMADS = G*XLS*.5 / VMFPS**2
        C GAMAOB = GAMADS*XLB/XLS
        C
        C CALCULATE T, W FOR ALL MODEL CONFIGURATION AT ONE

```

```

115      C      SPEED BY CALLING RWAVE
      C
      DO 732 I=1,NLOC
      B=CSTRUT(I)/XLS
      IF(NLOC2.NE.0) D=CSTRT2(I)/XLS
120      CALL RWAVE(B,D,NLOC2)
      IF (JTEST.EQ.0) GO TO 722
      C
      PRINT THE AUXILIARY FUNCTION TABLES , IF JTEST .NE. 0
      C
      CALL PRINT (I,NLOC2)
      C
      C
      C      CALCULATE DRAG COEFFICIENTS AND DRAG FORCES
      C
      CALL RWOUT(I,NLOC2)
722      CONTINUE
732      GO TO 710
      C
      C 210 FORMAT(8A10)
      215 FORMAT(8F10.5)
      220 FORMAT(5I5)
      603 FORMAT(F10.5,2I5)
      END
135
116      SWATH
117      SWATH
118      SWATH
119      SWATH
120      SWATH
121      SWATH
122      SWATH
123      SWATH
124      SWATH
125      SWATH
126      SWATH
127      SWATH
128      SWATH
129      SWATH
130      SWATH
131      SWATH
132      SWATH
133      SWATH
134      SWATH
135      SWATH
136      SWATH
137      SWATH
138      SWATH
139      SWATH

```

13.06.42

04/07/81

FTN 4.6+460

TRACE

74/74

BLOCK DATA BLKDAT.

OPT=0

ROUND=

/

TRACE

```

1      C      BLOCK DATA
COMMON/OUT/HS,HB,XLS,XLB,TSMAX,AX,PI,G,RHO,GNU,WETS,WETB,WTSURF,
1      VMFPS,DELCP,TITLE(8),SPACE
5      C
COMMON / PSI / NPTSZ,PTSFAF,EXP,NALMAX,NAL,TAIL,ALFA,ALMAX,NSTEPS
COMMON/PLOT/NFIRST,NLAST,NPOINT,XMAX,XMIN,NSCL1,NCHAR,NSCALE(4),
1      PCHAR(2)
COMMON/XRPLQTQ/I,J,K,L,NHL,NSBH,NVL,NSBV,HCHAR,VCHAR,ISX,ISY,V,H
10     C      LOGICAL NSCL1
C
C      DEFINE PHYSICAL CONSTANTS
C
15     DATA PI / 3.1415926535897 /
DATA RHO / 1.9367 /
DATA G / 32.155 /
DATA GNU / 1.2970E-5 /
DATA DELCF / 5.0E-4 /
20     C
C      DEFINE CONSTANTS FOR NUMERICAL INTEGRATION
C
25     DATA NPTSZ / 31 /
DATA PTSFAF / 10.0 /
DATA EXPN / 7.0 /
DATA NALMAX / 300 /
DATA ALSMAX / 26.0 /
30     C
C      DEFINE DATA FOR PLOT ROUTINE
C
35     DATA NHL, NSBH, NVL, NSBV, HCHAR, VCHAR, PCHAR(1), PCHAR(2) /
1      6, 10, 11, 10, 1H., 1H., 1HS, 1HB /
C
DATA NFIRST, NLAST, NPOINT / 1, 101, 1 /
C
DATA XMAX, XMIN, NSCL1 / 1.0, -1.0, .TRUE. /
C
DATA NCHAR / 0 /
40     C
DATA NSCALE / 0, 3, 0, 3 /
END

```

04/07/81 13.06.42

FTN 4.6+460

74/74 OPT=0 ROUND=0/ TRACE

SUBROUTINE ECHO

```

1      SUBROUTINE ECHO (NLOC)
      C
      C      ECHO SHIP DATA
      C
5      COMMON/AUX/TS(10,10),WS(10,10),TB(10,10),WB(10,10),TSB(10,10),
      1      WSB(10,10),TS12(8,10,0),WS12(8,10,10),TB12(8,10,10),
      2      WS12(8,10,10),TSB12(8,10,10),WSB12(8,10,10)
      3      ,TSBP(10,10),WSP(10,10),TSB12P(8,10,10),WSB12P(8,10,10)
      C
10     COMMON/OMEGA,MNAX,HSOLS,H8OLB,GAMAOS,GAMA0B,NOSEPS,
      1      SEPDIS(8),SEP(8),CSTRUT(8),CSTRT2(8),GOSQ,PHIS,PHIB,
      2      RATIOI
      C
15     COMMON/OUT/HS,HB,XLS,XLB,TSMAX,AX,PI,G,RHO,GNU,WETS,WETB,WTSURF,
      1      VMFPS,DELCF,TITLE(8),SPACE
      C
      COMMON/COEFS/ASM(10),BSM(10),ABM(10),BBM(10)
      WRITE (6,310) (TITLE(K),K=1,8)
      WRITE (6,315) MNAX
      WRITE (6,325) XLS,HS,TSMAX
      WRITE (6,330) (ASM(K),K=1,MNAX)
      WRITE (6,335) (BSM(K),K=1,MNAX)
      WRITE (6,340) (ABM(K),K=1,MNAX)
      WRITE (6,345) (BBM(K),K=1,MNAX)
      WRITE (6,350) (WTSURF)
      WRITE (6,355) (SEPDIS(I),I=1,NOSEPS)
      WRITE (6,356) (CSTRUT(I),I=1,NLOC)
      RETURN
310  FORMAT(1H1,///,3X,* SUMMARY OF DATA FOR *.8A10//)
315  FORMAT(1H,5X,45H NUMBER OF TERMS IN HULL COEFFICIENT SERIES =,I6)
325  FORMAT(1H,3X,35H STRUT GEOMETRIC CHARACTERISTICS ,///,5X,
      1      BH XLS =,F12.6,9X,BH HS =,F12.6,9X,8H TSMAX =,F12.6)
330  FORMAT(1H0,33X,*STRUT CHEBYCHEV COEFFICIENTS ASM(M)*/(10X,5E16.8))
335  FORMAT(1H0,33X,*STRUT CHEBYCHEV COEFFICIENTS BSM(M)*/(10X,5E16.8))
340  FORMAT(1H0,3X,35H BODY GEOMETRIC CHARACTERISTICS ,///,5X,
      1      BH XLB =,F12.6,9X,BH HB =,F12.6,9X,8H AX =,F12.6)
345  FORMAT(1H0,33X,*BODY CHEBYCHEV COEFFICIENTS ABM(M)*/(10X,5E16.8))
350  FORMAT(1H0,33X,*BODY CHEBYCHEV COEFFICIENTS BBM(M)*/(10X,5E16.8))
355  FORMAT(1H0,5X,* SEPARATION DISTANCE IN FEET =*,8F8.3)
356  FORMAT(1H0,3X,35H STRUT LOCATION IN FEET =*,8(2X,F10.4))
357  FORMAT(1H0,4X,* WEITED SURFACE IN SQUARE FEET =*,F12.3)
      END

```

```

1      SUBROUTINE PRINT (I,NLOC2)
      C
      C
      C      PRINT THE AUXILIARY FUNCTION TABLES OF T AND W
      COMMON/OUT/HS,HB,XLS,XLB,TSMAX,AX,PI,G,RHO,GNU,WETS,WETB,WTSURF,
      VMFPS,DELCF,TITLE(8),SPACE
      1
      C
      COMMON/AUX/TS(10,10),WS(10,10),TB(10,10),WB(10,10),TSB(10,10),
      WSB(10,10),TS12(8,10,10),WS12(8,10,10),TB12(8,10,10),
      10      WB12(8,10,10),TSB12(8,10,10),WSB12(8,10,10),
      2      ,TSBP(10,10),WSBP(10,10),TSB12P(8,10,10),WSB12P(8,10,10)
      3
      C
      COMMON/OMEGA/MMAX,HSOLS,HBOLB,GAMAOS,GAMAOSB,NOSEPS,
      1      SEPDIS(8),SEP(8),CSTRUT(8),CSTR2(8),GOSQ,PHIS,PHIB,
      2      RATIOL
      C
      C
      C
      15      C
      C
      C
      20      WRITE (6,906) CSTRUT(1)
      WRITE (6,904) CSTRUT(1)
      IF (NLOC2.NE.0) WRITE (6,904) CSTR2(1)
      WRITE (6,903) MMAX,NOSEPS
      WRITE (6,904) GAMAOS,RATIOL,HSOLS,HBOLB,TSMAX,AX,XLS
      25      WRITE (6,904) (SEP(J),J=1,NOSEPS)
      WRITE (6,905)
      WRITE (6,904) ((TS(M,N),M=1,MMAX),N=1,MMAX)
      WRITE (6,905)
      WRITE (6,904) ((WS(M,N),M=1,MMAX),N=1,MMAX)
      WRITE (6,905)
      WRITE (6,904) ((TB(M,N),M=1,MMAX),N=1,MMAX)
      WRITE (6,905)
      WRITE (6,904) ((WB(M,N),M=1,MMAX),N=1,MMAX)
      WRITE (6,905)
      WRITE (6,904) ((TSB(M,N),M=1,MMAX),N=1,MMAX)
      WRITE (6,905)
      WRITE (6,904) ((TSBP(M,N),M=1,MMAX),N=1,MMAX)
      WRITE (6,905)
      WRITE (6,904) ((WSB(M,N),M=1,MMAX),N=1,MMAX)
      WRITE (6,905)
      WRITE (6,904) ((WSBP(M,N),M=1,MMAX),N=1,MMAX)
      30      DO 712 L=1,NOSEPS
      WRITE (6,906)
      WRITE (6,904) ((TS12(L,M,N),M=1,MMAX),N=1,MMAX)
      WRITE (6,905)
      WRITE (6,904) ((WS12(L,M,N),M=1,MMAX),N=1,MMAX)
      WRITE (6,905)
      WRITE (6,904) ((TB12(L,M,N),M=1,MMAX),N=1,MMAX)
      WRITE (6,905)
      WRITE (6,904) ((WB12(L,M,N),M=1,MMAX),N=1,MMAX)
      WRITE (6,905)
      WRITE (6,904) ((TSB12(L,M,N),M=1,MMAX),N=1,MMAX)
      WRITE (6,905)
      WRITE (6,904) ((TSB12P(L,M,N),M=1,MMAX),N=1,MMAX)
      WRITE (6,905)
      WRITE (6,904) ((WSB12(L,M,N),M=1,MMAX),N=1,MMAX)
      WRITE (6,905)
      WRITE (6,904) ((WSB12P(L,M,N),M=1,MMAX),N=1,MMAX)
      35      40      45      50      55

```

PRINT 59
PRINT 60
PRINT 61
PRINT 62
PRINT 63
PRINT 64
PRINT 65

712 CONTINUE
RETURN
903 FORMAT(1X,8I10)
904 FORMAT(1X,10E13.5)
905 FORMAT(1X)
906 FORMAT (1H1,30X,* AUXILIARY FUNCTION TABLE OF T AND W *)
END

60


```

60      C
      CWB12 = R12B/AWTSF
      CMSB12 = AASB*SM12SB
      R12SB = CMSR12-RHO-G*TSMAX*HS*XL5
      CWSB12 = R12SB/AWTSF
      C2 = CWS12+CWB12+CWSB12
      C3 = C1+C2
      C
      FROUDE NUMBER
      FROUDS = VMFPS/SQRT(32.155*XL5)
      VLS = FROUDS*SQRT(G)/1.6878
      FROUDB = VMFPS/SQRT(32.155*XL5)
      VLB = FROUDB*SQRT(G)/1.6878
      VMKNITS = VMFPS/1.6878
      C
      COMPUTE THE FORM DRAG
      CFORM = FORMDR (VLS)
      CR = C3 + CFORM
      RR = CR * AWTSF
      C
      COMPUTE FRICTIONAL DRAG
      CFS = CFITTC (RNS)
      CFB = CFITTC (RNB)
      RFS = 0.5*RHO*VMFPS*VMFPS*WETS*(CFS+DELCF)
      RFB = 0.5*RHO*VMFPS*VMFPS*WETB*(CFS+DELCF)
      RF = RFS + RFB
      CFSP = RFS / AWTSF
      CFBP = RFB / AWTSF
      CF = CFSP + CFBP
      C
      COMPUTE TOTAL DRAG
      CT = CR + CF
      RT = RR + RF
      C
      COMPUTE EHP
      EHP = RT * VMFPS / 550.
      C
      OUTPUT
      WRITE(6,410) (TITLE(K),K=1,8)
      WRITE(6,325) XLS,HS,TSMAX
      WRITE(6,415) GAMAOB,FROUDS,MSOLS
      WRITE(6,340) XLB,HB,AX
      WRITE(6,420) GAMAOB,FROUDB,HBOB
      WRITE(6,356) SEPDIS(L),B2OLB
      IF(NLOC2.NE.0) GO TO 15
      WRITE(6,358) CSTRUT(1),CSOLB
      GO TO 20
      15 WRITE(6,359) CSTRUT(1),CSOLB,CSTRUT2(1),CSOLB2
      20 WRITE(6,357) WTSURF
      WRITE (6,360) VMFPS,VMKNITS,VLS,VLB
      WRITE(6,430) CWS1,RIS,CWS1

```

13.06.42

04/07/81

FTN 4.6+460

74/74 OPT=0 ROUND=0/ TRACE

SUBROUTINE RWOUT

```

115 WRITE(6,440) CNB1,R18,CWB1
    WRITE(6,445) CNSB1,R15B,CWSB1,C1
    WRITE(6,450) CNS12,R12S,CWS12
    WRITE(6,455) CNB12,R12B,CWB12
    WRITE(6,460) CNSB12,R12SB,CWSB12,C2,C3
120 WRITE(6,510) CR,RR
    WRITE(6,520) CF,RF
    WRITE(6,530) CT,RT
    WRITE(6,540) EHP
125 CONTINUE
    RETURN
325 FORMAT(1H,3X,35H STRUT GEOMETRIC CHARACTERISTICS,/,5X,
1 8H XLS =,F12.6,9X,8H HS =,F12.6,9X,8H TSMAX =,F12.6)
340 FORMAT(1H0,3X,35H BODY GEOMETRIC CHARACTERISTICS,/,5X,
1 8H XLB =,F12.6,9X,8H HB =,F12.6,9X,8H AX =,F12.6)
356 FORMAT(0,5X,* SEPARATION DISTANCE IN FEET =,F8.3,
1 5X,* B/LB =,F7.3)
357 FORMAT(1H0,4X,* WETTED SURFACE OF A DEMI-HULL IN SQUARE FEET =,
1 F12.3)
358 FORMAT(0,5X,* STRUT CL FROM BODY CL IN FEET =,F8.3,
1 5X,* STRUT OFFSET/LB =,F7.3)
359 FORMAT(0,5X,* FORWARD STRUT CL FROM BODY CL IN FEET =,F8.3,
1 5X,* FORWARD STRUT OFFSET/LB =,F7.3//
2 * AFT STRUT CL FROM BODY CL IN FEET =,F8.3,
3 5X,* AFT STRUT OFFSET/LB =,F7.3)
140 FORMAT(1H0,5X,* SHIP SPEED IS *,F8.3,* FPS,*,F8.3,* KNOTS,*,
1 6X,* STRUT V-L RATIO =,F8.3,6X,* BODY V-L RATIO =,F8.3)
410 FORMAT(*1,*,35H SHIP RESISTANCE CALCULATIONS FOR,8A10)
415 FORMAT(1H,4X,
1 F12.6,9X,8H HS/LS =,F12.6)
420 FORMAT(1H,4X,
1 F12.6,9X,8H HB/LB =,F12.6)
430 FORMAT(1H0,30H WAVE RESISTANCE IN POUNDS
1 30H RSW/(RHO*G*TSMAX*HS*LS) =,E16.8,
2 /,30H FOR A DEMI-HULL
3 32H STRUT WAVE DRAG IN POUNDS =,F10.3,/30X,
2 32H R15/(RHO/2*WTSURF*V**2) =,F10.3,4HE-03)
440 FORMAT(1H0,29X,30H RBW/(RHO*G*AX*LB)
1 32H BODY WAVE DRAG IN POUNDS =,E16.8,/30X,
2 32H R13/(RHO,2*WTSURF*V**2) =,F10.3,/30X,
2 32H R18/(RHO,2*WTSURF*V**2) =,F10.3,4HE-03)
445 FORMAT(1H0,29X,30H RSBW/(RHO*G*TSMAX*HS*LS) =,E16.8,/30X,
1 32H STRUT-BODY WAVE DRAG IN LBS =,F10.3,/30X,
2 32H R15B/(RHO/2*WTSURF*V**2) =,F10.3,4HE-03,
3 10X,10H CW1 TOT =,F10.3,4HE-03)
450 FORMAT(1H0,29X,30H R12SB/(RHO*G*TSMAX*HS*LS) =,E16.8,/30X,
1 32H STRUT INTERFERENCE DRAG LBS =,F10.3,/30X,
2 32H R12S/(RHO/2*WTSURF*V**2) =,F10.3,4HE-03)
455 FORMAT(1H0,29X,30H R12BW/(RHO*G*AX*LB)
1 32H BODY INTERFERENCE DRAG LBS =,E16.8,/30X,
2 32H R12B/(RHO,2*WTSURF*V**2) =,F10.3,/30X,
2 32H R12B/(RHO,2*WTSURF*V**2) =,F10.3,4HE-03)
460 FORMAT(1H0,29X,30H R12SBW/(RHO*G*TSMAX*HS*LS) =,E16.8,/30X,
1 32H STRUT-BODY INTERFERENCE DRAG LB =,F10.3,/30X,
2 32H R12SB/(RHO,2*WTSURF*V**2) =,F10.3,4HE-03,
3 10X,10H CW2 TOT =,F10.3,4HE-03,/,66X,10H CW TOT =,F10.3,
4 4HE-03)
510 FORMAT(1H0,29H RESIDUAL RESISTANCE
1 30H RESIDUAL DRAG COEFFICIENT =,F10.3,4HE-03,
170 RWOUT
171 RWOUT
172 RWOUT

```

```

      2 /30X,30H RESIDUAL RESISTANCE = ,F10.3,4H LBS)
      520 FORMAT(1H0.29X,30H FRICTIONAL DRAG COEFF.
      1 /30X,30H FRICTIONAL RESISTANCE =,F10.3,4H LBS)
      530 FORMAT(//13X,*TOTAL DRAG COEFFICIENT IS *,F10.3,4H LBS)
      1 //15X,*TOTAL RESISTANCE FOR A DEMI-HULL *,F10.3,4H LBS)
      540 FORMAT (//20X,* EHP FOR A DEMI-HULL *,F10.3)
      END
  
```

```

      RWOUT
      RWOUT
      RWOUT
      RWOUT
      RWOUT
      RWOUT
      RWOUT
  
```

```

      173
      174
      175
      176
      177
      178
      179
  
```



```

1      SUBROUTINE RWAVE(B,D,NLOC2)
C
C      SUBROUTINE FOR COMPUTING AUXILIARY FUNCTIONS T AND W
C      USING CHEBYSHEV EXPANSION - PROGRAM VERSION OF 8/05/72
C      MODIFICATIONS BY A M REED OCTOBER 1974
C
5      DIMENSION VJALFA(21), SIMP(100), SEPCOS(8)
C
10     COMMON/OUT/HS,HB,XLS,XLB,TSMAX,AX,PI,G,RHO,GNU,WETS,WETB,WTSURF,
C      VMFPS,DELCP,TITLE(8),SPACE
C
15     COMMON/AUX/TS(10,10),WS(10,10),TB(10,10),WB(10,10),TSB(10,10),
C      WSB(10,10),TS12(8,10,10),WS12(8,10,10),TB12(8,10,10),
C      WB12(8,10,10),TSB12(8,10,10),WSB12(8,10,10),
C      ,TSBP(10,10),WSBP(10,10),TSB12P(8,10,10),WSB12P(8,10,10)
C
20     COMMON/OMEGA/MNAX,HSOLS,HBOLB,GAMAOS,GMAOB,NOSEPS,
C      SEPDIS(8),SEP(8),CSTRUT(8),CSTRT2(8),GOSQ,PHIS,PHIB,
C      RATIOL
C
25     COMMON/PSI/NPTSZ,PTSAF,EXP,NALMAX,NAL,TAIL,ALFA,ALSMAX,NSTEPS
C
C      DATA PI / 3.1415926535897 /
C
C      NOSEPS IS THE NUMBER OF HULL SEPARATIONS
C      MNAX IS MAXIMUM NUMBER OF TERMS IN CHEBYSHEV SERIES
C
30     MNAX2 = 2*MNAX
C      GOSQ=GAMAOS**2
C      PHIS = 2.0*HSOLS/GAMAOS
C      PHIB = 2.0*HBOLB/GMAOB
C      RATIOL = GMAOB/GAMAOS
C
C      SET THE T AND W ARRAYS TO ZERO
C
35     CALL RINIT
C      DO 5 L=1,NOSEPS
C      SEP(L) = (2.0/GAMAOS)*SEPDIS(L)/XLS
C
40     INTEGRATION OF T AND W FROM ALPHA EQUALS GAMAOS TO GAMAOS+1.0
C      MAKE TRIGONOMETRIC SUBSTITUTION TO AVOID SQUARE ROOT
C      SINGULARITY AT ALPHA EQUALS GAMAOS
C
45     ZETA IS THE VARIABLE OF INTEGRATION--ZETA=SQRT(ALPHA-GAMAOS)
C      NPTSZ IS THE NUMBER OF POINTS TO TAKE IN THE INTERVAL
C      0.0 LESS THAN OR EQUAL ZETA LESS THAN OR EQUAL 1.0
C      AINCR IS THE STEP SIZE FOR THE ZETA INTEGRATION
C      ALFAZ IS THE VALUE OF ALPHA FOR ZETA VARYING FROM ZERO TO ONE
C
50     CALL SIMPSON(NPTSZ,SIMP)
C      XNPTZ=FLOAT(NPTSZ)
C      AINCR = 1.0/(XNPTZ-1.0)
C      HZETA = AINCR/3.0
C      DO 30 I=1,NPTSZ
C      SIMP(I) = SIMP(I)+HZETA
C      EI=FLOAT(I-1)

```


13.06.42

04/07/81

FTN 4.6+460

TRACE

74/74

SUBROUTINE RWAVE

```

115 IF(NLOC2.NE.0) CON=2.0*(1.0+COS((B-D)*ALFA*2.0))
    CALL BESSJ(ALFA,MNAX2,VJALFA)
    VS = PHIS*ALFASQ
    SQ=1.0/SQRT(ALFASQ-GOSQ)
    ESA = 1.0
    IF ( VS .LT. 300.0 ) ESA = ESA- EXP(-VS)
    ES = SQ * ESA**2 / ALFASQ
    DO 276 M = 1,MNAX
        MAA = 2*M
        MBB = MAA + 1
    DO 276 N = 1,MNAX
        NAA = 2*N
        NBB = NAA + 1
        WTINT = ANWT * WT
        TSA = WTINT*ES*VJALFA(MAA)*VJALFA(NAA)*CON
        WSA = WTINT*ES*VJALFA(MBB)*VJALFA(NBB)*CON
        TS(M,N) = TS(M,N) + TSA
        WS(M,N) = WS(M,N) + WSA
        ALFA = ALFA + SSIZE
        BNWT = 2.
    IF (ANWT.LE.2.) BNWT = 4.
    IF (100.EQ.NDO-1) BNWT = 1.
    ANWT = BNWT
276 ALFASQ=ALFA**2
277 RAT = GAMAOS/ALFA
278 IF (RAT.LT.1.E-4) GO TO 280
    TAIL=(-SQRT(ALFASQ-GOSQ)/ALFASQ+ASIN(RAT)/GAMAOS)/(2.0*PI*GOSQ)
    GO TO 290
    TAIL=1.0/(3.0*PI*ALFA*ALFASQ)
280 C
    C
    C
290 C
        CALCULATE TAIL RESULTS
        DO 296 M = 1,MNAX
        DO 296 N = 1,MNAX
        SIGN = (-1)**(N+N)
        TS(M,N) = TS(M,N) - SIGN*TAIL
        WS(M,N) = WS(M,N) + SIGN*TAIL
        CONTINUE
        MULTIPLY BY CONSTANTS AND REFLECT SYMMETRIC MATRICES)
        CALL REFLKT
        RETURN
        FORMAT (=1ALFA INTEGRATION REACHED NALMAX*)
302 END

```

```

1      SUBROUTINE RINIT
2
3      C
4      C
5      C
6      C
7      C
8      C
9      C
10     C
11     C
12     C
13     C
14     C
15     C
16     C
17     C
18     C
19     C
20     C
21     C
22     C
23     C
24     C
25     C
26     C
27     C
28     C
29     C
30     C
31     C
32     C
33     C
34     C
35     C
36     C
37     C
38     C
39     C
40     C
41     C
42     C
43     C
44     C
45     C
46     C

```

INITIALIZE THE T AND W ARRAYS TO ZERO

COMMON/AUX/TS(10,10),WS(10,10),TB(10,10),WB(10,10),TSB(10,10),
 1 WSB(10,10),TS12(8,10,10),WS12(8,10,10),TB12(8,10,10),
 2 WB12(8,10,10),TSB12(8,10,10),WSB12(8,10,10),
 3 TSBP(10,10),WSBP(10,10),TSB12P(8,10,10),WSB12P(8,10,10)

COMMON/OMEGA/MNAX,HSOLS,HBOLB,GAMAOS,GAMADB,NOSEPS,
 1 SEPDIS(8),SEP(8),CSTRUT(8),CSTR2(8),GOSO,PHIS,PHIB,
 2 RATIOI

DO 4 M=1,MNAX
 DO 4 N=M,MNAX
 TS(M,N) = 0.0
 WS(M,N) = 0.0
 TB(M,N) = 0.0
 WB(M,N) = 0.0
 TSB(M,N) = 0.0
 TSBP(M,N) = 0.0
 TSB(N,M) = 0.0
 TSBP(N,M) = 0.0
 WSB(M,N) = 0.0
 WSBP(M,N) = 0.0
 WSB(N,M) = 0.0
 WSBP(N,M) = 0.0

DO 4 L=1,NOSEPS
 TS12(L,M,N) = 0.0
 WS12(L,M,N) = 0.0
 TB12(L,M,N) = 0.0
 WB12(L,M,N) = 0.0
 TSB12(L,M,N) = 0.0
 TSB12P(L,M,N) = 0.0
 TSB12(L,N,M) = 0.0
 TSB12P(L,N,M) = 0.0
 WSB12(L,M,N) = 0.0
 WSB12P(L,M,N) = 0.0
 WSB12(L,N,M) = 0.0
 WSB12P(L,N,M) = 0.0

CONTINUE
 RETURN
 END

13.06.42

04/07/81

FTN 4.6+460

TRACE

74/74

SUBROUTINE RINTEG

OPT=0

ROUND=

=

TRACE

```

1      SUBROUTINE RINTEG(ALFA,B,D,NLOC2,WTINT,SEPCOS,SQ)
2      RINTG
3      C
4      RINTG
5      C
6      C
7      C
8      NUMERICAL INTEGRATION FOR AUXILIARY FUNCTIONS T AND W
9      RINTG
10     COMMON/AUX/TS(10,10),WS(10,10),TB(10,10),WB(10,10),TSB(10,10),
11     WSB(10,10),TS12(8,10,10),WS12(8,10,10),TB12(8,10,10),
12     WB12(8,10,10),TSB12(8,10,10),WSB12(8,10,10),
13     TSBP(10,10),WSBP(10,10),TSB12P(8,10,10),WSB12P(8,10,10)
14     RINTG
15     C
16     COMMON/OMEGA/MAX,HSOLS,HBOLB,GAMAOS,GAMA08,NOSEPS,
17     SEPDIS(B),SEP(B),CSTRUT(8),CSTRT2(8),GOSQ,PHIS,PHIB,
18     RATIOL
19     RINTG
20     C
21     DIMENSION VJALFA(21), VJBETA(21), SEPCOS(8)
22     RINTG
23     C
24     MMAX2=2*MMAX
25     BEJA = RATIOL*ALFA
26     ALFASQ=ALFA**2
27     RINTG
28     C
29     CALCULATE FACTORS FOR STRUT NOT CENTERED
30     C
31     CO=COS(B*ALFA*2.0)
32     S1=SIN(B*ALFA*2.0)
33     CON=1.0
34     IF (NLOC2.EQ.0) GO TO 32
35     C
36     CALCULATE FACTORS FOR TANDEM STRUTS
37     C
38     CO=(CO+COS(D*ALFA*2.0))
39     S1=(S1+SIN(D*ALFA*2.0))
40     CON=2.0*(1.0+COS((B-D)*ALFA*2.0))
41     CONTINUE
42     C
43     COMPUTE AUXILIARY FUNCTIONS VJALFA AND VJBETA
44     C
45     CALL BESSJ(ALFA,MMAX2,VJALFA)
46     CALL BESSJ(BETA,MMAX2,VJBETA)
47     C
48     COMPUTE AUXILIARY FUNCTIONS ES AND EB
49     C
50     VS=PHIS*ALFASQ
51     VB = PHIB*BETA**2
52     ESA = 1.0
53     EBA = 0.0
54     IF(VS .LT. 300.) ESA = ESA-EXP(-VS)
55     IF(VB .LT. 300.) EBA = EXP(-VB)
56     ES=SQ*ESA**2/ALFASQ
57     EB=SQ*EBA**2/ALFASQ
58     ESB = SQ*ESA*EBA
59     C
60     TB,TS,WB,WS,TW12,TS12,WB12, AND WS12 ARE ALL SYMMETRIC WRT M
61     AND N SO WE NEED ONLY CALCULATE THE UPPER HALF OF THESE
62     MATRICES
63     C
64     DO 55 M=1,MMAX
65     RINTG
66     RINTG
67     RINTG
68     RINTG
69     RINTG
70     RINTG
71     RINTG
72     RINTG
73     RINTG
74     RINTG
75     RINTG
76     RINTG
77     RINTG
78     RINTG
79     RINTG
80     RINTG
81     RINTG
82     RINTG
83     RINTG
84     RINTG
85     RINTG
86     RINTG
87     RINTG
88     RINTG
89     RINTG
90     RINTG
91     RINTG
92     RINTG
93     RINTG
94     RINTG
95     RINTG
96     RINTG
97     RINTG
98     RINTG
99     RINTG
100    RINTG
101    RINTG
102    RINTG
103    RINTG
104    RINTG
105    RINTG
106    RINTG
107    RINTG
108    RINTG
109    RINTG
110    RINTG
111    RINTG
112    RINTG
113    RINTG
114    RINTG
115    RINTG
116    RINTG
117    RINTG
118    RINTG
119    RINTG
120    RINTG
121    RINTG
122    RINTG
123    RINTG
124    RINTG
125    RINTG
126    RINTG
127    RINTG
128    RINTG
129    RINTG
130    RINTG
131    RINTG
132    RINTG
133    RINTG
134    RINTG
135    RINTG
136    RINTG
137    RINTG
138    RINTG
139    RINTG
140    RINTG
141    RINTG
142    RINTG
143    RINTG
144    RINTG
145    RINTG
146    RINTG
147    RINTG
148    RINTG
149    RINTG
150    RINTG
151    RINTG
152    RINTG
153    RINTG
154    RINTG
155    RINTG
156    RINTG
157    RINTG
158    RINTG
159    RINTG
160    RINTG
161    RINTG
162    RINTG
163    RINTG
164    RINTG
165    RINTG
166    RINTG
167    RINTG
168    RINTG
169    RINTG
170    RINTG
171    RINTG
172    RINTG
173    RINTG
174    RINTG
175    RINTG
176    RINTG
177    RINTG
178    RINTG
179    RINTG
180    RINTG
181    RINTG
182    RINTG
183    RINTG
184    RINTG
185    RINTG
186    RINTG
187    RINTG
188    RINTG
189    RINTG
190    RINTG
191    RINTG
192    RINTG
193    RINTG
194    RINTG
195    RINTG
196    RINTG
197    RINTG
198    RINTG
199    RINTG
200    RINTG
201    RINTG
202    RINTG
203    RINTG
204    RINTG
205    RINTG
206    RINTG
207    RINTG
208    RINTG
209    RINTG
210    RINTG
211    RINTG
212    RINTG
213    RINTG
214    RINTG
215    RINTG
216    RINTG
217    RINTG
218    RINTG
219    RINTG
220    RINTG
221    RINTG
222    RINTG
223    RINTG
224    RINTG
225    RINTG
226    RINTG
227    RINTG
228    RINTG
229    RINTG
230    RINTG
231    RINTG
232    RINTG
233    RINTG
234    RINTG
235    RINTG
236    RINTG
237    RINTG
238    RINTG
239    RINTG
240    RINTG
241    RINTG
242    RINTG
243    RINTG
244    RINTG
245    RINTG
246    RINTG
247    RINTG
248    RINTG
249    RINTG
250    RINTG
251    RINTG
252    RINTG
253    RINTG
254    RINTG
255    RINTG
256    RINTG
257    RINTG
258    RINTG
259    RINTG
260    RINTG
261    RINTG
262    RINTG
263    RINTG
264    RINTG
265    RINTG
266    RINTG
267    RINTG
268    RINTG
269    RINTG
270    RINTG
271    RINTG
272    RINTG
273    RINTG
274    RINTG
275    RINTG
276    RINTG
277    RINTG
278    RINTG
279    RINTG
280    RINTG
281    RINTG
282    RINTG
283    RINTG
284    RINTG
285    RINTG
286    RINTG
287    RINTG
288    RINTG
289    RINTG
290    RINTG
291    RINTG
292    RINTG
293    RINTG
294    RINTG
295    RINTG
296    RINTG
297    RINTG
298    RINTG
299    RINTG
300    RINTG
301    RINTG
302    RINTG
303    RINTG
304    RINTG
305    RINTG
306    RINTG
307    RINTG
308    RINTG
309    RINTG
310    RINTG
311    RINTG
312    RINTG
313    RINTG
314    RINTG
315    RINTG
316    RINTG
317    RINTG
318    RINTG
319    RINTG
320    RINTG
321    RINTG
322    RINTG
323    RINTG
324    RINTG
325    RINTG
326    RINTG
327    RINTG
328    RINTG
329    RINTG
330    RINTG
331    RINTG
332    RINTG
333    RINTG
334    RINTG
335    RINTG
336    RINTG
337    RINTG
338    RINTG
339    RINTG
340    RINTG
341    RINTG
342    RINTG
343    RINTG
344    RINTG
345    RINTG
346    RINTG
347    RINTG
348    RINTG
349    RINTG
350    RINTG
351    RINTG
352    RINTG
353    RINTG
354    RINTG
355    RINTG
356    RINTG
357    RINTG
358    RINTG
359    RINTG
360    RINTG
361    RINTG
362    RINTG
363    RINTG
364    RINTG
365    RINTG
366    RINTG
367    RINTG
368    RINTG
369    RINTG
370    RINTG
371    RINTG
372    RINTG
373    RINTG
374    RINTG
375    RINTG
376    RINTG
377    RINTG
378    RINTG
379    RINTG
380    RINTG
381    RINTG
382    RINTG
383    RINTG
384    RINTG
385    RINTG
386    RINTG
387    RINTG
388    RINTG
389    RINTG
390    RINTG
391    RINTG
392    RINTG
393    RINTG
394    RINTG
395    RINTG
396    RINTG
397    RINTG
398    RINTG
399    RINTG
400    RINTG
401    RINTG
402    RINTG
403    RINTG
404    RINTG
405    RINTG
406    RINTG
407    RINTG
408    RINTG
409    RINTG
410    RINTG
411    RINTG
412    RINTG
413    RINTG
414    RINTG
415    RINTG
416    RINTG
417    RINTG
418    RINTG
419    RINTG
420    RINTG
421    RINTG
422    RINTG
423    RINTG
424    RINTG
425    RINTG
426    RINTG
427    RINTG
428    RINTG
429    RINTG
430    RINTG
431    RINTG
432    RINTG
433    RINTG
434    RINTG
435    RINTG
436    RINTG
437    RINTG
438    RINTG
439    RINTG
440    RINTG
441    RINTG
442    RINTG
443    RINTG
444    RINTG
445    RINTG
446    RINTG
447    RINTG
448    RINTG
449    RINTG
450    RINTG
451    RINTG
452    RINTG
453    RINTG
454    RINTG
455    RINTG
456    RINTG
457    RINTG
458    RINTG
459    RINTG
460    RINTG
461    RINTG
462    RINTG
463    RINTG
464    RINTG
465    RINTG
466    RINTG
467    RINTG
468    RINTG
469    RINTG
470    RINTG
471    RINTG
472    RINTG
473    RINTG
474    RINTG
475    RINTG
476    RINTG
477    RINTG
478    RINTG
479    RINTG
480    RINTG
481    RINTG
482    RINTG
483    RINTG
484    RINTG
485    RINTG
486    RINTG
487    RINTG
488    RINTG
489    RINTG
490    RINTG
491    RINTG
492    RINTG
493    RINTG
494    RINTG
495    RINTG
496    RINTG
497    RINTG
498    RINTG
499    RINTG
500    RINTG
501    RINTG
502    RINTG
503    RINTG
504    RINTG
505    RINTG
506    RINTG
507    RINTG
508    RINTG
509    RINTG
510    RINTG
511    RINTG
512    RINTG
513    RINTG
514    RINTG
515    RINTG
516    RINTG
517    RINTG
518    RINTG
519    RINTG
520    RINTG
521    RINTG
522    RINTG
523    RINTG
524    RINTG
525    RINTG
526    RINTG
527    RINTG
528    RINTG
529    RINTG
530    RINTG
531    RINTG
532    RINTG
533    RINTG
534    RINTG
535    RINTG
536    RINTG
537    RINTG
538    RINTG
539    RINTG
540    RINTG
541    RINTG
542    RINTG
543    RINTG
544    RINTG
545    RINTG
546    RINTG
547    RINTG
548    RINTG
549    RINTG
550    RINTG
551    RINTG
552    RINTG
553    RINTG
554    RINTG
555    RINTG
556    RINTG
557    RINTG
558    RINTG
559    RINTG
560    RINTG
561    RINTG
562    RINTG
563    RINTG
564    RINTG
565    RINTG
566    RINTG
567    RINTG
568    RINTG
569    RINTG
570    RINTG
571    RINTG
572    RINTG
573    RINTG
574    RINTG
575    RINTG
576    RINTG
577    RINTG
578    RINTG
579    RINTG
580    RINTG
581    RINTG
582    RINTG
583    RINTG
584    RINTG
585    RINTG
586    RINTG
587    RINTG
588    RINTG
589    RINTG
590    RINTG
591    RINTG
592    RINTG
593    RINTG
594    RINTG
595    RINTG
596    RINTG
597    RINTG
598    RINTG
599    RINTG
600    RINTG
601    RINTG
602    RINTG
603    RINTG
604    RINTG
605    RINTG
606    RINTG
607    RINTG
608    RINTG
609    RINTG
610    RINTG
611    RINTG
612    RINTG
613    RINTG
614    RINTG
615    RINTG
616    RINTG
617    RINTG
618    RINTG
619    RINTG
620    RINTG
621    RINTG
622    RINTG
623    RINTG
624    RINTG
625    RINTG
626    RINTG
627    RINTG
628    RINTG
629    RINTG
630    RINTG
631    RINTG
632    RINTG
633    RINTG
634    RINTG
635    RINTG
636    RINTG
637    RINTG
638    RINTG
639    RINTG
640    RINTG
641    RINTG
642    RINTG
643    RINTG
644    RINTG
645    RINTG
646    RINTG
647    RINTG
648    RINTG
649    RINTG
650    RINTG
651    RINTG
652    RINTG
653    RINTG
654    RINTG
655    RINTG
656    RINTG
657    RINTG
658    RINTG
659    RINTG
660    RINTG
661    RINTG
662    RINTG
663    RINTG
664    RINTG
665    RINTG
666    RINTG
667    RINTG
668    RINTG
669    RINTG
670    RINTG
671    RINTG
672    RINTG
673    RINTG
674    RINTG
675    RINTG
676    RINTG
677    RINTG
678    RINTG
679    RINTG
680    RINTG
681    RINTG
682    RINTG
683    RINTG
684    RINTG
685    RINTG
686    RINTG
687    RINTG
688    RINTG
689    RINTG
690    RINTG
691    RINTG
692    RINTG
693    RINTG
694    RINTG
695    RINTG
696    RINTG
697    RINTG
698    RINTG
699    RINTG
700    RINTG
701    RINTG
702    RINTG
703    RINTG
704    RINTG
705    RINTG
706    RINTG
707    RINTG
708    RINTG
709    RINTG
710    RINTG
711    RINTG
712    RINTG
713    RINTG
714    RINTG
715    RINTG
716    RINTG
717    RINTG
718    RINTG
719    RINTG
720    RINTG
721    RINTG
722    RINTG
723    RINTG
724    RINTG
725    RINTG
726    RINTG
727    RINTG
728    RINTG
729    RINTG
730    RINTG
731    RINTG
732    RINTG
733    RINTG
734    RINTG
735    RINTG
736    RINTG
737    RINTG
738    RINTG
739    RINTG
740    RINTG
741    RINTG
742    RINTG
743    RINTG
744    RINTG
745    RINTG
746    RINTG
747    RINTG
748    RINTG
749    RINTG
750    RINTG
751    RINTG
752    RINTG
753    RINTG
754    RINTG
755    RINTG
756    RINTG
757    RINTG
758    RINTG
759    RINTG
760    RINTG
761    RINTG
762    RINTG
763    RINTG
764    RINTG
765    RINTG
766    RINTG
767    RINTG
768    RINTG
769    RINTG
770    RINTG
771    RINTG
772    RINTG
773    RINTG
774    RINTG
775    RINTG
776    RINTG
777    RINTG
778    RINTG
779    RINTG
780    RINTG
781    RINTG
782    RINTG
783    RINTG
784    RINTG
785    RINTG
786    RINTG
787    RINTG
788    RINTG
789    RINTG
790    RINTG
791    RINTG
792    RINTG
793    RINTG
794    RINTG
795    RINTG
796    RINTG
797    RINTG
798    RINTG
799    RINTG
800    RINTG
801    RINTG
802    RINTG
803    RINTG
804    RINTG
805    RINTG
806    RINTG
807    RINTG
808    RINTG
809    RINTG
810    RINTG
811    RINTG
812    RINTG
813    RINTG
814    RINTG
815    RINTG
816    RINTG
817    RINTG
818    RINTG
819    RINTG
820    RINTG
821    RINTG
822    RINTG
823    RINTG
824    RINTG
825    RINTG
826    RINTG
827    RINTG
828    RINTG
829    RINTG
830    RINTG
831    RINTG
832    RINTG
833    RINTG
834    RINTG
835    RINTG
836    RINTG
837    RINTG
838    RINTG
839    RINTG
840    RINTG
841    RINTG
842    RINTG
843    RINTG
844    RINTG
845    RINTG
846    RINTG
847    RINTG
848    RINTG
849    RINTG
850    RINTG
851    RINTG
852    RINTG
853    RINTG
854    RINTG
855    RINTG
856    RINTG
857    RINTG
858    RINTG
859    RINTG
860    RINTG
861    RINTG
862    RINTG
863    RINTG
864    RINTG
865    RINTG
866    RINTG
867    RINTG
868    RINTG
869    RINTG
870    RINTG
871    RINTG
872    RINTG
873    RINTG
874    RINTG
875    RINTG
876    RINTG
877    RINTG
878    RINTG
879    RINTG
880    RINTG
881    RINTG
882    RINTG
883    RINTG
884    RINTG
885    RINTG
886    RINTG
887    RINTG
888    RINTG
889    RINTG
890    RINTG
891    RINTG
892    RINTG
893    RINTG
894    RINTG
895    RINTG
896    RINTG
897    RINTG
898    RINTG
899    RINTG
900    RINTG
901    RINTG
902    RINTG
903    RINTG
904    RINTG
905    RINTG
906    RINTG
907    RINTG
908    RINTG
909    RINTG
910    RINTG
911    RINTG
912    RINTG
913    RINTG
914    RINTG
915    RINTG
916    RINTG
917    RINTG
918    RINTG
919    RINTG
920    RINTG
921    RINTG
922    RINTG
923    RINTG
924    RINTG
925    RINTG
926    RINTG
927    RINTG
928    RINTG
929    RINTG
930    RINTG
931    RINTG
932    RINTG
933    RINTG
934    RINTG
935    RINTG
936    RINTG
937    RINTG
938    RINTG
939    RINTG
940    RINTG
941    RINTG
942    RINTG
943    RINTG
944    RINTG
945    RINTG
946    RINTG
947    RINTG
948    RINTG
949    RINTG
950    RINTG
951    RINTG
952    RINTG
953    RINTG
954    RINTG
955    RINTG
956    RINTG
957    RINTG
958    RINTG
959    RINTG
960    RINTG
961    RINTG
962    RINTG
963    RINTG
964    RINTG
965    RINTG
966    RINTG
967    RINTG
968    RINTG
969    RINTG
970    RINTG
971    RINTG
972    RINTG
973    RINTG
974    RINTG
975    RINTG
976    RINTG
977    RINTG
978    RINTG
979    RINTG
980    RINTG
981    RINTG
982    RINTG
983    RINTG
984    RINTG
985    RINTG
986    RINTG
987    RINTG
988    RINTG
989    RINTG
990    RINTG
991    RINTG
992    RINTG
993    RINTG
994    RINTG
995    RINTG
996    RINTG
997    RINTG
998    RINTG
999    RINTG
1000   RINTG

```

13.06.42

04/07/81

FTN 4.6+460

SUBROUTINE RINTEG 74/74 OPT=0 ROUND=0/ TRACE

```

60      MAA = 2*M
      MBB = 2*M+1
      DO 55 N=M,MMAX
      NAA = 2*N
      NBB = 2*N+1
      TSA = WTINT*ES-VJALFA(MAA)*VJALFA(NAA)*CON
      WSA = WTINT*ES-VJALFA(MBB)*VJALFA(NBB)*CON
      TBA = WTINT*EB-VJBETA(MAA)*VJBETA(NAA)
      WBA = WTINT*EB-VJBETA(MBB)*VJBETA(NBB)
      TSBA = -WTINT*ESB-VJBETA(NAA)*VJALFA(MAA)*CO
      TSBA = -WTINT*ESB-VJBETA(NBB)*VJALFA(MBB)*CO
      WBSA = WTINT*ESB-VJBETA(NBB)*VJALFA(MBB)*SI
      WBSA = WTINT*ESB-VJBETA(NBB)*VJALFA(MBB)*SI
      TSBB = -WTINT*ESB-VJBETA(MAA)*VJALFA(NAA)*CO
      TSBB = -WTINT*ESB-VJBETA(MAA)*VJALFA(NAA)*CO
      WBSB = WTINT*ESB-VJBETA(MBB)*VJALFA(NBB)*CO
      WBSB = WTINT*ESB-VJBETA(MBB)*VJALFA(NBB)*CO
      TBB = TB(M,N)+TBA
      WBB = WB(M,N)+WBA
      TSBB(M,N) = TSBB(M,N)+TSBA
      TSBB(M,N) = TSBB(M,N)+TSBA
      WBSB(M,N) = WBSB(M,N)+WSBA
      WBSB(M,N) = WBSB(M,N)+WSBA
      IF(M.EQ.N) GO TO 54
      TSBN(M) = TSBN(M)+TSBB
      TSBN(M) = TSBN(M)+TSBBP
      WBSN(M) = WBSN(M)+WSBB
      WBSN(M) = WBSN(M)+WSBBP
      CONTINUE
      TS(M,N) = TS(M,N)+TSA
      WS(M,N) = WS(M,N)+WSA
      DO 55 L=1,NDSEPS
      TS12(L,M,N) = TS12(L,M,N)+TSA*SEPCOS(L)
      WS12(L,M,N) = WS12(L,M,N)+WSA*SEPCOS(L)
      IF(EBA.EQ.0.0) GO TO 55
      TB12(L,M,N) = TB12(L,M,N)+TBA*SEPCOS(L)
      WB12(L,M,N) = WB12(L,M,N)+WBA*SEPCOS(L)
      TS12(L,M,N) = TS12(L,M,N)+TSBA*SEPCOS(L)
      TS12P(L,M,N) = TS12P(L,M,N)+TSBAP*SEPCOS(L)
      WS12(L,M,N) = WS12(L,M,N)+WSBA*SEPCOS(L)
      WS12P(L,M,N) = WS12P(L,M,N)+WSBAP*SEPCOS(L)
      IF(M.EQ.N) GO TO 55
      TS12(L,N,M) = TS12(L,N,M)+TSBB*SEPCOS(L)
      TS12P(L,N,M) = TS12P(L,N,M)+TSBBP*SEPCOS(L)
      WS12(L,N,M) = WS12(L,N,M)+WSBB*SEPCOS(L)
      WS12P(L,N,M) = WS12P(L,N,M)+WSBBP*SEPCOS(L)
      CONTINUE
      RETURN
      END
55

```

13.06.42

04/07/81

FTN 4.6+460

TRACE

SUBROUTINE REFLK

74/74

OPT=0

ROUND=

```

1      SUBROUTINE REFLK
2      C
3      C
4      C
5      C
6      C
7      C
8      C
9      C
10     C
11     C
12     C
13     C
14     C
15     C
16     C
17     C
18     C
19     C
20     C
21     C
22     C
23     C
24     C
25     C
26     C
27     C
28     C
29     C
30     C
31     C
32     C
33     C
34     C
35     C
36     C
37     C
38     C
39     C
40     C
41     C
42     C
43     C
44     C
45     C
46     C
47     C
48     C
49     C
50     C
51     C
52     C
53     C
54     C
55     C
56     C
57     C
58     C

```

REFLECT THE SYMMETRICAL MATRICES OF T AND W

COMMON/AUX/TS(10,10),WS(10,10),TB(10,10),WB(10,10),TSB(10,10),
 WSB(10,10),TS12(8,10,10),WS12(8,10,10),TB12(8,10,10),
 WB12(8,10,10),TSB12(8,10,10),WSB12(8,10,10),
 TSBP(10,10),WSBP(10,10),TSB12P(9,10,10),WSB12P(8,10,10)

COMMON/OMEGA/MAX,HSOLB,HBOLB,GAMAOS,GAMA0B,NOSEPS,
 SEPDIS(8),SEP(8),CSTRUT(8),CSTR2(8),GOSQ,PHIS,PHIB,
 RATIOI

DO 63 M=1,MMAX
 EM=FLOAT(M)
 U = 2.0*EM-1.0
 DO 63 N=M,MMAX
 EN=FLOAT(N)
 FOURMN=4.0*EN*EN
 UV=U*(2.0*EN-1.0)
 TS(M,N) = TS(M,N)*UV
 TS(N,M) = TS(M,N)
 WS(M,N) = WS(M,N)*FOURMN
 WS(N,M) = WS(M,N)
 TB(M,N) = TB(M,N)*UV
 TB(N,M) = TB(M,N)
 WB(M,N) = WB(M,N)*FOURMN
 WB(N,M) = WB(M,N)
 TSB(M,N) = TSB(M,N)*UV
 TSBP(M,N) = TSBP(M,N)*UV
 WSB(M,N) = WSB(M,N)*FOURMN
 WSBP(M,N) = WSBP(M,N)*FOURMN
 IF (M.EQ.N) GO TO 62
 TSB(N,M)=TSB(M,N)*UV
 TSBP(N,M)=TSBP(M,N)*UV
 WSB(N,M)=WSB(M,N)*FOURMN
 WSBP(N,M)=WSBP(M,N)*FOURMN
 CONTINUE

DO 53 L=1,NOSEPS
 TS12(L,M,N) = TS12(L,M,N)*UV
 TS12(L,N,M) = TS12(L,M,N)
 WS12(L,M,N) = WS12(L,M,N)*FOURMN
 WS12(L,N,M) = WS12(L,M,N)
 TB12(L,M,N) = TB12(L,M,N)*UV
 TB12(L,N,M) = TB12(L,M,N)
 WB12(L,M,N) = WB12(L,M,N)*FOURMN
 WB12(L,N,M) = WB12(L,M,N)
 TSB12(L,M,N) = TSB12(L,M,N)*UV
 TSB12P(L,M,N) = TSB12P(L,M,N)*UV
 WSB12(L,M,N) = WSB12(L,M,N)*FOURMN
 WSB12P(L,M,N) = WSB12P(L,M,N)*FOURMN
 IF (M.EQ.N) GO TO 63
 TSB12(L,N,M)=TSB12(L,M,N)*UV
 TSB12P(L,N,M)=TSB12P(L,M,N)*UV
 WSB12(L,N,M)=WSB12(L,M,N)*FOURMN

04/07/81 13.06.42

FTN 4.6+460

SUBROUTINE REFLK 74/74 OPT=0 ROUND=0/ TRACE

60
 63
 WSB12P(L,N,M)=WSB12P(L,N,M)*FOURMN
 CONTINUE
 RETURN
 END
 REFLK 59
 REFLK 60
 REFLK 61
 REFLK 62

04/07/81 13.06.42

FTN 4.6+460

74/74 OPT=0 ROUND=0/ TRACE

SUBROUTINE BESSJ

```

1      SUBROUTINE BESSJ(X,N,VJ)
2
3      C
4      EVALUATE THE VALUES OF BESSEL FUNCTION FROM ORDER 0 TO
5      ORDER N WITH ARGUMENT X
6
7      REAL J, J1, JLL, JLLL
8      DIMENSION VJ(N)
9      DATA PI /3.1415926535898/
10     EPS /1.0E-30/
11     EXPON /2.7182818284590/
12
13     C
14     DETERMINE THE MAXIMUM N = NU
15
16     D = ALOG(EPS)/X
17     C = ALOG(EXPON/2.0)
18     B = D-2.0-C
19     ALPHA = (-B+SQRT(B**2-4.0*D*(2.0-C)))/(2.0*(2.0-C))
20     CLN = ALOG(2.0*PI-X)
21     IF(ALPHA.LT.1.5) ALPHA = 1.5
22     DO 1000 I=1,20
23         ALN = ALOG(ALPHA)
24         F = -(CLN+ALN)/(2.0*X)+ALPHA*(C-ALN)-D
25         FP = -1.0/(2.0*X*ALPHA)+C-1.0-ALN
26         DEL = F/FP
27         ALPHA = ALPHA-DEL
28         IF(X*ABS(DEL) .LE. 0.01) GO TO 1100
29
30     CONTINUE
31     GO TO 8888
32
33     1000 CONTINUE
34     GNU = ALPHA*X
35     NU = IFIX(GNU)+1
36     IF(NU.LT.N+10) NU = N+10
37     IF((NU/2)*2.NE.NU) NU=NU+1
38
39     C
40     RECURSE IN PLACE FOR M.GT.N
41
42     SUM = 0.0
43     JLLL = 0.0
44     JLL = EPS
45     NUM1 = NU-1
46     NP = N
47     IF((NP/2)*2.EQ.NP) NP = NP-1
48     DO 1200 MM = NP,NUM1,2
49         W = NUM1-MM+NP
50         JL = 2.0*FLOAT(MM+2)*JLL/X-JLLL
51         J = 2.0*FLOAT(MM+1)*JL/X-JLL
52         SUM = SUM+2.0*JL
53         JLLL = JL
54         JLL = J
55     CONTINUE
56     IF(NP.NE.N) GO TO 1300
57     JL = 2.0*FLOAT(N)*JLL/X-JLLL
58     SUM = SUM+2.0*JL
59     JLLL = JLL
60     JLL = JL
61     CONTINUE
62     I = N+1

```

```

60      VJ(I) = JLL
        VJ(N) = JLL
        NP = N-1
        DO 1400 MM = 1,NP
            M = NP-MM
            K = M+1
            VJ(K) = 2.0*FLOAT(K)*VJ(M+2)/X-VJ(M+3)
            IF((M/2)*2.EQ.M) SUM = SUM+2*VJ(M+1)
1400    CONTINUE
        SUM = SUM-VJ(1)
        K = N+1
        DO 1500 M = 1,K
            VJ(M) = VJ(M)/SUM
1500    CONTINUE
        RETURN
88888   CONTINUE
        WRITE(6,100) X, ALPHA, DEL
        STOP
100     FORMAT(28H NO CONVERGENCE, X,ALPHA,DEL/3E14.7)
        END

```

```

BESSJ 59
BESSJ 60
BESSJ 61
BESSJ 62
BESSJ 63
BESSJ 64
BESSJ 65
BESSJ 66
BESSJ 67
BESSJ 68
BESSJ 69
BESSJ 70
BESSJ 71
BESSJ 72
BESSJ 73
BESSJ 74
BESSJ 75
BESSJ 76
BESSJ 77
BESSJ 78

```

```

1      SUBROUTINE PCHEB (AS, BS, AB, BB, NN, TITLE)
2      PCHEB
3      PCHEB
4      PCHEB
5      PCHEB
6      PCHEB
7      PCHEB
8      PCHEB
9      PCHEB
10     PCHEB
11     PCHEB
12     PCHEB
13     PCHEB
14     PCHEB
15     PCHEB
16     PCHEB
17     PCHEB
18     PCHEB
19     PCHEB
20     PCHEB
21     PCHEB
22     PCHEB
23     PCHEB
24     PCHEB
25     PCHEB
26     PCHEB
27     PCHEB
28     PCHEB
29     PCHEB
30     PCHEB
31     PCHEB
32     PCHEB
33     PCHEB
34     PCHEB
35     PCHEB
36     PCHEB
37     PCHEB
38     PCHEB
39     PCHEB
40     PCHEB
41     PCHEB
42     PCHEB
43     PCHEB
44     PCHEB
45     PCHEB
46     PCHEB
47     PCHEB
48     PCHEB
49     PCHEB
50     PCHEB
51     PCHEB
52     PCHEB
53     PCHEB
54     PCHEB
55     PCHEB
56     PCHEB
57     PCHEB
58     PCHEB

      PLOT BODY SECTIONAL AREA CURVE AND STRUT WATERPLANE OUTLINE CURVE
      INPUTS
      AS AND BS ARE, RESPECTIVELY, THE VECTORS OF SYMMETRIC-
      AND OF ANTISYMMETRIC-CHEBYCHEV COEFFICIENTS FOR THE STRUT.
      AB AND BB ARE, RESPECTIVELY, THE VECTORS OF SYMMETRIC-
      AND OF ANTISYMMETRIC-CHEBYCHEV COEFFICIENTS FOR THE BODY.

      NN = DIMENSION OF AS, BS, AB, BB
      MAXIMUM VALUE OF NN IS NMAX
      NMAX = 10

      INITIALIZE THE GRID GEOMETRY, THE PLOT VARIABLES AND
      THE ABSCESSA VARIABLE.

      COMMON/PLOT/NFIRST,NLAST,NPOINT,XMAX,XMIN,NSCL1,NCHAR,NSCALE(4),
1      PCHAR(2)
      COMMON/XRPLQTQ/II,JJ,KK,LL,NHL,NSBH,NVL,NSBV,HCHAR,VCHAR,
1      ISX,ISY,V,H

      LOGICAL NSCL1

      DIMENSION AS(NN), BS(NN), AB(NN), BB(NN), TITLE(8)
      DIMENSION STAT(101),YS(101),YB(101),LABLE(3)

      GENERATE ORDNATE VALUES FOR STRUT AND BODY POINTS

      DO 2000 I=1, 101
        STAT(I)=2.0*(FLOAT(I-1)*.01-.5)
        THETA=ASIN(STAT(I))
        YS(I)=0.0
        YB(I)=0.0
        DO 1000 NCO=1,NN
          UM=COS(FLOAT(2*NCO-1)*THETA)
          VM=SIN(FLOAT(2*NCO)*THETA)
          YS(I)=YS(I)+AS(NCO)*UM+BS(NCO)*VM
          YB(I)=YB(I)+AB(NCO)*UM+BB(NCO)*VM
        1000 CONTINUE
      2000 CONTINUE

      DETERMINE THE MINIMUM AND MAXIMUM ORDNATE VALUES

      YMAX = -999999.0
      YMIN = +999999.0
      DO 3000 J=1,101
        IF(YMAX.LT. YB(J)) YMAX=YB(J)
        IF(YMAX.LT. YS(J)) YMAX=YS(J)
        IF(YMIN.GT. YB(J)) YMIN=YB(J)
        IF(YMIN.GT. YS(J)) YMIN=YS(J)
      3000 CONTINUE
    
```

```

C      CALL PLOT1 TO SET UP GRID SPACING AND DETERMINE THE AXIS VALUES
C
C      CALL PLOT1 (NSCALE,NHL,NSBH,NVL,NSBV,HCHAR,VCHAR)
C
C      CALL PLOT2 TO EXAMINE THE MINIMUM AND MAXIMUM VALUES OF
C      ABSCISSA AND ORDINATE AND TO ESTABLISH AN INTERNAL FORMULA
C      FOR COMPUTING THE LOCATION IN THE IMAGE REGION CORRESPONDING
C      TO THE POINT TO BE PLOTTED.
C      WHEN NSCL1=.TRUE., THE STANDARD GRID WILL NOT BE USED
C
C      CALL PLOT2 (XMAX,XMIN,YMAX,YMIN,NSCL1)
C
C      CALL PLOT3 FOR ASSIGNING AN ALPHA-CHARACTER TO EACH POINT
C      WHICH WILL BE PLOTTED.
C
C      CALL PLOT3 (PCHAR(1),STAT,YS,NFIRST,NLAST,NPOINT)
C      CALL PLOT3 (PCHAR(2),STAT,YB,NFIRST,NLAST,NPOINT)
C
C      PRINT THE HEADING OF THE PLOT
C
C      WRITE (6,9000) (TITLE(I), I=1,8)
C
C      CALL PLOT4 TO PRINT THE IMAGE OF THE COMPLETED GRAPH ON THE
C      PRINTER, INCLUDING THE VALUES OF ABSCISSA AND ORDINATE AT
C      THE GRID LINES OUTSIDE THE BOTTOM AND LEFT EDGE OF THE GRAPH
C
C      CALL PLOT4 (LABLE,NCHAR)
C
C      RETURN
C
9000 FORMAT(1H1,32X,64HBODY SECTIONAL AREA CURVE AND STRUT WATERPLANE O
1UTLINE CURVE FOR//25X,8A10)
END

```

PCHEB 59
PCHEB 60
PCHEB 61
PCHEB 62
PCHEB 63
PCHEB 64
PCHEB 65
PCHEB 66
PCHEB 67
PCHEB 68
PCHEB 69
PCHEB 70
PCHEB 71
PCHEB 72
PCHEB 73
PCHEB 74
PCHEB 75
PCHEB 76
PCHEB 77
PCHEB 78
PCHEB 79
PCHEB 80
PCHEB 81
PCHEB 82
PCHEB 83
PCHEB 84
PCHEB 85
PCHEB 86
PCHEB 87
PCHEB 88
PCHEB 89
PCHEB 90
PCHEB 91
PCHEB 92

```

1      C
      SUBROUTINE PLOT1 (NSCALE,A,B,C,D,E,F)
      SETUP SPACING AND DETERMINE THE AXIS VALUES
5      C
      INTEGER A,B,C,D
      LOGICAL V,H
      DIMENSION NSCALE(4)
10     C
      COMMON/XRPLQTQ/I,J,K,L,NHL,NSBH,NVL,NSBV,HCHAR,VCHAR,ISX,ISY,V,H
      C
      I=NSCALE(1)
      J=NSCALE(2)
      K=NSCALE(3)
      L=NSCALE(4)
      NHL=A
      NSBH=B-1
      NVL=C
      NSBV=D-1
      HCHAR=E
      VCHAR=F
      RETURN
      END
2      PLOT1
3      PLOT1
4      PLOT1
5      PLOT1
6      PLOT1
7      PLOT1
8      PLOT1
9      PLOT1
10     PLOT1
11     PLOT1
12     PLOT1
13     PLOT1
14     PLOT1
15     PLOT1
16     PLOT1
17     PLOT1
18     PLOT1
19     PLOT1
20     PLOT1
21     PLOT1
22     PLOT1
23     PLOT1
24     PLOT1
25     PLOT1

```



```

60      7 CONTINUE
        HCT=HCT+1
        IQ=IX+1
        DO 10 J=IQ,110
            LIN(J)=1L
10      CONTINUE
        ENCODE(110,9,GRAF(1,1)) (LIN(J),J=1,110)
65      2 CONTINUE
        3 CONTINUE
        XL=XMIN
        XH=XMAX
        YL=YMIN
        YH=YMAX
        XI=(XH-XL)/FLOAT(IX-1)
        YI=(YH-YL)/FLOAT(IY-1)
        XMOV=1.0-XL/XI
        YMOV=1.0-YL/YI
        V=.TRUE.
        H=.TRUE.
        RETURN
        9 FORMAT(110A1)
        END
70
75
59      PLOT2
60      PLOT2
61      PLOT2
62      PLOT2
63      PLOT2
64      PLOT2
65      PLOT2
66      PLOT2
67      PLOT2
68      PLOT2
69      PLOT2
70      PLOT2
71      PLOT2
72      PLOT2
73      PLOT2
74      PLOT2
75      PLOT2
76      PLOT2
77      PLOT2
78      PLOT2
79      PLOT2
80      PLOT2

```



```

1      SUBROUTINE PLOT4(MCHAR,NCHAR)
      C
      C      PRINT THE IMAGE OF THE COMPLETED GRAPH ON THE PRINTER.
      C      INCLUDING THE VALUES OF THE ABSCISSA AND ORDINATE AT THE
      C      GRID LINES OUTSIDE THE BOTTOM AND LEFT EDGES OF THE GRAPH
      C
      C
      C      INTEGER SA, SB, SC
      C      INTEGER CT, PDQ
      C      LOGICAL V, H
      C
      C      DIMENSION HNUM(15), MCHAR(3), LCHAR(30), VGFMT(3), HLFMT(3)
      C
      C      COMMON/XRPLDTF/XL,XH,YL,YH,XI,YI,XMOV,YMOV
      C      COMMON/XRPLDTG/GRAF(11,204)
      C      COMMON/XRPLDTQ/I,J,K,L,NHL,NSBH,NVL,NSBV,MCHAR,VCHAR,ISX,ISY,V,H
      C
      C      SA=NSBV
      C      SC=NCHAR
      C      SB=NVL
      C      KQ=1
      C      CALL QPLOTZ5(PDQ)
      C      ENCODE(26.2,VGFMT(1))J
      C      IF(NSBV.LE.0) NSBV=10
      C
      C      4 CONTINUE
      C      IF(NSBV.GT. 10) GO TO 3
      C      NSBV=NSBV*2
      C      NVL=NVL
      C      NVL=XVL/2.0+.5
      C      KQ=KQ+1
      C      GO TO 4
      C
      C      3 CONTINUE
      C      NNN=NVL-1
      C      MMM=NSBV-8
      C      ENCODE(25.5,HLFMT(1)) PDQ,L,NNN,MMM,L
      C      IF (NCHAR.NE.0) GOTO 14
      C      NCHAR=1
      C      LCHAR(1)=1L
      C      GO TO 15
      C
      C      14 CONTINUE
      C      DECODE(10,17,MCHAR(1)) (LCHAR(IJ),IJ=1,10)
      C      DECODE(10,17,MCHAR(2)) (LCHAR(IJ),IJ=11,20)
      C      DECODE(10,17,MCHAR(3)) (LCHAR(IJ),IJ=21,30)
      C
      C      15 CONTINUE
      C      NCT=0
      C      NSPC=(ISY-NCHAR)/2
      C      LCT=0
      C      CT=NSBH
      C      DO 6 MMX=1,ISV
      C      N=ISY-MMX+1
      C      IF (MMX.LE.NSPC.OR.LCT.GE.NCHAR) GO TO 7
      C      LCT=LCT+1
      C      LCR=LCHAR(LCT)
      C      GO TO 8
      C
      C      7 CONTINUE
      C      LCR=1L
      C
      C      8 CONTINUE

```

04/07/81 13.06.42

FTN 4.6+460

SUBROUTINE PLOT4 74/74 OPT=0 RCUND=0/ TRACE

```

60      IF (CT.EQ.NSBM.AND.H) GO TO 9
        PRINT 1,LCR,(GRAF(II,N),II=1,11)
        GO TO 13
9      CONTINUE
        CT=-1
        ONN=(YH-MCT*YI)*10.**I
        PRINT VGFMT,LCR,ONN,(GRAF(II,N),II=1,11)
65      MCT=MCT+NSBH+1
        CONTINUE
        CT=CT+1
6      CONTINUE
        MCT=0
        IF (.NOT.V) GO TO 12
        DO 10 N=1,NVL
            HNUM(N)=(XL+MCT*XI)*10.**K
            MCT=MCT+NSBV+KQ
10      CONTINUE
        PRINT 11
        PRINT HLFMT,(HNUM(N),N=1,NVL)
12      CONTINUE
        NSBV=SA
        NVL=SB
        NCHAR=SC
        RETURN
        1  FORMAT(1H,A1,16X,11A10)
        2  FORMAT(15H(1H ,A1,3X,F10.,I1,10H,3X,11A10))
85      5  FORMAT(1H( ,I2,6HX,F10.,I1,1H, ,I2,1H( ,I2,6HX,F10.,I1,2H)))
        11 FORMAT(1H0)
        17 FORMAT(10A1)
        END
59      PLOT4
60      PLOT4
61      PLOT4
62      PLOT4
63      PLOT4
64      PLOT4
65      PLOT4
66      PLOT4
67      PLOT4
68      PLOT4
69      PLOT4
70      PLOT4
71      PLOT4
72      PLOT4
73      PLOT4
74      PLOT4
75      PLOT4
76      PLOT4
77      PLOT4
78      PLOT4
79      PLOT4
80      PLOT4
81      PLOT4
82      PLOT4
83      PLOT4
84      PLOT4
85      PLOT4
86      PLOT4
87      PLOT4
88      PLOT4

```

```

1      SUBROUTINE QPLOT25(PDQ)
      C
      INTEGER A,B,C,D,PDQ
      LOGICAL V,H
      COMMON/XRPLDTF/XL,XH,YL,YH,XI,YI,XM,YM
      COMMON/XRPLDTQ/II,JJ,KK,LL,A,B,C,D,E,F,M,N,V,H
      C
      X=ABS(XL)
      IF(X.LT.AH) X=XH
      K=0
      DO 1 I=1,11
        XU=10** (I-1)
        IF(X.GT.XU) GOTO 2
        K=K+1
      1 CONTINUE
      2 CONTINUE
      PDQ=1
      IF(LL.EQ.0) PDQ=0
      QDQ=PDQ+KK+LL+K
      PDQ=14.0-(10.0-QDQ)/2.0
      RETURN
      END
20
2      QPLOT
3      QPLOT
4      QPLOT
5      QPLOT
6      QPLOT
7      QPLOT
8      QPLOT
9      QPLOT
10     QPLOT
11     QPLOT
12     QPLOT
13     QPLOT
14     QPLOT
15     QPLOT
16     QPLOT
17     QPLOT
18     QPLOT
19     QPLOT
20     QPLOT
21     QPLOT
22     QPLOT
23     QPLOT
24     QPLOT

```

```

1      SUBROUTINE CHEB (AMC,BMC,MMAX)
2      C
3      C INPUT STRUT OR BODY DATA. COMPUTE CHEBYCHEV COEFFICIENTS
4      C CHECK THE ACCURACY OF CHEBYCHEV APPROXIMATION.
5      C
6      C DIMENSION AMC(10), BMC(10)
7      C
8      C COMMON/NAME/INCHAR(8),NAME(7)
9      C
10     COMMON/OFFSET/X(100),Y(100),D(100),THETA(100),N,NV(10),NNV
11     LOGICAL ISBODY
12     C
13     C READ INPUT OFFSET DATA
14     C
15     C READ (5,31) INCHAR
16     C IF (INCHAR(1).EQ.5)STRUT) ISBODY=.FALSE.
17     C IF (INCHAR(1).EQ. 4)BODY) ISBODY=.TRUE.
18     C
19     C CALL READ
20     C
21     C COMPUTE SCALING FACTOR
22     C
23     C CALL SCALE(ISBODY)
24     C
25     C COMPUTE CHEBYCHEV COEFFICIENTS
26     C
27     C CALL COMPUT (AMC,BMC,MMAX,ISBODY)
28     C
29     C CHECK INPUT OFFSET AGAINST OFFSET OBTAINED BY EVALUATING
30     C THE CHEBYCHEV SERIES
31     C
32     C CALL CHECK (AMC,BMC,MMAX,ISBODY)
33     C
34     C COMPUTE WETTED SURFACE AREA
35     C
36     C CALL SURFACE (AMC,BMC,MMAX,ISBODY)
37     C
38     C CHECK NUMBER OF OFFSET INPUT CARDS
39     C
40     C READ (5,32) NINPUT
41     C NINPUT = NINPUT + NNV - 1
42     C IF (NINPUT.EQ.N) GO TO 999
43     C
44     C ERROR IN INPUT CARDS
45     C
46     C ARITE (6,200) INCHAR
47     C STOP
48     C 999 CONTINUE
49     C RETURN
50     C
51     C 31 FORMAT (3A10)
52     C 32 FORMAT (1I3)
53     C 200 FORMAT (' ERROR IN INPUT DATA CARDS ----,8A10)
54     C
55     C END
56     C
57     C
58     C

```

```

1      SUBROUTINE WTSURFB (AMC,BMC,MMAX)
      C
      C      DETERMINE THE SURFACE AREA OF A BODY OF
      C      REVOLUTION GIVEN BY CHEBYCHEV COEFFICIENTS
      C
      C      DIMENSION AMC(10),BMC(10)
      C
      C      COMMON/OUT/HS,HB,XLS,XLB,TSMAX,AX,PI,G,RHO,GNU,WETS,WETB,WTSURF,
      C      VMFPS,DELFCF,TITLE(B),SPACE
      C
      C      RAD=SQRT(AX/PI)
      C      CON=(2.0*RAD/XLB)**2
      C      DX=0.05
      C      DXSO=DX*DX
      C      R=0.0
      C      WETB=0.0
      C      DO 10 I=1,40
      C      RL=R
      C      X=DX*FLOAT(I)-1.0
      C      CALL EVAL (X,MMAX,R,AMC,BMC)
      C      IF (R.LT.0.0) R=0.0
      C      R=SQRT(R)
      C      RM=(PL+R)*0.5
      C      DR=R-RL
      C      WETB=WETB+RM*SQRT(DXSO*CON+DR**2)
      C
      C      10 CONTINUE
      C      WETB=PI*RAD*XLB*WETB
      C      RETURN
      C      END

```

```

1      C
2      SUBROUTINE WTSURFS (AMC,BMC,MMAX)
3
4      C      DETERMINE THE SURFACE AREA OF A STRUT WHOSE THICKNESS
5      C      DISTRIBUTION IS GIVEN BY CHEBYCHEV COEFFICIENTS
6
7      C
8      DIMENSION AMC(10),BMC(10)
9      COMMON/OUT/HS,HB,XLS,XLB,TSMAX,AX,PI,G,RHO,GNU,WETS,WETB,WTSURF,
10     VMFPS,DELCP,TITLE(8),SPACE
11
12     C
13     CON=(TSMAX/XLS)**2
14     DX=0.05
15     DXSQ=DX*DX
16     WETS=0.0
17     T=0.0
18     DO 10 I=1,40
19         TL=T
20         X=DX*FLOAT(I)-1.0
21         CALL EVAL (X,MMAX,T,AMC,BMC)
22         DT=T-TL
23         WETS=WETS+SQRT(DXSQ+CON*DT**2)
24
25     10 CONTINUE
26     WETS = WETS - SPACE * 2.0
27     WETS=XLS*HS*WETS
28     RETURN
29     END

```

```

1      SUBROUTINE READ
2      READ
3      READ
4      READ
5      READ
6      READ
7      READ
8      READ
9      READ
10     READ
11     READ
12     READ
13     READ
14     READ
15     READ
16     READ
17     READ
18     READ
19     READ
20     READ
21     READ
22     READ
23     READ
24     READ
25     READ
26     READ
27     READ
28     READ
29     READ
30     READ
31     READ
32     READ
33     READ
34     READ
35     READ
36     READ
37     READ
38     READ
39     READ
40     READ
41     READ
42     READ
43     READ
44     READ
45     READ
46     READ
47     READ
48     READ
49     READ
50     READ
51     READ
52     READ
53     READ

10     COMMON/OFFSET/X(100),Y(100),D(100),THETA(100),N,NV(10),NNV
11     COMMON/NAME/INCHAR(8),NAME(7)
12     COMMON/NORM/YMAX,XDIVER,YDIVER,XADDER
13     LOGICAL REPEAT
14     YMAX = 0.
15     DO 47 I = 2,8
16     NAME(I-1) = INCHAR(I)
17     REPEAT = .FALSE.
18     NNV = 1
19     DO 40 I = 1,100
20     N = 1
21     N = NUMBER OF INPUT POINTS
22     IF (REPEAT) GO TO 339
23     READ(5,12) X(I),Y(I), IEND
24     IF (I.EQ.1) GO TO 39
25     IF (I.GT.2) GO TO 38
26     SIGN = X(I) - XLAST
27     IF (SIGN.EQ.0.) GO TO 45
28     GO TO 39
29     IF ((X(I)-XLAST)-SIGN .LE. 0.) GO TO 45
30     XLAST = X(I)
31     IF (Y(I).GT.YMAX) YMAX = Y(I)
32     IF (IEND.EQ.5H END) GO TO 42
33     IF (IEND.NE.5HBREAK) GO TO 40
34     REPEAT = .TRUE.
35     NV(NNV) = I
36     NNV = NNV + 1
37     GO TO 40
38     X(I) = X(I-1)
39     Y(I) = Y(I-1)
40     REPEAT = .FALSE.
41     CONTINUE
42     WRITE(6,18) NAME
43     STOP
44     42 CONTINUE
45     NV(NNV) = N
46     GO TO 8
47     45 CONTINUE
48     PRINT 46, X(I),Y(I) ,NAME
49     STOP
50     8 CONTINUE
51     RETURN
52     FORMAT (2F10.0, 5X, A5)
53     FORMAT (* TOO MANY INPUT CARDS FOR *,7A10)
54     FORMAT (* CARD OUT OF ORDER *,2E16.8/20X,=FOR-- *,7A10)
55     END

```

```

1      SUBROUTINE SCALE (ISBODY)
2      C
3      C
4      C
5      C
6      C
7      C
8      C
9      C
10     C
11     C
12     C
13     C
14     C
15     C
16     C
17     C
18     C
19     C
20     C
21     C
22     C
23     C
24     C
25     C
26     C
27     C
28     C
29     C
30     C
31     C
32     C
33     C
34     C
35     C
36     C
37     C
38     C
39     C

      COMPUTE SCALING FACTORS AND SCALE THE INPUT FOR LENGTH
      BETWEEN -1.0 AND +1.0 AND MAXIMUM BEAM OR SECTIONAL
      AREA OF 1.0

      COMMON/OFFSET/X(100),Y(100),D(100),THETA(100),N,NV(10),NNV
      COMMON/NAME/INCHAR(8),NAME(7)
      COMMON/NORM/YMAX,XDIVER,YDIVER,XADDER
      LOGICAL ISBODY

      YDIVER = YMAX
      XDIVER = (X(N)-X(1))*0.5
      XADDER = -1. - X(1) / XDIVER
      IF (ISBODY) WRITE (6,17)
      IF (.NOT. ISBODY) WRITE (6,18)
      WRITE(6,19) NAME
      WRITE(6,25)
      WRITE(6,26) (X(I),Y(I),I=1,N)

      IF *YDIVER* IS NOT ZERO, THE Y INPUTS WILL BE DIVIDED BY *YDIVER*
      .... SAME FOR *XDIVER* .....
      THEN *XADDER* IS ADDED TO X INPUTS
      THEN IF *ISBODY*, THE Y INPUTS WILL BE SQUARED

      WRITE(6,206) XDIVER,XADDER,YDIVER
      IF(ISBODY) WRITE(6,207)
      RETURN
17  FORMAT ('H1,* INPUT OFFSET DATA OF BODY *')
18  FORMAT ('H1,* INPUT OFFSET DATA OF STRUT *')
19  FORMAT ('H0.7A10,/')
25  FORMAT ('7X,*X-VALUES=,10X,*Y-VALUES=,10X,*BEFORE SCALING=//')
26  FORMAT ('2I6X,F10.5')
35  FORMAT ('=0 SCALING=--*/' * X VALUES ARE DIVIDED BY *.E16.8,
1      *, THEN ADD *.E16.8 / * Y VALUES ARE DIVIDED BY *.E16.8)
207  FORMAT ('15X,*AND SQUARED*')
      END

```

```

1      SUBROUTINE COMPUT (AMC,BMC,MMAX,ISBODY)
      C
      C      COMPUTE THE COEFFICIENTS OF THE CHEBYCHEV POLYNOMIALS
      C
5      DIMENSION AMC(10),BMC(10)
      COMMON/OFFSET/X(100),Y(100),D(100),THETA(100),N,NV(10),NNV
      COMMON/NAME/INCHAR(8),NAME(7)
      COMMON/NORM/YMAX,XDIVER,YDIVER,XADDER
      LOGICAL ISBODY
10     C
      DO 100 I=1,N
        Y(I) = Y(I) / YDIVER
        IF (ISBODY) Y(I) = Y(I)**2
        XWAS = X(I)
        X(I) = X(I)/XDIVER + XADDER
        IF (ABS(X(I)) .LE. 1.) GO TO 90
        IF (ABS(X(I)) .GT. 1.01) GO TO 90
        WRITE(6,201) X(I),XWAS
        X(I) = AINT(X(I))
        THETA(I) = ASIN(X(I))
90      CONTINUE
100    CONTINUE
      IF (ISBODY) WRITE (6,13)
      IF (.NOT.ISBODY) WRITE (6,14)
      WRITE(6,19) NAME
      WRITE(6,15)
      WRITE(6,16) (X(I),Y(I),THETA(I),I=1,N)
      C
      C      COMPUTE FOLLOWS
      C
      CALL CHEV (AMC,BMC,MMAX,ISBODY)
      C
      RETURN
13    FORMAT(1H1,*, NORMALIZED OFFSET DATA FOR BODY *)
14    FORMAT(1H1,*, NORMALIZED OFFSET DATA FOR STRUT *)
15    FORMAT (7X,*X-VALUES*,10X,*Y-VALUES*,10X,* THETA *//)
16    FORMAT (3(6X,F10.5))
19    FORMAT(1H0,7A10,/)
201   FORMAT (*TRUNCATED X= *,2E16.8)
      END

```

13.06.42

04/07/81

FTN 4.6+460

SUBROUTINE SURFACE 74/74 OPT=0 ROUND=*/ TRACE

```

1      SUBROUTINE SURFACE (AMC,BMC,MMAX,ISBODY)
      C
      C      FIND THE WETTED SURFACE AREA OF STRUT OR BODY. ALSO THE
      C      WATER PLANE AREA OF STRUT AND DISPLACED VOLUME OF BODY
      C
5     DIMENSION AMC(10),BMC(10)
      COMMON/OUT/HS,HB,XLS,XLB,TSMAX,AX,PI,G,RHO,GNU,WETS,WETB,WTSURF,
1     VMFPS,DELCF,TITLE(8),SPACE
      COMMON/NAME/INCHAR(8),NAME(7)
      LOGICAL ISBODY
      DATA PI / 3.1415926535897 /
      C
      C      IF (ISBODY) GO TO 100
      C
      C      COMPUTE WETTED SURFACE AREA OF STRUT
      C
      C      CALL WTSURFS (AMC,BMC,MMAX)
      C
      C      COMPUTE WATER PLANE AREA OF STRUT
      C
20     WPA = XLS * TSMAX * AMC(1) * PI / 4.0
      WRITE (6,110) WETS,XLS,TSMAX,WPA
      GOTO 999
      C
      C      100 CONTINUE
      C
      C      COMPUTE WETTED SURFACE AREA OF BODY
      C
      C      CALL WTSURFB (AMC,BMC,MMAX)
      C
      C      WETB = WETB - WPA
      C
      C      COMPUTE DISPLACED VOLUME
      C
      C      DISPVOL=XLB*AX*PI/4.0*MC(1)
      WRITE (6,120) WETB,XLB,AX,DISPVOL
      RETURN
999 CONTINUE
      C
110 FORMAT (1H0,'WETTED SURFACE AREA OF STRUT IS ',E16.8,
1     ' WHERE STRUT LENGTH IS ',E16.8,
2     ' AND MAXIMUM THICKNESS IS ',E16.8,
3     '/.20X,'WATER PLANE AREA IS ',E16.8)
120 FORMAT(1H0,'WETTED SURFACE AREA OF BODY IS ',E16.8,
1     ' WHERE BODY LENGTH IS ',E16.8,
2     ' AND MAXIMUM AREA IS ',E16.8/
3     ' 20X,' BODY VOLUME IS ',E16.8)
      END

```


04/07/81 13.06.42

FTN 4.6+460

74/74 OPT=0 ROUND=0/ TRACE

SUBROUTINE CHEV

```

204  FORMAT (//10X," BMC ARRAY  "/ (8E16.8))
650  FORMAT ("AREA COEFFICIENT = AREA/(LSTRUT-TMAX) = ",E16.8)
850  FORMAT ("OVOLUME COEFFICIENT = VOLUME/(LBODY*AMAX) = ",E16.8)
      END

```

```

      CHEV
      CHEV
      CHEV

```

```

59
60
61
62

```

60

13.06.42

04/07.81

FTN 4.6+460

74/74 OPT=0 ROUND=0/ TRACE

SUBROUTINE CHECK

```

1      SUBROUTINE CHECK (AMC,BMC,MMAX,ISBODY)
      C
      C      THIS SUBROUTINE CHECKS THE DIFFERENCE BETWEEN APPROXIMATED
      C      OFFSETS AND ORIGINAL OFFSETS
      C
5     DIMENSION AMC(10),BMC(10)
      COMMON/OFFSET/X(100),Y(100),D(100),THETA(100),N,NV(10),NNV
      COMMON/NAME/INCHAR(8),NAME(7)
      LOGICAL ISBODY
      C
10     N = NUMBER OF ORIGINAL OFFSETS GIVEN
      C
      C      MMAX = MAX ORDER OF CHEBYCHEV EXPANSION USED
      C
      C      IF (ISBODY) WRITE (6,121)
      C      IF (.NOT.ISBODY) WRITE (6,122)
15     123 WRITE(6,126) MMAX,NAME
      WRITE(6,200)
      DO 100 I = 1,N
      C
      C      CALL EVAL(X(I),MMAX,YX,AMC,BMC)
      C
20     IF (I.EQ.N) GO TO 10
      XX = (X(I) + X(I+1)) * .5
      C
      C      CALL EVAL (XX,MMAX,YXX,AMC,BMC)
      C
25     WRITE(6,201) (Y(I),X(I),YX,XX,YXX)
      GO TO 100
100    CONTINUE
      RETURN
30     121 FORMAT(1H1," CHECK OFFSET OF BODY USING CHEBYCHEV APPROXIMATION")
      122 FORMAT(1H1," CHECK OFFSET OF STRUT USING CHEBYCHEV APPROXIMATION")
      126 FORMAT (1H0,"CHECK OUT PUT,MMAX=,15=,FOR=.7A10./)
200    FORMAT ('0=,5X,-REAL Y=,12X,-X=,12X,-APPROX Y=,27X,*XX=,10X,
      1      *APPROX Y(XX)=,/)
201    FORMAT (1X,3E16.8,16X,2E16.8)
      END
35

```


13.06.42

04/07/81

FTN 4.6+460

TRACE

OPT=0

ROUND=0

74,74

SUBROUTINE SPLINE

```

1      SUBROUTINE SPLINE (X,Y,D,N,BC,KODE,KKODE,ISBODY)
2      THIS SUBROUTINE FITS SMOOTH SPLINE SEGMENTS THROUGH A GIVEN SET
3      OF DISCRETE DATA POINTS
4      DIMENSION X(N),Y(N),D(N)
5      DIMENSION A(100),B(100),C(100),BC(8)
6      LOGICAL ISBODY
7
8      X ARRAY CONTAINS ABSCISSAS FOR INPUT DATA
9      Y ARRAY CONTAINS ORDINATES FOR INPUT DATA
10     ARRAYS A,B,C ARE MAINLY SUB DIAG., DIAGONAL, AND SUPER DIAG.
11     D ARRAY IS THE RIGHT HAND SIDE OF MATRIX EQUATION
12     FINAL ANSWERS ARE PLACED IN D
13
14     SOLUTION TECHNIQUE IS GAUSSIAN ELIMINATION
15     ARRAY BC CONTAINS SPECIFIED BOUNDARY CONDITIONS
16
17     KODE = 0 USER IS SPECIFYING BOUNDARY CONDITIONS
18     KODE = 1 USER IS TAKING AN EXTRAPOLATION OF SECOND
19     DERIVATIVES AS THE BOUNDARY CONDITION
20     KKODE = 0 NO PRINT OUT FROM SPLINE ROUTINE
21     KKODE = 1 SPLINE PRINTS ABSCISSAS,ORDINATES AND SECOND DERIVATIVE
22
23     IF (N.GT.2) GO TO 25
24     D(1) = 0.
25     D(2) = 0.
26     NPRINT = 2
27     GO TO 20
28
29     IF (N.GT.3) GO TO 50
30     YDD = 2.*((X(3)-X(2))*Y(1)+(X(2)-X(1))*Y(3)-(X(3)-X(1))*Y(2))
31     / ((X(3)-X(2))*(X(2)-X(1))-(X(3)-X(1)))
32     D(1) = YDD
33     D(2) = YDD
34     D(3) = YDD
35     NPRINT = 3
36     GO TO 20
37
38     NPRINT = 0
39     II = 1
40     NL1 = N-1
41     NL2 = N-2
42     DO 12 I = 1,N
43     A(I) = 0.
44     B(I) = 0.
45     C(I) = 0.
46     D(I) = 0.
47
48     SET UP MATRICES(A TRIAGONAL STRUCTURE)
49
50     A(1) = (X(3)-X(2))/(X(3)-X(1))
51     C(1) = 2.0
52     B(1) = 1.0 - A(1)
53     D(1) = 6.0*((Y(3)-Y(2))/(X(3)-X(2))-(Y(2)-Y(1))/(
54     (X(2)-X(1)))/(X(3)-X(1))
55     H = X(3) - X(2)
56     DO 19 J=3,NL1
57     HP = X(J+1) - X(J)
58     C(J) = HP/(H+HP)

```

60	19	B(J) = 2.0	SPLINE	59
	C	A(J) = 1.0 - C(J)	SPLINE	60
	C	D(J) = 6.0*((Y(J+1)-Y(J))/HP-(Y(J)-Y(J-1))/H)/(HP+H)	SPLINE	61
	C	H = HP	SPLINE	62
	C	CHECK BOUNDARY CONDITIONS	SPLINE	63
65	C	IF (KODE .EQ. 1) GO TO 6	SPLINE	64
		IF(BC(2) .EQ. 0.0 .AND. BC(3) .EQ. 0.0) GO TO 1	SPLINE	65
		GO TO 2	SPLINE	66
	1	WRITE(6,300) BC(2),BC(3)	SPLINE	67
		GO TO 6	SPLINE	68
70	2	C(2) = 0.0	SPLINE	69
		A(2) = BC(3)-BC(2)*(X(2)-X(1))/3.0	SPLINE	70
		B(2) = -BC(2) + (X(2)-X(1))/6.0	SPLINE	71
		D(2) = BC(4)-BC(1)*Y(1)-BC(2)*(Y(2)-Y(1))/(X(2)-X(1))	SPLINE	72
75	3	IF(BC(6) .EQ. 0.0 .AND. BC(7) .EQ. 0.0) GO TO 4	SPLINE	73
		GO TO 5	SPLINE	74
	4	WRITE(6,301) BC(6),BC(7)	SPLINE	75
		GO TO 7	SPLINE	76
	5	C(N) = 0.0	SPLINE	77
		A(N) = BC(6)*(X(N)-X(N-1))/6.0	SPLINE	78
80		B(N) = BC(6)*(X(N)-X(N-1))/3.0+BC(7)	SPLINE	79
		D(N) = BC(8)-BC(5)*Y(N)-BC(6)*(Y(N)-Y(N-1))/(X(N)-X(N-1))	SPLINE	80
		GO TO 10	SPLINE	81
	6	C(2) = (X(2)-X(1))/(X(3)-X(2))	SPLINE	82
		A(2) = 1.0	SPLINE	83
85		B(2) = -1.0-C(2)	SPLINE	84
		D(2) = 0.0	SPLINE	85
		IF (KODE .EQ. 0) GO TO 3	SPLINE	86
90	7	C(N) = (X(N)-X(N-1))/(X(N-1)-X(N-2))	SPLINE	87
		A(N) = 1.0	SPLINE	88
		B(N) = 1.0	SPLINE	89
		D(N) = 0.0	SPLINE	90
	C	SOLVE EQUATIONS	SPLINE	91
	C		SPLINE	92
	C		SPLINE	93
95	10	C(2) = - A(2)*A(1)/B(1) + C(2)	SPLINE	94
		DO 13 I=1,NL2	SPLINE	95
		AUGH = ABS(B(1))	SPLINE	96
		IF (AUGH .LT. 1.0E-06) GO TO 9	SPLINE	97
		CONST = A(I+1)/B(I)	SPLINE	98
100		B(I+1) = B(I+1) - CONST*C(I)	SPLINE	99
		D(I+1) = D(I+1) - CONST*D(I)	SPLINE	100
		IF (I .NE. NL2) GO TO 13	SPLINE	101
		A(N) = A(N) - C(N)*C(I) / B(I)	SPLINE	102
		D(N) = D(N) - C(N)*D(I) / B(I)	SPLINE	103
105		GO TO 13	SPLINE	104
	9	II = I+1	SPLINE	105
		D(I) = D(I)/C(I)	SPLINE	106
		D(I+1) = D(I+1) - B(I+1)*D(I)	SPLINE	107
		B(I+1) = A(I+1)	SPLINE	108
		A(I+1) = 0.0	SPLINE	109
110		D(I+2) = D(I+2) - A(I+2)*D(I)	SPLINE	110
		A(I+2) = 0.0	SPLINE	111
		IF (I .NE. NL2) GO TO 13	SPLINE	112
		A(N) = C(N)	SPLINE	113
			SPLINE	114
			SPLINE	115

```

115      13      CONTINUE
      DET = B(N-1)*B(N) - C(N-1)*A(N)
      STORE = D(N)
      D(N) = (B(N-1)*D(N) - D(N-1)*A(N))/DET
      D(N-1) = (D(N-1)*B(N) - C(N-1)*STORE)/DET
      IP = 0
      DO 15 I = 2, NL2
      JI = N-I
      IF (JI .EQ. IP) GO TO 15
      IF (JI .EQ. II) GO TO 8
      D(JI) = (D(JI)-C(JI)*D(JI+1))/B(JI)
      GO TO 15
      8      IP = JI-1
      STORE = D(JI)
      D(JI) = D(JI-1)
      D(JI-1) = (STORE - C(JI-1)*D(JI+1))/B(JI-1)
      15      CONTINUE
      D(1) = (D(1) - A(1)*D(3) - C(1)*D(2))/B(1)
      C
      C
      C
      20      ALL DONE
      IF (KKODE .NE. 1) GO TO 1000
      IF (ISBODY) WRITE(6,100) (I,X(I),Y(I),D(I),I=1,N)
      IF (.NOT.ISBCDY) WRITE(6,101) (I,X(I),Y(I),D(I),I=1,N)
      IF (NPRINT.EQ.3) WRITE(6,901)
      IF (NPRINT.EQ.2) WRITE(6,902)
      1000      RETURN
      100      FORMAT (1H1,
      1          3CX,"SPLINE APPROXIMATION OF BODY",//27X,
      2          *ABSCISSA
      3          (12X,110.3E16.8))
      101      FORMAT (1H1,30X,"SPLINE APPROXIMATION OF STRUT",//27X,
      1          *ABSCISSA
      2          (12X,110.3E16.8))
      300      FORMAT(//,"THE BOUNDARY CONDITIONS SPECIFIED AT THE FIRST POINT
      1      ARE NOT SUFFICIENT", BC(2) = ".E16.8/" BC(3)=".E16.8/" THE
      2      PROGRAM WILL CONTINUE WITH THE SECOND DERIVATIVE AT THE FIRST
      3      POINT EQUAL TO A LINEAR COMBINATION OF THE SECOND AND THIRD"//)
      301      FORMAT(//,"THE BOUNDARY CONDITIONS SPECIFIED AT THE LAST POINT
      1      ARE NOT SUFFICIENT", BC(6)=".E16.8/" BC(7)=".E16.8/"
      2      " THE PROGRAM WILL CONTINUE WITH THE SECOND DERIVATIVE AT THE LAST
      3      POINT EQUAL TO A LINEAR COMBINATION OF THE THIRD AND SECOND
      4      LAST")
      901      FORMAT (*0THREE POINT SPLINE, INTERPOLATION IS QUADRATIC*)
      902      FORMAT (*0TWO POINT SPLINE, INTERPOLATION IS LINEAR*)
      END

```

```

1      REAL FUNCTION SUMSPL(X,Y,S,N,AK,KODE)
2
3      THIS PROGRAM USES THE SPLINE COEFFICIENTS AND INTEGRATES THE
4      TRIGONOMETRIC INTEGRALS
5      OF THE FORM  $\int \sin(AK \cdot X)$  OR  $\int \cos(AK \cdot X)$ 
6      IF KODE = +1 PROGRAM PERFORMS SINE INTEGRATION
7      IF KODE = -1 PROGRAM PERFORMS COSINE INTEGRATION
8      INPUT DATA ARE GIVEN AS (Y(I) = F(X(I)), I=1,N)
9      N IS THE TOTAL NUMBER OF INPUT POINTS (ABSCISSAS)
10     S-ARRAY STORES THE SECOND DERIVATIVES OF THE SPLINE FUNCTIONS
11     THE RANGE OF INTEGRATION IS FROM X(1) TO X(N)
12
13     DIMENSION X(100),Y(100),S(100)
14     NL = N-1
15     SUMSPL = 0.0
16     D2 = AK*AK
17     D3 = AK*D2
18     D4 = AK*D3
19     DO 1 I=1,NL
20         H = X(I+1) - X(I)
21         IF (H) 1,1,9
22         ZAA = (S(I+1)-S(I))/(E.*H)
23         ZBB = S(I)/2.0
24         ZCC = (Y(I+1)-Y(I))/H - H*(S(I+1)+2.0*S(I))/6.
25         E = SIN(AK*H)
26         F = COS(AK*H)
27         AA1 = (3.*D2*H*H-6.)*F/D4+(D2*H*H*H-6.*H)*E/D3+6./D4
28         AA2 = 3.*H*H*E/D2-6.*E/D4-H*H*H*F/AK+6.*H*F/D3
29         BB1 = 2.*H*F/D2-2.*E/D3+H*H*E/AK
30         BB2 = 2.*H*E/D2+2.*F/D3-H*H*F/AK-2./D3
31         CC1 = F/D2 + H*E/AK -1.0/D2
32         CC2 = E/D2 -H*F/AK
33         DD1 = E/AK
34         DD2 = -F/AK+1.0/AK
35         UU = SIN(AK*X(I))
36         VV = COS(AK*X(I))
37         IF (KODE)2,3,4
38         SUMSPL = SUMSPL + VV*(AA1*ZAA+BB1*ZBB+CC1*ZCC+DD1*Y(I))
39         1 - UU*(AA2*ZAA+BB2*ZBB+CC2*ZCC+DD2*Y(I))
40         GO TO 1
41         SUMSPL = SUMSPL + UU*(AA1*ZAA+BB1*ZBB+CC1*ZCC+DD1*Y(I)) +
42         1 VV*(AA2*ZAA + BB2*ZBB + CC2*ZCC+DD2*Y(I))
43         1 CONTINUE
44         GO TO 5
45         WRITE(6,100)
46         SUMSPL = 1.0
47         RETURN
48     100 FORMAT (1H1,"KODE = 0"/)" SUMSPL IS SET EQUAL TO 1."
49     END
50

```

SUMSPL 2
SUMSPL 3
SUMSPL 4
SUMSPL 5
SUMSPL 6
SUMSPL 7
SUMSPL 8
SUMSPL 9
SUMSPL 10
SUMSPL 11
SUMSPL 12
SUMSPL 13
SUMSPL 14
SUMSPL 15
SUMSPL 16
SUMSPL 17
SUMSPL 18
SUMSPL 19
SUMSPL 20
SUMSPL 21
SUMSPL 22
SUMSPL 23
SUMSPL 24
SUMSPL 25
SUMSPL 26
SUMSPL 27
SUMSPL 28
SUMSPL 29
SUMSPL 30
SUMSPL 31
SUMSPL 32
SUMSPL 33
SUMSPL 34
SUMSPL 35
SUMSPL 36
SUMSPL 37
SUMSPL 38
SUMSPL 39
SUMSPL 40
SUMSPL 41
SUMSPL 42
SUMSPL 43
SUMSPL 44
SUMSPL 45
SUMSPL 46
SUMSPL 47
SUMSPL 48
SUMSPL 49
SUMSPL 50

```

1      C
2      C
3      C
4      C
5      SUBROUTINE SPLFIT(X,Y,S,N,T,YVAL,YD,YDD)
6      FIND THE DERIVATIVES (1ST AND 2ND) OF THE SPLINE FITTED DATA
7      DIMENSION X(100),Y(100),S(100)
8      I = 1
9      IF (T-X(I)) 6,6,4
10     I = I + 1
11     GO TO 3
12     I = I - 1
13     IF (I.EQ.0) I=1
14     H = X(I+1) - X(I)
15     HTX = X(I+1) - T
16     HTT = T - X(I)
17     YVAL = S(I) + (HTX**3)/(6.*H)+S(I+1)*HTT**3/(6.*H)
18     YD = -S(I)*HTX*HTX/(2.*H)+S(I+1)*HTT*HTT/(2.*H)
19     YDD = S(I) *HTX/H+S(I+1)*HTT/H
20     RETURN
21     END
22
SPLFIT 2
SPLFIT 3
SPLFIT 4
SPLFIT 5
SPLFIT 6
SPLFIT 7
SPLFIT 8
SPLFIT 9
SPLFIT 10
SPLFIT 11
SPLFIT 12
SPLFIT 13
SPLFIT 14
SPLFIT 15
SPLFIT 16
SPLFIT 17
SPLFIT 18
SPLFIT 19
SPLFIT 20
SPLFIT 21
SPLFIT 22

```

04/07/81 13.06.42

FTN 4.6+460

FUNCTION FORMDR 74/74 OPT=0 ROUND=0/ TRACE

2 FORMDR
3 FORMDR
4 FORMDR
5 FORMDR
6 FORMDR
7 FORMDR
8 FORMDR
9 FORMDR
10 FORMDR
11 FORMDR
12 FORMDR
13 FORMDR
14 FORMDR

```

1      C
      C
      C
      FUNCTION FORMDR(VL)
      EVALUATE FORM DRAG COEFFICIENT
      DIMENSION X(9), Y(9)
      DATA X / 1.325, 1.38, 1.43, 1.48, 1.55, 1.68, 1.9, 2.2, 2.5 /
      DATA Y / .51, .71, .84, .85, .83, .75, .65, .57, .50 /
      IF (VL-GE.1.325.AND.VL.LT.2.5) GO TO 4
      FORMDR=0.5
      GO TO 5
      4 FORMDR = YINTP (VL,X,Y,9)
      5 RETURN
      END
10

```

```

1      FUNCTION YINTP (XA,X,Y,N)
      C
      C
      C      INTERPOLATING FROM A SET OF DATA (X,Y)
5      DIMENSION X(1), Y(1)
      DO 10 I=1,N
      IF (X(I)-XA) 10,10,2
      2 IN=I-2
      IF(IN) 4,4,6
      4 IN=1
      GO TO 12
      6 NN=N-3
      IF (IN-NN) 12,12,8
      8 IN = NN
      GO TO 12
      10 CONTINUE
      12 IO=IN+3
      YINTP=0.
      DO 20 I=IN,IO
      PROD=Y(I)
      DO 16 J=IN,IO
      IF (I-J) 15,16,15
      15 PROD = PROD*(XA-X(J))/(X(I)-X(J))
      16 CONTINUE
      20 YINTP=YINTP+PROD
      RETURN
      END

```

04/07/81 13.06.42

FTN 4.6+460

74/74 OPT=0 ROUND=*/ TRACE

FUNCTION C-ITTC

CFITTC 2
CFITTC 3
CFITTC 4
CFITTC 5
CFITTC 6
CFITTC 7
CFITTC 8
CFITTC 9

FUNCTION CFITTC (RN)

C

C

C

THIS FUNCTION DETERMINES THE ITTC CORRELATION ALLOWANCE

DEN = ALOG10 (RN) - 2.0

CFITTC = 0.075 / DEN**2

RETURN

END

1

5

PROGRAM LISTING OF SYNTHESIS

```

1      PROGRAM SWATH(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE8)
2      MAIN
3      PROGRAM FOR THE RESISTANCE AND PROPULSION OF SWATH SHIPS
4      PROGRAMMED BY A. M. REED JUNE 1975
5      POWERING MODULE BY W. G. DAY
6      APPEN AGE DRAG MODULE BY J. W. GRANT
7      REVISED APRIL 1978 BY E. C. CHAI
8
9      COMMON / COEFS / ASM(3), BSM(3), ABM(3), BBM(3), MMAX
10     COMMON / INPUT / XLS, HS, TSMAX, CWP, CLCF, CIYY, XLB, HB, AX, CP,
11     1 CLCB, SEPDIS, CSTRUT, CSTR12, FDIS, SPAN, CHORD, TFINS, NLOC
12
13     COMMON / OMEGA / VMFPS, GAMAS, GAMASB, GOSQ, HSOLS, HBOLB, WETS,
14     1 WETB, WETFIN, WTSURF, SEP, PHIS, PHIB, RATIOL, CFS, CFB, APPDRG
15
16     COMMON / PHYSICO / RHO, G, PI, DELCF
17
18     COMMON/PSI/NPTSZ,PTSAF,EXP,NALMAX,NAL,TAIL,ALFA
19     1,ALSMAX,NSTEPS
20
21     DIMENSION TITLE(8), VOFF(24), EHP(24), SHP(24)
22
23     DEFINE THE PSI CONSTANTS
24
25     READ AND CHECK INPUT
26
27     CALL RIN(TITLE)
28
29     CALCULATE THE OMEGA AND BODY CONSTANTS
30
31     HSOLS = HS/XLS
32     HBOLB = HB/XLB
33     RATIOL = XLB/XLS
34     B = CSTRUT/XLS
35     D = 0.0
36     IF ( NLOC .NE. 0 ) D = CSTR2/XLS
37     SFINS = SPAN*CHORD
38     TBFIN = 0.01*TFINS
39     IF(XLB .GT. 30.0 .AND. TBFIN .LT. 0.041666) TBFIN = 0.041666
40     IF(XLB .LE. 30.0 .AND. TBFIN .LT. 0.005208) TBFIN = 0.005208
41     SBFIN = TBFIN*SPAN
42     SORTL = SORT(XLS)
43     BDIA = SQRT(AX/PI)
44
45     DETERMINE THE CHEBYSHEV COEFFICIENTS FOR THE SHIP
46     CALCULATE THE THE WETTED SURFACE OF THE BODY AND STRUT
47     PLOT THE STRUT PROFILE AND SECTIONAL AREA CURVE
48
49     CALL CHEB(TITLE)
50
51     DETERMINE THE SPEEDS AND PROCEED THROUGH THE RESISTANCE LOOP
52
53     READ(5,100) VKMIN, VKMAX, NC
54     IF ( NC .GT. 20 ) GO TO 90
55

```


PROGRAM	SWATH	74/74	OPT=0	ROUND=*	TRACE	FTN	4.6+460	04/07/81	13.10.59	PAGE	3
115											
	100	FORMAT(2F10.5,I5)									
	101	FORMAT(1H1,BA10//T5,"V(KNOTS)",T16,"V(FT/SEC)",T27,"SLR(STRUT)",						SYNTHS	116		
		1 T46,"EMP",T58,"SHP",T71,"PC")						SYNTHS	117		
	102	FORMAT(6(2X,F10.3))						SYNTHS	118		
	103	FORMAT("*****ERROR NC GREATER THAN 20, NC = ",I3)						SYNTHS	119		
		END						SYNTHS	120		
120								SYNTHS	121		

04/07/81 13.10.59

FTN 4.6+460

DPT=0 ROUND=0/ TRACE

74/74

BLOCK DATA BLKDAT.

```

1      BLOCK DATA
      C
      COMMON / COEFS / ASM(3), BSM(3), ABM(3), BBM(3), MMAX
      C
      COMMON / PHYSCO / RHO,GNU,G,PI,DELFCF
      C
      COMMON/PSI/NPTSZ,PTSFAF,EXP,NALMAX,NAL,TAIL,ALFA
      C
      COMMON/PLOT/NFIRST,NLAST,NPOINT,XMAX,XMIN,NSCL1,NCHAR,NSCALE(4),
      C      PCHAR(2)
      C
      COMMON/XRPLOTO/I,J,K,L,NHL,NSBH,NVL,NSBV,HCHAR,VCHAR,ISX,ISY,V,H
      C
      LOGICAL NSCL1
      C
      MAXIMUM ORDER OF CHEBYCHEV APPROXIMATION
      C
      DATA MMAX / 3 /
      C
      DEFINE PHYSICAL CONSTANTS
      C
      DATA RHO / 1.9908 /
      DATA GNU / 1.297E-5 /
      DATA G / 32.174 /
      DATA PI / 3.1415926535898 /
      DATA DELFCF / 0.0005 /
      C
      DEFINE CONSTANTS FOR NUMERICAL INTEGRATION
      C
      DATA NPTSZ / 31 /
      DATA PTSFAF / 10.0 /
      DATA EXPN / 7.0 /
      DATA NALMAX / 300 /
      DATA ALSMAX / 26.0 /
      C
      DEFINE DATA FOR PLOT ROUTINE
      C
      DATA NHL, NSBH, NVL, NSBV, HCHAR, VCHAR, PCHAR(1), PCHAR(2) /
      C      6., 10., 11., 10., 1H., 1H., 1H., 1H. /
      C
      DATA NFIRST, NLAST, NPOINT / 1., 101., 1 /
      C
      DATA XMAX, XMIN, NSCL1 / 1.0, -1.0, .TRUE. /
      C
      DATA NCHAR / 0 /
      C
      DATA NSCALE / 0., 3., 0., 3 /
      C
      END

```

```

1      SUBROUTINE CHEB(TITLE)
2      COMMON / COEFS , AS ( ), BSM(3), ABM(3), BBM(3), MMAX
3      COMMON / INPUT / XLS, MS, TSMAX, CWP, CLCF, CIYY, XLB, HB, AX, CP,
4      1 CLCB, SEPDIS, CSTRUT, CSTRT2, PDIA, SPAN, CHORD, TFINS, NLOC
5      COMMON / OMEGA / VMFPS, GAMAOS, GAMAOSB, GOSQ, HSOLS, HBOLB, WETS,
6      1 WETB, WETFIN, WTSURF, SEP, PHIS, PHIB, RATIOL, CFS, CFB, APPDRG
7      COMMON / PHYSCO / RHO,GNU,G.PI,DELCF
8      DIMENSION TITLE(8)
9      DETERMINE THE CHEBYSHEV COEFFICIENTS
10     ASM(1) = 4.0*CWP/PI
11     ASM(2) = (1.0 - 16.0*CIYY)*ASM(1)
12     ASM(3) = 1.0 - ASM(1) - ASM(2)
13     BSM(1) = 4.0*CLCF*ASM(1)
14     BSM(2) = 0.0
15     BSM(3) = 0.0
16     ABM(1) = 4.0*CP/PI
17     ABM(2) = 1.0 - ABM(1)
18     ABM(3) = 0.0
19     BBM(1) = 4.0*CLCB*ABM(1)
20     BBM(2) = 0.0
21     BBM(3) = 0.0
22     DETERMINE THE WETTED SURFACE
23     CALL WSURFB(WETB)
24     CALL WSURFS(WETS)
25     WETB = WETB - XLS*TSMAX*CWP
26     IF ( NLOC .NE. 0 ) WETB = WETB - XLS*TSMAX*CWP
27     WETFIN = 2.0*CHORD*SPAN
28     WTSURF = WETB + WETS + WETFIN
29     IF ( NLOC .NE. 0 ) WTSURF = WTSURF + WETS
30     PRINT THE CHEBYSHEV COEFFICIENTS AND THE WETTED SURFACE
31     WRITE(6,340) (TITLE(I),I = 1, 8 )
32     WRITE(6,330) (ASM(M),M=1,MMAX)
33     WRITE(6,335) (BSM(M),M=1,MMAX)
34     WRITE(6,345) (ABM(K),K=1,MMAX)
35     WRITE(6,350) (BBM(K),K=1,MMAX)
36     WRITE(6,357) WETS,WETB, WETFIN, WTSURF
37     PLOT THE STRUT WATER LINE AND THE BODY AREA DISTRIBUTION
38     CALL PCHEB (ASN,BSM,ABM,BBM,MMAX,TITLE)
39     RETURN
40     FORMAT(1H0,33X,'STRUT CHEBYSHEV COEFFICIENTS ASM(M)=',(10X,SE16.8))
41     330  FORMAT(1H0,33X,'STRUT CHEBYSHEV COEFFICIENTS BSM(M)=',(10X,SE16.8))
42     335  FORMAT(1H0,33X,'STRUT CHEBYSHEV COEFFICIENTS ABM(K)=',(10X,SE16.8))
43     340  FORMAT(1H0,33X,'STRUT CHEBYSHEV COEFFICIENTS BBM(K)=',(10X,SE16.8))
44     345  FORMAT(1H0,33X,'STRUT CHEBYSHEV COEFFICIENTS WETS,WETB, WETFIN, WTSURF')
45     350  FORMAT(1H0,33X,'STRUT CHEBYSHEV COEFFICIENTS WETS,WETB, WETFIN, WTSURF')

```

13.10.59

04/07/81

FTN 4.6+460

OPT=0 ROUND=0/ TRACE

74/74

SUBROUTINE CHEB

CHEB 59
 CHEB 60
 CHEB 61
 CHEB 62
 CHEB 63

357 FORMAT(1H0,4X,* STRUT WETTED SURFACE IN SQUARE FEET =*,E14.7,
 1/ 4X,* BODY WETTED SURFACE IN SQUARE FEET =*,E14.7,
 2/ 4X,* FOIL WETTED SURFACE IN SQUARE FEET =*,E14.7,
 3/ 4X,* TOTAL WETTED SURFACE IN SQUARE FEET =*,E14.7)

END

04/07/81 13.10.59

FTN 4.6+460

74/74 OPT=0 ROUND=0/ TRACE

SUBROUTINE RIN

```

1      SUBROUTINE RIN(TITLE)
2      RIN
3      RIN
4      RIN
5      COMMON / INPUT / XLS, HS, TSMAX, CWP, CLCF, CIYY, XLB, HB, AX, CP,
6      1 CLCB, SEPDIS, CSTRUT, CSTR2, PDIA, SPAN, CHORD, TFINS, NLOC
7      RIN
8      RIN
9      RIN
10     DIMENSION TITLE(8)
11     RIN
12     RIN
13     RIN
14     RIN
15     RIN
16     RIN
17     RIN
18     RIN
19     RIN
20     RIN
21     RIN
22     RIN
23     RIN
24     RIN
25     RIN
26     RIN
27     RIN
28     RIN
29     RIN
30     RIN
31     RIN
32     RIN
33     RIN
34     RIN
35     RIN
36     RIN
37     RIN
38     RIN
39     RIN
40     RIN
41     RIN
42     RIN
43     RIN
44     RIN
45     RIN
46     RIN
47     RIN
48     RIN
49     RIN
50     RIN
51     RIN
52     RIN
53     RIN
54     RIN
55     RIN
56     RIN
57     RIN
58     RIN

    TITLE = LABEL PRINTED ON OUTPUT
    XLS = LENGTH OF STRUT (FT)
    HS = DRAFT OF STRUT (FT)
    TSMAX = MAXIMUM THICKNESS OF STRUT (FT)
    CWP = WATERPLANE AREA COEFFICIENT
    CWP = AWP / (TSMAX * LS)
    AWP = LS * (TSMAX/2) * (PI/2) * AS1
    CLCF = WATERPLANE MOMENT COEFFICIENT
    CLCF = MY(STRUT) / (AWP*LS)
    MY(STRUT) = (TSMAX/2) * (LS/2)**2 * (PI/2) * BS1
    CIYY = WATERPLANE INERTIA COEFFICIENT
    CIYY = IYY / (AWP*LS**2)
    IYY = (1/2) * (TSMAX/2) * (LS/2)**3 * (PI/2) * (A1 - A2)
    XLB = LENGTH OF BODY (FT)
    HB = DEPTH OF SUBMERGENCE TO BODY CENTERLINE (FT)
    AX = MAXIMUM BODY CROSS-SECTIONAL AREA (FT**2)
    CP = BODY PRISMATIC COEFFICIENT
    CP = DISPLACEMENT(BODY) / (AX * LB)
    DISPLACEMENT(BODY) = LB * AX * (PI/4) * AB1
    CLCB = BODY MOMENT COEFFICIENT
    CLCB = MY(BODY) / (AX*LB**2)
    MY(BODY) = AX * (LB/2)**2 * (PI/4) * BB1
    SEPDIS = SEPARATION DISTANCE OF HULL CENTERLINES (FT)
    CSTRUT = DISTANCE OF CL OF STRUT FROM CL OF BODY (FT)
    (+ FORWARD OF THE BODY CL. - AFT OF THE BODY CL)
    CSTR2 = DISTANCE OF CL OF SECOND STRUT ( IF PRESENT ) FROM
    CL OF BODY (FT)
    NLOC = SWITCH INDICATING THE PRESENCE OF A SECOND STRUT
    NLOC = 0 - ONE STRUT ONLY
    NLOC = 1 - TWO STRUTS
    PDIA = PROPELLER DIAMETER (FT)
    SPAN = SPAN OF A DEMI-FIN (FT)
    CHORD = CHORD OF A DEMI-FIN (FT)
    TFINS = MAXIMUM THICKNESS OF A DEMI-FIN (FT)

    READ INPUT
    READ (5,210) (TITLE(K),K=1,8)
    READ(5,215) XLS,HS,TSMAX
    READ (5,215) CWP,CLCF, CIYY
    READ(5,215) XLB,HB,AX
    READ(5,215) CP,CLCB
    READ(5,215) SEPDIS
    READ(5,216) CSTRUT, CSTR2, NLOC
    READ(5,215) PDIA
    READ(5,215) SPAN,CHORD,TFINS
    EC=0 INPUT
  
```

```

C
60      WRITE(6,310) (TITLE(K),K=1,8)
        WRITE(6,325) XLS,HS,TSMAX
        WRITE(6,330) CWP,CLCF,CIVY
        WRITE(6,340) XLB,HB,AX
        WRITE(6,335) CP,CLCB
        WRITE(6,355) SEPDIS
        IF (NLOC.NE.0) GO TO 20
        WRITE(6,356) CSTRUT
        GO TO 25
        CONTINUE
20      WRITE(6,357) CSTRUT, CSTRUT2
25      CONTINUE
        WRITE(6,358) PDIA
        WRITE(6,359) SPAN, CHORD, TFINS
        RETURN
C
75      C
        FORMATS
210      FORMAT(8A10)
215      FORMAT(8F10.5)
216      FORMAT(2F10.5,15)
310      FORMAT(1H1,/,/,/,3X,*, INPUT DATA FOR *.BA10//)
325      FORMAT(1H1,3X,35H STRUT GEOMETRIC CHARACTERISTICS ,/,/,5X,
1      BH XLS =,F12.6,9X,8H HS =,F12.6,9X,8H TSMAX =,F12.6)
330      FORMAT(1" WATERPLANE AREA COEFFICIENT = ",F12.6/
1" WATERPLANE MOMENT COEFFICIENT = ",F12.6/
2" WATERPLANE INERTIA COEFFICIENT = ",F12.6)
335      FORMAT(1" BODY PRISMATIC COEFFICIENT = ",F10.5/
1" BODY MOMENT COEFFICIENT = ",F10.5)
340      FORMAT(1H0,3X,35H BODY GEOMETRIC CHARACTERISTICS ,/,/,5X,
1      BH XLB =,F12.6,9X,8H HB =,F12.6,9X,8H AX =,F12.6)
355      FORMAT(1H0,5X,*, SEPARATION DISTANCE IN FEET =,8F8.3)
356      FORMAT(*0 STRUT LOCATION IN FEET =,8(2X,F10.4))
357      FORMAT(1" FORWARD STRUT LOCATION IN FEET = ",F10.4/
1" AFT STRUT LOCATION IN FEET = ",F10.4)
358      FORMAT(1" PROPELLER DIAMETER IN FEET = ",F10.4)
359      FORMAT(1" FIN SPAN IN FEET = ",F10.4/
1" FIN CHORD IN FEET = ",F10.5/
2" FIN THICKNESS IN FEET = ",F10.4)
        END
59      RIN
60      RIN
61      RIN
62      RIN
63      RIN
64      RIN
65      RIN
66      RIN
67      RIN
68      RIN
69      RIN
70      RIN
71      RIN
72      RIN
73      RIN
74      RIN
75      RIN
76      RIN
77      RIN
78      RIN
79      RIN
80      RIN
81      RIN
82      RIN
83      RIN
84      RIN
85      RIN
86      RIN
87      RIN
88      RIN
89      RIN
90      RIN
91      RIN
92      RIN
93      RIN
94      RIN
95      RIN
96      RIN
97      RIN
98      RIN
99      RIN

```

04/07/81 13.10.59

FTN 4.6*460

74/74 OPT=0 ROUND=0 TRACE

SUBROUTINE ROUT

```

1      SUBROUTINE ROUT(TITLE, EHP)
2      COMMON / INPUT / XLS, HS, TSMAX, CWP, CLCF, CIYV, XLB, HB, AX, CP,
3      1 CLCB, SEPDIS, CSTRUT, CSTRT2, PDIA, SPAN, CHORD, TFINS, NLOC
4      COMMON / OMEGA / VMFPS, GAMAOB, CAMAOB, COSQ, HSOLS, HBOLB, WETS,
5      1 WETB, WETFIN, WTSURF, SEP, PHIS, PHIB, RATIOL, CFS, CFB, APPDRG
6      COMMON / PHYSIO / RHO, GNU, G.PI, DELCF
7      DIMENSION TITLE(8)
8      SUM = A(TRANSPDSE)*T*A + B(TRANSPDSE)*W*B
9      CALL SUM(SUM1S, SUM1B, SUM1SB, SUM12S, SUM12B, SM12SB)
10     COMPUTE THE CONSTANTS NEEDED FOR OUTPUT
11     FROUDS = VMFPS/SORT(32.155*XLB)
12     VLS = FROUDS*SORT(G)/1.6878
13     FROUDB = VMFPS/SORT(32.155*XLB)
14     VLB = FROUDB*SORT(G)/1.6878
15     VMKNTS = VMFPS/1.6878
16     B2OLB = SEPDIS/XLB
17     CSOLB=CSTRUT/XLB
18     IF ( NLOC .NE. 0 ) CSOLB2 = CSTRT2/XLB
19     AAS = (TSMAX/HS)*(PI/2.0)*GAMAOB
20     AAB = 2.0*PI*(AX/XLB**2)*(GAMAOB/GAMAOB**2)
21     AASB = (2.0*PI)*AX/(HS*XLB)
22     AAWTSF = (0.5*RHO*WTSURF*VMFPS**2)/1000.
23     COMPUTE THE WAVE RESISTANCE COEFFICIENTS
24     CMS1 = AAS*SUM1S
25     R1S = CMS1*RHO*G*TSMAX*HS*XLB
26     CWS1 = R1S/AAWTSF
27     CMB1 = AAB*SUM1B
28     R1B = CMB1*RHO*G*AX*XLB
29     CWB1 = R1B/AAWTSF
30     CMSB1 = AASB*SUM1SB
31     R1SB = CMSB1*RHO*G*TSMAX*HS*XLB
32     CWSB1 = R1SB/AAWTSF
33     CMS12 = AAS*SUM12S
34     R12S = CMS12*RHO*G*TSMAX*HS*XLB
35     CWS12 = R12S/AAWTSF
36     CMB12 = AAB*SUM12B
37     R12B = CMB12*RHO*G*AX*XLB
38     CWB12 = R12B/AAWTSF
39     CMSB12 = AASB*SUM12SB
40     R12SB = CMSB12*RHO*G*TSMAX*HS*XLB
41     CWSB12 = R12SB/AAWTSF
42
43     ROUT
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58

```


04/07/81 13.10.59

FTN 4.6+460

74/74 OPT=0 ROUND=0/ TRACE

SUBROUTINE ROUT

```

115 325 FORMAT(1H ,3X,35H STRUT GEOMETRIC CHARACTERISTICS //,5X,
1 BH XLS =,F12.6,9X,8H HS =,F12.6,9X,8H TSMAX =,F12.6)
340 FORMAT(1H0,3X,35H BODY GEOMETRIC CHARACTERISTICS //,5X,
1 BH XLS =,F12.6,9X,8H HB =,F12.6,9X,8H AX =,F12.6)
354 FORMAT(//T25,"RESISTANCES IN POUNDS ARE FOR A DEMI-HULL")
356 FORMAT(//T25,"RESISTANCES IN POUNDS ARE FOR A DEMI-HULL")
1 120 1 5X,* 3/LB =,F7.3)
358 FORMAT(//T25,"STRUT CL FROM BODY CL IN FEET =,F8.3,
1 5X,*STRUT OFFSET/LB =,F7.3)
359 FORMAT(//T25,"FORWARD STRUT CL FROM BODY CL IN FEET =,F8.3,
1 5X,*FORWARD STRUT OFFSET/LB =,F7.3//
2 * AFT STRUT CL FROM BODY CL IN FEET =,F8.3,
3 5X,*AFT STRUT OFFSET/LB =,F7.3)
360 FORMAT(1H0,5X,* SHIP SPEED IS ,F8.3,* FPS, *F8.3,* KNOTS, *
1 6X,* STRUT V-L RATIO =,F8.3,6X,*BODY V-L RATIO =,F8.3)
410 FORMAT(//T25,"35H WAVE RESISTANCE CALCULATIONS FOR ,8A10)
415 FORMAT(1H ,4X,
1 F12.6,9X,8H HS/LS =,F12.6)
420 FORMAT(1H ,4X,
1 F12.6,9X,8H HB/LB =,F12.6)
430 FORMAT(//T30,32H STRUT WAVE DRAG IN POUNDS = ,E14.7,/30X,
2 32H R15/(RHO/2*WTSURF*V**2) = ,F10.3,4HE-03)
440 FORMAT(//T30,32H BODY WAVE DRAG IN POUNDS = ,E14.7,/30X,
2 32H R13/(RHO/2*WTSURF*V**2) = ,F10.3,4HE-03)
445 FORMAT(//T30,32H STRUT-BODY WAVE DRAG IN LBS = ,E14.7,/30X,
2 32H R15B/(RHO/2*WTSURF*V**2) = ,F10.3,4HE-03,
3 10X,10H CW1 TOT =,F10.3,4HE-03)
450 FORMAT(//T30,32H STRUT INTERFERENCE DRAG LBS = ,E14.7,/30X,
2 32H R12S/(RHO/2*WTSURF*V**2) = ,F10.3,4HE-03)
455 FORMAT(//T30,32H BODY INTERFERENCE DRAG LBS = ,E14.7,/30X,
2 32H R12B/(RHO/2*WTSURF*V**2) = ,F10.3,4HE-03)
460 FORMAT(//T30,32H STRUT-BODY INTERFERENCE DRAG LB = ,E14.7,/30X,
2 32H R12SB/(RHO/2*WTSURF*V**2) = ,F10.3,4HE-03,
3 10X,10H CW2 TOT =,F10.3,4HE-03,/,
4 86X," RW TOTAL = ,E14.7,/,
5 86X," CW TOTAL = ,F10.3,"E-03")
461 FORMAT(//30X,* CFORM = ,F10.3,4HE-03,
1/30X,*RESIDUAL DRAG COEFF. = ,F10.3,*E-03,
2/30X,*RESIDUAL RESISTANCE IN LBS= ,E14.7)
465 FORMAT(30X,32H STRUT FRICTION DRAG IN LBS = ,E14.7,
1/30X,32H RFS/(RHO/2*WTSURF*V**2) = ,F10.3,4HE-03,
2/30X,32H BODY FRICTION DRAG IN LBS = ,E14.7,
3/30X,32H RFB/(RHO/2*WTSURF*V**2) = ,F10.3,4HE-03,
4/86X," RF TOTAL = ,E14.7,
5/86X," CF TOTAL = ,F10.3,"E-03")
470 FORMAT(//30X,"APPENDAGE DRAG IN POUNDS = ,E14.7,
1/30X,"APPENDAGE DRAG COEFFICIENT = ,F10.3,"E-03")
475 FORMAT(//86X," RT = ,E14.7,
1/86X," CT = ,F10.3,"E-03",
2/86X," EHP = ,F10.3)
END
155

```

```

1      SUBROUTINE RWAVE(B,D,NLOC2)
C
C
5      SUBROUTINE FOR COMPUTING AUXILIARY FUNCTIONS T AND W
      USING CHEBYCHEV EXPANSION - PROGRAM VERSION OF 8/05/72
      MODIFICATIONS BY A M REED OCTOBER 1974
C
C
10     COMMON / AUX / TS(3,3), WS(3,3), TB(3,3), WB(3,3), TSB(3,3),
1      WSB(3,3), TS12(3,3), WS12(3,3), TB12(3,3), WB12(3,3), TSB12(3,3),
2      WSB12(3,3), TSBP(3,3), WSBP(3,3), TSB12P(3,3), WSB12P(3,3)
C
C
15     COMMON / COEFS / ASM(3), BSM(3), ABM(3), BBM(3), MMAX
C
C
C      COMMON / INPUT / XLS, HS, TSMAX, CWP, CLCF, CIYY, XLB, HB, AX, CP,
1      CLCB, SEPDIS, CSTRUT, CSTR2, PDIA, SPAN, CHORD, TFINS, NLOC
C
C
20     COMMON / OMEGA / VMFPS, GAMMAOS, GAMAOB, GOSQ, HSOLS, HBOLB, WETS,
1      WETB, WETFIN, WTSURF, SEP, PHIS, PHIB, RATIOL, CFS, CFB, APPDRG
C
C
C      COMMON / PHYSCO / RHO,GNU,G,PI,DELFCF
C
25     COMMON/PSI/NPTSZ,PTSAF,EXPN,NALMAX,NAL,TAIL,ALFA
1      ,ALSMAX,NSTEPS
C
C      DIMENSION SIMP(100),VJALFA(7)
C
30     MMAX2=MMAX*2
C
C      SET THE T AND W ARRAYS TO ZERO
C
C      CALL RINIT
C
35     INTEGRATION OF T AND W FROM ALPHA EQUALS GAMMAOS TO GAMMAOS+1.0
      MAKE TRIGONOMETRIC SUBSTITUTION TO AVOID SQUARE ROOT
      SINGULARITY AT ALPHA EQUALS GAMMAOS
C
C      ZETA IS THE VARIABLE OF INTEGRATION--ZETA=SQRT(ALPHA-GAMMAOS)
C      NPTSZ IS THE NUMBER OF POINTS TO TAKE IN THE INTERVAL
C      0.0 LESS THAN OR EQUAL ZETA LESS THAN OR EQUAL 1.0
C      AINCR IS THE STEP SIZE FOR THE ZETA INTEGRATION
C      ALFAZ IS THE VALUE OF ALPHA FOR ZETA VARYING FROM ZERO TO ONE
C
45     SEP = (2.0/GAMMAOS)*SEPDIS/XLS
      CALL SIMPSN(NPTSZ,SIMP)
      XNPTZ=FLOAT(NPTSZ)
      AINCR = 1.0/(XNPTZ-1.0)
      HZETA = AINCR/3.0
      DO 30 I=1,NPTSZ
        WTINT = SIMP(I) * HZETA
        EI=FLOAT(I-1)
        ZETA = AINCR*EI
        ALFAZ = GAMMAOS+ZETA**2
        ALFAQ=ALFAZ**2
        SQZTA=2.0/SORT(GAMMAOS+ALFAZ)
C
55
58

```

74/74 OPT=0 ROUND=0 / TRACE

SUBROUTINE RWAVE

60	C	SEPCOS = COS(SEP*ALFAZ+SQRT(ALFASQ-GOSQ))	60	RWAVE
		CALL RINTEG(ALFAZ,B,D,NLOC2,WTINT,SEPCOS,SQZTA)	61	RWAVE
	C		62	RWAVE
	30		63	RWAVE
	C		64	RWAVE
	C		65	RWAVE
65	C	INTEGRATION FOR T AND W FROM ALPHA EQUALS GAMMAOS+1.0 TO ALMAX	66	RWAVE
			67	RWAVE
			68	RWAVE
			69	RWAVE
			70	RWAVE
70	20	1 ALMAX = SORT((2.302585*EXPN*GAMAOS) / (2.*HBOLB*RATIOIOL) + (GAMAOS+1.)*2)	71	RWAVE
			72	RWAVE
			73	RWAVE
			74	RWAVE
			75	RWAVE
75		WRITE (6,302)	76	RWAVE
		GO TO 215	77	RWAVE
		CONTINUE	78	RWAVE
212		NAL = NAL + 2	79	RWAVE
		DDDA = (2.0*ALFASQ-GOSQ)/SQRT(ALFASQ-GOSQ)	80	RWAVE
80		SSIZE=(2.*PI)/((2.*RATIOIOL+DDDA*SEP)*PTSAP)	81	RWAVE
		TWT= SSIZE/3.	82	RWAVE
		WTINT = TWT + PWT	83	RWAVE
		PWT = TWT	84	RWAVE
		GO TO 250	85	RWAVE
215		CONTINUE	86	RWAVE
		ISIMP=3	87	RWAVE
		WTINT = PWT	88	RWAVE
		GO TO 250	89	RWAVE
220		CONTINUE	90	RWAVE
		WTINT = 4.*TWT	91	RWAVE
		ISIMP = 2	92	RWAVE
250		CONTINUE	93	RWAVE
		ALFASQ=ALFA*2	94	RWAVE
		SQ=1.0/SORT(ALFASQ-GOSQ)	95	RWAVE
95		SEPCOS = COS(SEP*ALFA/SQ)	96	RWAVE
	C		97	RWAVE
		CALL RINTEG(ALFA,B,D,NLOC2,WTINT,SEPCOS,SQ)	98	RWAVE
	C		99	RWAVE
		ALFA = ALFA + SSIZE	100	RWAVE
100		GO TO (220,210,270), ISIMP	101	RWAVE
	C		102	RWAVE
	C	ADD TAIL INTEGRATION TO TS AND WS	103	RWAVE
	C		104	RWAVE
270		CONTINUE	105	RWAVE
		ALFA = ALFA - SSIZE	106	RWAVE
105		NSTEPS = 0	107	RWAVE
		IF (ALSMAX.LE.ALFA) GO TO 278	108	RWAVE
		SSIZE = PI/PTSAP	109	RWAVE
		ANSTEP = (ALSMAX-ALFA)/SSIZE	110	RWAVE
110		NSTEPS = IFIX(ANSTEP/2.)*2 + 2	111	RWAVE
		SSIZE = (ALSMAX-ALFA) / NSTEPS	112	RWAVE
		*T = SSIZE/3.	113	RWAVE
		ANW/T = 1.	114	RWAVE
		NDO = NSTEPS + 1	115	RWAVE

AD-A099 533

DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CE--ETC F/G 20/4
DOCUMENTATION FOR SWATH SHIP RESISTANCE AND PROPULSION PREDICTI--ETC(U)
APR 81 A M REED

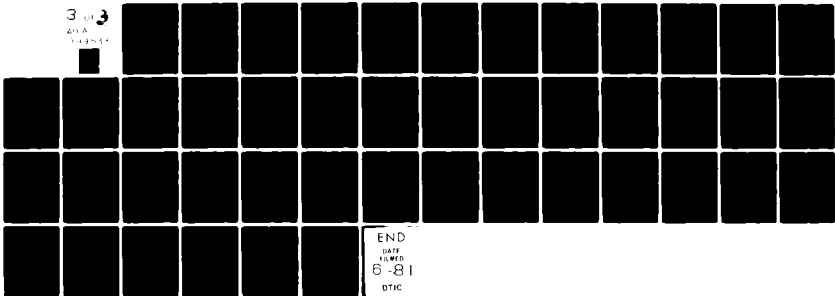
UNCLASSIFIED

DTNSRDC/SPD-0927-02

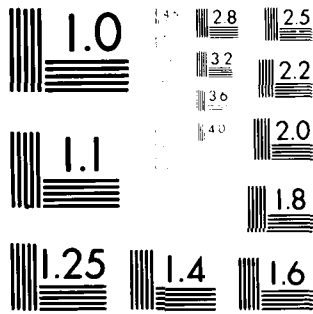
NL

3 of 3

DTNSRDC



END
DATE
FILMED
6-81
DTIC



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

04/07/81 13.10.59

FTN 4.6+460

74/74 OPT=0 ROUND=0 TRACE

SUBROUTINE RWAVE

```

115      CON=1.0
      DO 277 IDO = 1, NDO
        ALFASQ=ALFA**2
        IF (NLOC2.NE.0) CON=2.0*(1.0+CON*((B-D)*ALFA**2.0))
        CALL BESSJ (ALFA,MMAX2,VJALFA)
        VS = PHIS*ALFASQ
        SQ=1.0/SQRT(ALFASQ-GOSQ)
        ESA = 1. - EXP (-VS)
        ES = SQ * ESA**2 / ALFASQ
        DO 276 M = 1,MMAX
          MAA = 2*N
          MBB = MAA + 1
          DO 275 N = M, MMAX
            NAA = 2*N
            NBB = NAA + 1
            WTINT = ANWT * WT
            TSA = WTINT*ES*VJALFA(MAA)*VJALFA(NAA)*CON
            WSA = WTINT*ES*VJALFA(MBB)*VJALFA(NBB)*CON
            TS(M,N) = TS(M,N) + TSA
            WS(M,N) = WS(M,N) + WSA
          CONTINUE
        CONTINUE
        ALFA = ALFA + SSIZE
        BNWT = 2.
        IF (ANWT.LE.2.) BNWT = 4.
        IF (IDO.EQ.NDO-1) BNWT = 1.
        ANWT = BNWT
      CONTINUE
      CONTINUE
      ALFASQ = ALFA**2
      RAT = GAMADS/ALFA
      IF (RAT.LT.1.E-4) GO TO 280
      TAIL=(-SQRT(ALFASQ-GOSQ)/ALFASQ+ASIN(RAT)/GAMADS)/(2.0*PI*GOSQ)
      GO TO 290
    CONTINUE
    TAIL=1.0/(3.0*PI*ALFA*ALFASQ)
  C
  C   CALCULATE TAIL RESULTS
  C
290  CONTINUE
      DO 296 M = 1,MMAX
        DO 295 N = M, MMAX
          SIGN = (-1)**(M+N)
          TS(M,N) = TS(M,N) - SIGN*TAIL
          WS(M,N) = WS(M,N) + SIGN*TAIL
        CONTINUE
      CONTINUE
      MULTIPLY BY CONSTANTS AND REFLECT SYMMETRIC MATRICES)
  C
  C   CALL REFLKT
  C
      RETURN
      FORMAT (*1ALFA INTEGRATION REACHED NALMAX*)
      END

```

```

1      SUBROUTINE RINTEG(ALFA,B,D,NLOC2,TWINT,SEPCOS,SQ)
      C
      C      NUMERICAL INTEGRATION FOR AUXILIARY FUNCTIONS T AND W
      C
5      COMMON / AUX / TS(3,3), WS(3,3), TB(3,3), WB(3,3), TSB(3,3),
      1 WSB(3,3), TS12(3,3), WS12(3,3), TB12(3,3), WB12(3,3), TSB12(3,3),
      2 WSB12(3,3), TSBP(3,3), WSBP(3,3), TSB12P(3,3), WSB12P(3,3)
      C
10     COMMON / COEFS / ASM(3), BSM(3), ABM(3), BBM(3), MMAX
      C
      COMMON / INPUT / XLS, HS, TSMAX, CWP, CLCF, CIYY, XLB, HB, AX, CP,
      1 CLCB, SEPDIS, CSTRUT, CSTRY2, POIA, SPAN, CHORD, TFINS, NLOC
      C
15     COMMON / OMEGA / VMFPS, GAMAOS, GAMA0B, GOSQ, HSOLS, HBOLB, WETS,
      1 WETB, WETFIN, WTSURF, SEP, PHIS, PHIB, RATIOL, CFS, CFB, APPDRG
      C
      DIMENSION VJALFA(7),VJBETA(7)
      C
      C      MMAX IS MAXIMUM NUMBER OF TERMS IN CHEBCHEV SERIES
      C
20     MMAX2 = 2*MMAX
      BETA = RATIOL*ALFA
      ALFASQ=ALFA**2
      C
      C      CALCULATE FACTORS FOR STRUT NOT CENTERED
      C
25     CO=COS(B*ALFA*2.0)
      SI=SIN(B*ALFA*2.0)
      CON=1.0
      IF (NLOC2.EQ.0) GO TO 32
      C
30     CALCULATE FACTORS FOR TANDEM STRUTS
      C
      C      CO=(CO+COS(D*ALFA*2.0))
      C      SI=(SI+SIN(D*ALFA*2.0))
      C      CON=2.0*(1.0+COS((B-D)*ALFA*2.0))
      C      CONTINUE
      C
35     COMPUTE AUXILIARY FUNCTIONS VJALFA AND VJBETA
      C
40     CALL BESSJ (ALFA, MMAX2, VJALFA )
      CALL BESSJ ( BETA, MMAX2, VJBETA )
      C
      C      COMPUTE AUXILIARY FUNCTIONS ES AND EB FOR ALPHA FROM
      C
45     VS=PHIS*ALFASQ
      VB = PHIB*BETA**2
      ESA = 1.0
      EBA = 0.0
      IF(VS .LT. 300.) ESA = ESA-EXP(-VS)
      IF(VB .LT. 300.) EBA = EXP(-VB)
      ES=SQ*ESA**2/ALFASQ
      EB=SQ*EBA**2/ALFASQ
      ESB = SQ*ESA*EBA
      C
55     TB,TS,WB,WS,TW12,TS12,WB12, AND WS12 ARE ALL SYMMETRIC WRT M
      C      AND N SO WE NEED ONLY CALCULATE THE UPPER HALF OF THESE
      C
      RINTEG 2
      RINTEG 3
      RINTEG 4
      RINTEG 5
      RINTEG 6
      RINTEG 7
      RINTEG 8
      RINTEG 9
      RINTEG 10
      RINTEG 11
      RINTEG 12
      RINTEG 13
      RINTEG 14
      RINTEG 15
      RINTEG 16
      RINTEG 17
      RINTEG 18
      RINTEG 19
      RINTEG 20
      RINTEG 21
      RINTEG 22
      RINTEG 23
      RINTEG 24
      RINTEG 25
      RINTEG 26
      RINTEG 27
      RINTEG 28
      RINTEG 29
      RINTEG 30
      RINTEG 31
      RINTEG 32
      RINTEG 33
      RINTEG 34
      RINTEG 35
      RINTEG 36
      RINTEG 37
      RINTEG 38
      RINTEG 39
      RINTEG 40
      RINTEG 41
      RINTEG 42
      RINTEG 43
      RINTEG 44
      RINTEG 45
      RINTEG 46
      RINTEG 47
      RINTEG 48
      RINTEG 49
      RINTEG 50
      RINTEG 51
      RINTEG 52
      RINTEG 53
      RINTEG 54
      RINTEG 55
      RINTEG 56
      RINTEG 57
      RINTEG 58

```

04/07/81 13.10.59

FTN 4.6+460

74/74 OPT=0 ROUND=0/ TRACE

SUBROUTINE RINTEG

```

C
C
60      DO 56 M=1,MMAX
      MAA = 2*M
      MBB = 2*M+1
      DO 55 N=M,MMAX
      NAA = 2*N
      NBB = 2*N+1
      TSA = WTINT*ES*VJALFA(MAA)*VJALFA(NAA)*CON
      WSA = WTINT*ES*VJALFA(MBB)*VJALFA(NBB)*CON
      IF(EB.EQ.0.0) GO TO 54
      TBA = WTINT*EB*VJBETA(MAA)*VJBETA(NAA)
      WBA = WTINT*EB*VJBETA(MBB)*VJBETA(NBB)
      TSBA = WTINT*ESB*VJBETA(NAA)*VJALFA(MAA)*CO
      TSBB = - WTINT*ESB*VJBETA(NAA)*VJALFA(MBB)*SI
      WBSA = WTINT*ESB*VJBETA(NBB)*VJALFA(MBB)*CO
      WBSB = WTINT*ESB*VJBETA(NBB)*VJALFA(MAA)*SI
      TSBBP = - WTINT*ESB*VJBETA(MAA)*VJALFA(NBB)*SI
      WBSBP = WTINT*ESB*VJBETA(MBB)*VJALFA(NBB)*CO
      TB(M,N) = TB(M,N)+TBA
      WB(M,N) = WB(M,N)+WBA
      TSB(M,N) = TSB(M,N)+TSBA
      TSBP(M,N) = TSBP(M,N)+TSBAP
      WSB(M,N) = WSB(M,N)+WSBA
      WSBP(M,N) = WSBP(M,N)+WSBAP
      IF(M.EQ.N) GO TO 54
      TSBN(M) = TSBN(M)+TSBB
      TSBP(N,M) = TSBP(N,M)+TSBBP
      WSB(N,M) = WSB(N,M)+WSBB
      WSBP(N,M) = WSBP(N,M)+WSBBP
      CONTINUE
54      TS(M,N) = TS(M,N)+TSA
      WS(M,N) = WS(M,N)+WSA
      TS12(M,N) = TS12(M,N)+TSA*SEPCOS
      WS12(M,N) = WS12(M,N)+WSA*SEPCOS
      IF(EBA.EQ.0.0) GO TO 55
      TB12(M,N) = TB12(M,N)+TBA*SEPCOS
      WB12(M,N) = WB12(M,N)+WBA*SEPCOS
      TS12(M,N) = TS12(M,N)+TSBA*SEPCOS
      TS12P(M,N) = TS12P(M,N)+TSBAP*SEPCOS
      WS12(M,N) = WS12(M,N)+WSBA*SEPCOS
      WS12P(M,N) = WS12P(M,N)+WSBAP*SEPCOS
      IF(M.EQ.N) GO TO 55
      TS12(N,M) = TS12(N,M)+TSBB*SEPCOS
      TS12P(N,M) = TS12P(N,M)+TSBBP*SEPCOS
      WSB12(N,M) = WSB12(N,M)+WSBB*SEPCOS
      WSB12P(N,M) = WSB12P(N,M)+WSBBP*SEPCOS
55      CONTINUE
56      CONTINUE
      RETURN
      END

```

```

1      SUBROUTINE RINIT
      C
      C      INITIALIZE T AND W ARRAY
      C
5      COMMON / AUX / TS(3,3), WS(3,3), TB(3,3), WB(3,3), TSB(3,3),
      1 WSB(3,3), TS12(3,3), WS12(3,3), TB12(3,3), WB12(3,3), TSB12(3,3),
      2 WSB12(3,3), TSBP(3,3), WSBP(3,3), TSB12P(3,3), WSB12P(3,3)
      C
10     COMMON / COEFS / ASM(3), BSM(3), ABM(3), BBM(3), MMAX
      C
      DO 5 M=1,MMAX
      DO 4 N=M,MMAX
      TS(M,N) = 0.0
      WS(M,N) = 0.0
      TB(M,N) = 0.0
      WB(M,N) = 0.0
      TSB(M,N) = 0.0
      TSBP(M,N) = 0.0
      TSB(N,M)=0.0
      TSBP(N,M)=0.0
      WSB(M,N) = 0.0
      WSBP(M,N) = 0.0
      WSB(N,M)=0.0
      WSBP(N,M)=0.0
      TS12(M,N) = 0.0
      WS12(M,N) = 0.0
      TB12(M,N) = 0.0
      WB12(M,N) = 0.0
      TSB12(M,N) = 0.0
      TSB12P(M,N) = 0.0
      TSB12(N,M)=0.0
      TSB12P(N,M)=0.0
      WSB12(M,N) = 0.0
      WSB12P(M,N) = 0.0
      WSB12(N,M)=0.0
      WSB12P(N,M)=0.0
      4      CONTINUE
      5      CONTINUE
      RETURN
      END
2      RINIT
3      RINIT
4      RINIT
5      RINIT
6      RINIT
7      RINIT
8      RINIT
9      RINIT
10     RINIT
11     RINIT
12     RINIT
13     RINIT
14     RINIT
15     RINIT
16     RINIT
17     RINIT
18     RINIT
19     RINIT
20     RINIT
21     RINIT
22     RINIT
23     RINIT
24     RINIT
25     RINIT
26     RINIT
27     RINIT
28     RINIT
29     RINIT
30     RINIT
31     RINIT
32     RINIT
33     RINIT
34     RINIT
35     RINIT
36     RINIT
37     RINIT
38     RINIT
39     RINIT
40     RINIT
41     RINIT

```

```

1      SUBROUTINE REFLECT
      C
      C      REFLECT THE SYMMETRICAL MATRICES OF T AND W
      C
      COMMON / AUX / TS(3,3), WS(3,3), TB(3,3), WB(3,3), TSB(3,3),
1      WSB(3,3), TS12(3,3), WS12(3,3), TB12(3,3), WB12(3,3), TSB12(3,3),
2      WSB12(3,3), TSBP(3,3), WSBP(3,3), TSB12P(3,3), WSB12P(3,3)
      C
      COMMON / COEFS / ASM(3), BSM(3), ABM(3), BBM(3), MMAX
      C
      DO 64 M = 1, MMAX
      EN=FLOAT(M)
      U = 2.0*EM-1.0
      DO 63 N=M,MMAX
      EN=FLOAT(N)
      FOURMN=4.0*EM*EN
      UV=U*(2.0*EN-1.0)
      TS(M,N) = TS(M,N)*UV
      TS(N,M) = TS(M,N)
      WS(M,N) = WS(M,N)*FOURMN
      WS(N,M) = WS(M,N)
      TB(M,N) = TB(M,N)*UV
      TB(N,M) = TB(M,N)
      WB(M,N) = WB(M,N)*FOURMN
      WB(N,M) = WB(M,N)
      TSB(M,N) = TSB(M,N)*UV
      TSBP(M,N) = TSBP(M,N)*UV
      WSB(M,N) = WSB(M,N)*FOURMN
      WSBP(M,N) = WSBP(M,N)*FOURMN
      IF (M.EQ.N) GO TO 62
      TSB(N,M)=TSB(M,N)*UV
      TSBP(N,M)=TSBP(M,N)*UV
      WSB(N,M)=WSB(M,N)*FOURMN
      WSBP(N,M)=WSBP(M,N)*FOURMN
      CONTINUE
62
      TS12(M,N) = TS12(M,N)*UV
      TS12(N,M) = TS12(M,N)
      WS12(M,N) = WS12(M,N)*FOURMN
      WS12(N,M) = WS12(M,N)
      TB12(M,N) = TB12(M,N)*UV
      TB12(N,M) = TB12(M,N)
      WB12(M,N) = WB12(M,N)*FOURMN
      WB12(N,M) = WB12(M,N)
      TSB12(M,N) = TSB12(M,N)*UV
      TSB12P(M,N) = TSB12P(M,N)*UV
      WSB12(M,N) = WSB12(M,N)*FOURMN
      WSB12P(M,N) = WSB12P(M,N)*FOURMN
      IF (M.EQ.N) GO TO 63
      TSB12(N,M) = TSB12(M,N)*UV
      TSB12P(N,M) = TSB12P(M,N)*UV
      WSB12(N,M) = WSB12(M,N)*FOURMN
      WSB12P(N,M) = WSB12P(M,N)*FOURMN
      CONTINUE
63
      CONTINUE
64
      RETURN
      END

```

FUNCTION CFITTC	74/74	DPT=0	ROUND=*	TRACE	FTN 4.6+460	04/07/81	13.10.59	PAGE	1
-----------------	-------	-------	---------	-------	-------------	----------	----------	------	---

1	C	FUNCTION CFITTC(RN)							
	C	THIS FUNCTION DETERMINES THE ITTC CORRELATION ALLOWANCE							
	C								
5	C	THIS FUNCTION DETERMINES THE ITTC CORRELATION ALLOWANCE							
	C								
10		DEN=ALOG10(RN)-2.0							
		CFITTC=0.075/DEN**2							
		RETURN							
		END							

CFITTC	2
CFITTC	3
CFITTC	4
CFITTC	5
CFITTC	6
CFITTC	7
CFITTC	8
CFITTC	9
CFITTC	10
CFITTC	11
CFITTC	12

FUNCTION EVAL	74/74	OPT=0 ROUND=*	TRACE	FTN 4.6+460	04/07/81	13.10.59	PAGE 1
---------------	-------	---------------	-------	-------------	----------	----------	--------

```

1      FUNCTION EVAL(X,A,B,MAX)
2
3      C
4      C      THIS FUNCTION EVALUATES THE CHEBYSHEV SERIES
5      C      F(X)=SUMMATION M=1,MAX A(M)*U(M,X)+B(M)*V(M,X)
6      C
7
8      DIMENSION A(MAX),B(MAX)
9      THETA=ASIN(X)
10     T=0.0
11     DO 10 M=1,MAX
12         XM=FLOAT(M)
13         ARGU=(2.0*XM-1.0)*THETA
14         ARGV=2.0*XM*THETA
15         U=COS(ARGU)
16         V=SIN(ARGU)
17         T=T+A(M)*U+B(M)*V
18     10 CONTINUE
19     EVAL=T
20     RETURN
21     END

```

```

1      SUBROUTINE WSURFB(AREA)
      C
      C      THIS SUBROUTINE DETERMINES THE SURFACE AREA OF A BODY OF
      C      REVOLUTION GIVEN BY CHEBYSHEV COEFFICIENTS
      C
      C      COMMON / COEFS / ASM(3), BSM(3), ABM(3), BBM(3), MMAX
      C
      C      COMMON / INPUT / XLS, HS, TSMAX, CWP, CLCF, CIYY, XLB, HB, AX, CP,
1      CLCB, SEPDIS, CSTRUT, CSTR2, PDIA, SPAN, CHORD, TFINS, NLOC
      C
      C      COMMON / PHYSCO / RHO,GNU,G,PI,DELCF
      C
      RAD=SQRT(AX/PI)
      CON=(2.0*PI/RAD/XLB)**2
      DX=0.05
      DXSQ=DX*DX
      R=0.0
      AREA=0.0
      DO 10 I=1,40
      RL=R
      X=DX*FLOAT(I)-1.0
      R=EVAL(X,ABM,BBM,MMAX)
      IF (R.LT.0.0) R=0.0
      R=SQRT(R)
      RM=(RL+R)*0.5
      DR=R-RL
      AREA=AREA+RM*SQRT(DXSQ+CON*DR**2)
10     CONTINUE
      AREA=PI*RAD*XLB*AREA
      RETURN
      END
30
2      WSURFB
3      WSURFB
4      WSURFB
5      WSURFB
6      WSURFB
7      WSURFB
8      WSURFB
9      WSURFB
10     WSURFB
11     WSURFB
12     WSURFB
13     WSURFB
14     WSURFB
15     WSURFB
16     WSURFB
17     WSURFB
18     WSURFB
19     WSURFB
20     WSURFB
21     WSURFB
22     WSURFB
23     WSURFB
24     WSURFB
25     WSURFB
26     WSURFB
27     WSURFB
28     WSURFB
29     WSURFB
30     WSURFB
31     WSURFB
32     WSURFB

```

04/07/81 13.10.59

FIN 4.6+460

SUBROUTINE WSURFS 74/74 OPT=0 ROUND=0/ TRACE

```

1      SUBROUTINE WSURFS(AREA)
2
3      C
4      C      THIS SUBROUTINE DETERMINES THE SURFACE AREA OF A STRUT WHOSE
5      C      THICKNESS DISTRIBUTION IS GIVEN BY CHEBYSHEV COEFFICIENTS
6      C
7      C      COMMON / COEFS / ASM(3), BSM(3), ABM(3), BBM(3), MMAX
8      C
9      C      COMMON / INPUT / XLS, HS, TSMAX, CWP, CLCF, CIYY, XLB, HB, AX, CP,
10     C      1 CLCB, SEPDIS, CSTRUT, CSTRT2, PDIA, SPAN, CHORD, TFINS, NLOC
11     C
12     C      CON=(TSMAX/XLS)**2
13     C      DX=0.05
14     C      DXSQ=DX**2
15     C      AREA=0.0
16     C      T=0.0
17     C      DO 10 I=1,40
18     C          TL=T
19     C          X=DX*FLOAT(I)-1.0
20     C          T=EVAL(X,ASN,BSM,MMAX)
21     C          DT=T-TL
22     C          AREA=AREA+SORT(DXSQ+CON*DT**2)
23     C      CONTINUE
24     C      AREA=XLS*HS*AREA
25     C      RETURN
26     C      END

```

1		SUBROUTINE SIMPSN(NPTS,SIMP)	SIMPSN	2
C			SIMPSN	3
C		SET UP SIMPSON'S MULTIPLIERS FOR INTEGRATION	SIMPSN	4
C		ODD NUMBER OF POINTS GREATER THAN 1 REQUIRED	SIMPSN	5
5			SIMPSN	6
		DIMENSION SIMP(NPTS)	SIMPSN	7
		IF((NPTS/2)*2.NE.NPTS) GO TO 3	SIMPSN	8
		WRITE(6,90)	SIMPSN	9
		NPTS = NPTS+1	SIMPSN	10
10	3	CONTINUE	SIMPSN	11
		SIMP(1) = 1.0	SIMPSN	12
		SIMP(NPTS) = 1.0	SIMPSN	13
		NPTSM1=NPTS-1	SIMPSN	14
		SM=2.0	SIMPSN	15
15		DO 5 J=2,NPTSM1	SIMPSN	16
		SM=6.0-SM	SIMPSN	17
		SIMP(J)=SM	SIMPSN	18
	5	CONTINUE	SIMPSN	19
		RETURN	SIMPSN	20
20	90	FORMAT(1H0,*EVEN NO. OF POINTS GIVEN - ADDITIONAL POINT SUPPLIED*)	SIMPSN	21
		END	SIMPSN	22

04/07/81 13.10.59

FTN 4.6+460

74/74 OPT=0 ROUNO=*/ TRACE

SUBROUTINE SUM

```

1      C
2      C
3      C
4      C
5      C
6      C
7      C
8      C
9      C
10     C
11     C
12     C
13     C
14     C
15     C
16     C
17     C
18     C
19     C
20     C
21     C
22     C
23     C
24     C
25     C
26     C
27     C
28     C
29     C
30     C
31     C
32     C
33     C
34     C
35     C
36     C
37     C
38     C
39     C
40     C
41     C
42     C
43     C
44     C
45     C
46     C
47     C
48     C

SUBROUTINE SUM(SUM1S,SUM1B,SUM1SB,SUM12S,SUM12B,SUM12SB)
      SUM COMPUTES THE MATRIX PRODUCTS
      SUM = A(TRANPOSE) * T * A + B(TRANPOSE) * W * B

      COMMON / AUX / TS(3,3), WS(3,3), TB(3,3), WB(3,3), TSB(3,3),
1     WSB(3,3), TS12(3,3), WS12(3,3), TB12(3,3), WB12(3,3), TSB12(3,3),
2     WSB12(3,3), TSBP(3,3), WSBP(3,3), TSB12P(3,3), WSB12P(3,3)

      COMMON / COEFS / ASM(3), BSM(3), ABM(3), BBM(3), MMAX

      SUMA1 = 0.0
      SUMB1 = 0.0
      SUMA2 = 0.0
      SUMB2 = 0.0
      SUMA3 = 0.0
      SUMB3 = 0.0
      SUMA4 = 0.0
      SUMB4 = 0.0
      SUMA5 = 0.0
      SUMB5 = 0.0
      SUMA6 = 0.0
      SUMB6 = 0.0
      DO 20 M=1,MMAX
        DO 10 N=1,MMAX
          SUMA1 = SUMA1+ASM(M)*ASM(N)*TS(M,N)
          SUMB1 = SUMB1+BSM(M)*BSM(N)*WS(M,N)
          SUMA2 = SUMA2+ABM(M)*ABM(N)*TB(M,N)
          SUMB2 = SUMB2+BBM(M)*BBM(N)*WB(M,N)
          SUMA3 = SUMA3+ABM(N)*ASM(M)*TSB(M,N)+BSM(M)*TSBP(M,N)
          SUMB3 = SUMB3+BBM(N)*BSM(M)*WSB(M,N)+ASM(M)*WSBP(M,N)
          SUMA4 = SUMA4+ASM(M)*ASM(N)*TS12(M,N)
          SUMB4 = SUMB4+BSM(M)*BSM(N)*WS12(M,N)
          SUMA5 = SUMA5+ABM(M)*ABM(N)*TB12(M,N)
          SUMB5 = SUMB5+BBM(M)*BBM(N)*WB12(M,N)
          SUMA6 = SUMA6+ABM(N)*ASM(M)*TSB12(M,N)+BSM(M)*TSB12P(M,N)
          SUMB6 = SUMB6+BBM(N)*BSM(M)*WSB12(M,N)+ASM(M)*WSB12P(M,N)
        CONTINUE
      CONTINUE
      SUM1S = SUMA1+SUMB1
      SUM1B = SUMA2+SUMB2
      SUM1SB = SUMA3+SUMB3
      SUM12S = SUMA4+SUMB4
      SUM12B = SUMA5+SUMB5
      SUM12SB = SUMA6+SUMB6
      RETURN
      END

```



```

1      SUBROUTINE FOILSC(XS,CFS,TOC,TOC2,TOC4,SS,TS2,SBS,CLOTB,RFS,
      +RVAS,RPS,RINTS,RBS)
      C
      C      FLAT FRICTIONAL RESISTANCE
      C
      C      RFS=XS*2.*CFS
      C
      C      RESISTANCE DUE TO VELOCITY AUGMENTATION
      C
      C      RVAS=XS*4.*CFS*TOC
      C
      C      RESISTANCE DUE TO PRESSURE OR SEPARATION (VISCOUS IN NATURE)
      C
      C      RPS=XS*120.*CFS*TOC4
      C
      C      ADDED RESISTANCE DUE TO INTERSECTION WITH HULL
      C
      C      RINTS=(XS/SS)*TS2*((.75*TOC)-(.0003/TOC2))
      C
      C      BASE DRAG DUE TO BLUNTNESS OF TRAILING EDGE
      C
      C      IF(SBS.LE.0.) GO TO 9000
      C      RBS=(XS/SS)*SBS*(.135/((CLOTB*CFS)**(1./3.)))
      C      CONTINUE
      C      RBS=0.
      C      RETURN
      C      END
9000
25
28

```

```

1      SUBROUTINE FRICT(CF,VF,XLEN,XNU)
      C
      C
      C      FLAT PLATE FRICTIONAL COEFFICIENT BASED ON LOCAL REYNOLDS NO.
      RE=VF*XLEN/XNU
      IF(RE.LT.50000.) GO TO 30
      CFTURB=.075/ALOG10(RE/100.)**2
      Y=CFTURB
      CONTINUE
10     CFTURB=.058564/ALOG10(RE*Y)**2
      IF(ABS(CFTURB-Y).LE.1.0E-07) GO TO 20
      Y=(CFTURB+Y)/2.
      GO TO 10
      CONTINUE
20     CF=CFTURB*(143.18/RE)
      RETURN
      CONTINUE
30     CF=1.328/SQRT(RE)
      RETURN
      END
21

```

199

```

C
60      CMT = 1. - (0.71) EXP(1.58*(DIAM,IS) DIA1)
      CMTD(IS)=CMT
      H(IS) = HB
      TDES(IS) = EMPDES*550.0/(VDES*OMT)
      VADES(IS) = VDES*OMT/1.6878

C
65      DETERMINE THE CHARACTERISTICS AND PERFORMANCE OF PROPULSOR
C
C      CALL PRODES(IS)
C
C      PRINT PERFORMANCE DATA OF PROPULSOR
C
C      CALL PROPPR(IS,TITLE)
C
C      SHDES = 6.2831853*QDES(IS)-RPMDES(IS)/35000.
      PC = TDES(IS)*VDES*OMT/(SHDES*550.0)
      SHPINV) = SHDES
      WRITE(6,703) SHDES,PC
      CT=8.0*TDES(IS)/13.14159*RHOD(IP)*VADES(IS)*DIAM(IS)*1.6878)**2)
      WRITE(6,704) CT,OMT,OMWT

C
80      CALCULATE OPEN WATER CURVE OF PROPULSOR
C
C      CALL PROPRIS(VOF,TOF,DUMMY,TITLE)
C
C      DO 22 I=1,NV
      VKOFF = VOFF(I)/1.6878
      TOFF = EMP(I)*550.0/(VOFF(I)*OMT)
C
C      DETERMINE OFF DESIGN PERFORMANCE
C
C      CALL PROPRO(IIS,VKOFF,TOFF,SHP(I),TITLE)
C
22      CONTINUE
      RETURN
C
703      FORMAT(/,1X,*SHP= ,F12.3,5X,*ETA = ,F6.3////)
C
704      FORMAT(20X,*THRUST COEFFICIENT = ,F6.3/20X,*THRUST DEDUCTION(1-T)
1 = ,F6.3/20X,*THRUST WAKE (1-WT) = ,F6.3)
      END

```

```

1      SUBROUTINE PRODES(IS)
2
3      C
4      C PRODES IS USED IN DESIGN OF PROPULSION SYSTEM TO DETERMINE THE
5      C CHARACTERISTICS AND PERFORMANCE AT DESIGN CONDITION
6
7      C
8      C INPUT-OUTPUT PARAMETERS ARE INTRODUCED THROUGH COMMON BLOCK /CPROP/.
9      C ARGUMENT -IS- INDICATES PROPULSION SYSTEM BEING CONSIDERED.
10
11      C
12      C COMMON/BARCK/IE
13
14      C
15      C COMMON/CDESC/RHOD(3),ANJD(3),IP
16
17      C
18      C COMMON/CPROP/IPTYP(3),DIAM(3),VADES(3),RPMDES(3),TDES(3),QDES(3),
19      1  EFFDES(3),OMWTD(3),CAV(3),Z(3),H(3),BAR(3),PD(3),J(3),
20      2  ERROR(3),KT(3),KQ(3)
21
22      C
23      REAL J,KT,KQ
24
25      C
26      INTEGER Z,IPTYP
27
28      C
29      ERROR(1S)=0
30      IE=0
31      ITQ=1
32      IPJ=2
33      IPTYP=IPTYP(1S)
34      BAR(1S) = 0.70
35      WRITE(6,600)
36      CONTINUE
37
38      511
39      C
40      C OPTIMIZE RPM OR DIAMETER OF PROPULSOR
41
42      C
43      CALL TROOST(VADES(1S),QDES(1S),TDES(1S),RPMDES(1S),DIAM(1S),Z(1S),
44      1PD(1S),BAR(1S),KT(1S),KQ(1S),EFFDES(1S),J(1S),ITQ,IPJ,RHOD(IP),
45      2IE)
46
47      IF(1ERROR(1S).EQ.3.OR.1ERROR(1S).EQ.4) GO TO 515
48
49      C
50      C EVALUATE BLADE AREA RATIO
51
52      C
53      CALL BURIL(TDES(1S),DIAM(1S),H(1S),VADES(1S),RPMDES(1S),
54      1CAV(1S),BAR(1S),1ERROR(1S),PD(1S),NRET)
55      IF ( NRET .NE. 0 ) GO TO 511
56
57      515
58      C
59      C CONTINUE
60      ERROR(1S)=100*IE+1ERROR(1S)
61      RETURN
62      600
63      C
64      C FORMAT(1H1)
65      END

```

04/07/81 13.10.59

FTN 4.6+460

74/74 OPT=0 ROUND=0/ TRACE

SUBROUTINE PROPER

```

1      SUBROUTINE PROPER(IS,VOF,TOF,PWR,TITLE)
2      C
3      C SUBROUTINE "PROPER" CALCULATES THE OPEN WATER CURVES AND OFF-DESIGN
4      C PERFORMANCE FOR PROPELLOR SYSTEM "IS"
5      C
6      C COMMON/CDESC/RHOD(3),ANUD(3),IP
7      C
8      C COMMON/CPROP/IPTYP(3),DIAM(3),VADES(3),RPMDES(3),TDES(3),QDES(3),
9      C EFFDES(3),OMWTD(3),CAV(3),Z(3),H(3),BAR(3),PD(3),J(3),
10     C IERROR(3),KT(3),KQ(3)
11     C
12     C COMMON/PCOF/CT(4),CO(4)
13     C
14     C COMMON/ROOTC/A(4)
15     C
16     C DIMENSION TITLE(8)
17     C REAL J,KT,KQ,JOF,JOF,KTT,KQQ
18     C
19     C INTEGER Z
20     C
21     C IJK=0
22     C PDC=PD(IS)
23     C
24     C EVALUATE THE COEFFICIENTS OF THE KT AND KQ POLYNOMIAL
25     C
26     C CALL POLCOF(PDC ,BAR(IS),Z(IS))
27     C JK=0.0
28     C
29     C PRINT OPEN WATER CURVE TABLE
30     C
31     C WRITE(6,905) TITLE
32     C WRITE (6,906)
33     C CONTINUE
34     C
35     C KT =CT(1)+(CT(2)+(CT(3)+CT(4)*JK )*JK )*JK
36     C KQ =CQ(1)+(CQ(2)+(CQ(3)+CQ(4)*JK )*JK )*JK
37     C EEE=KTT*JK/(KQ*6.2831853)
38     C WRITE (6,902) JK,KTT,KQ,EEE
39     C IF(KTT.LT.0.03) GO TO 970
40     C JK=JK+0.05
41     C IF ( JK .LE. 2.0 ) GO TO 210
42     C GO TO 970
43     C ENTRY PROPERO
44     C
45     C CALCULATE OFF DESIGN PERFORMANCE
46     C
47     C VAFF=VOF*1.6879*OMWTD(IS)
48     C COEFF=TOF/(RHOD(IP)*DIAM(IS)*DIAM(IS)*VAFF*VAFF)
49     C DO 950 I=1,4
50     C A(I)=CT(I)
51     C CONTINUE
52     C CALL ROOTC(PD(IS),Z(IS),BAR(IS),COEFF,2.1,XKT,XKQ,EEE,JOF)
53     C XKQ =CQ(1)+(CQ(2)+(CQ(3)+CQ(4)*JOF )*JOF )*JOF
54     C KXT =CT(1)+(CT(2)+(CT(3)+CT(4)*JOF )*JOF )*JOF
55     C EEE=XKT*JOF/(XKQ*6.2831853)
56     C PWR=RHOD(IP)*XKQ*DIAM(IS)*DIAM(IS)*VAFF*VAFF*VAFF/(87.5352188*
57     C JOF*JOF*JOF)
58     C
59     C CONTINUE
60     C
61     C
62     C
63     C
64     C
65     C
66     C
67     C
68     C
69     C
70     C
71     C
72     C
73     C
74     C
75     C
76     C
77     C
78     C
79     C
80     C
81     C
82     C
83     C
84     C
85     C
86     C
87     C
88     C
89     C
90     C
91     C
92     C
93     C
94     C
95     C
96     C
97     C
98     C
99     C
100    C
101    C
102    C
103    C
104    C
105    C
106    C
107    C
108    C
109    C
110    C
111    C
112    C
113    C
114    C
115    C
116    C
117    C
118    C
119    C
120    C
121    C
122    C
123    C
124    C
125    C
126    C
127    C
128    C
129    C
130    C
131    C
132    C
133    C
134    C
135    C
136    C
137    C
138    C
139    C
140    C
141    C
142    C
143    C
144    C
145    C
146    C
147    C
148    C
149    C
150    C
151    C
152    C
153    C
154    C
155    C
156    C
157    C
158    C
159    C
160    C
161    C
162    C
163    C
164    C
165    C
166    C
167    C
168    C
169    C
170    C
171    C
172    C
173    C
174    C
175    C
176    C
177    C
178    C
179    C
180    C
181    C
182    C
183    C
184    C
185    C
186    C
187    C
188    C
189    C
190    C
191    C
192    C
193    C
194    C
195    C
196    C
197    C
198    C
199    C
200    C
201    C
202    C
203    C
204    C
205    C
206    C
207    C
208    C
209    C
210    C
211    C
212    C
213    C
214    C
215    C
216    C
217    C
218    C
219    C
220    C
221    C
222    C
223    C
224    C
225    C
226    C
227    C
228    C
229    C
230    C
231    C
232    C
233    C
234    C
235    C
236    C
237    C
238    C
239    C
240    C
241    C
242    C
243    C
244    C
245    C
246    C
247    C
248    C
249    C
250    C
251    C
252    C
253    C
254    C
255    C
256    C
257    C
258    C
259    C
260    C
261    C
262    C
263    C
264    C
265    C
266    C
267    C
268    C
269    C
270    C
271    C
272    C
273    C
274    C
275    C
276    C
277    C
278    C
279    C
280    C
281    C
282    C
283    C
284    C
285    C
286    C
287    C
288    C
289    C
290    C
291    C
292    C
293    C
294    C
295    C
296    C
297    C
298    C
299    C
300    C
301    C
302    C
303    C
304    C
305    C
306    C
307    C
308    C
309    C
310    C
311    C
312    C
313    C
314    C
315    C
316    C
317    C
318    C
319    C
320    C
321    C
322    C
323    C
324    C
325    C
326    C
327    C
328    C
329    C
330    C
331    C
332    C
333    C
334    C
335    C
336    C
337    C
338    C
339    C
340    C
341    C
342    C
343    C
344    C
345    C
346    C
347    C
348    C
349    C
350    C
351    C
352    C
353    C
354    C
355    C
356    C
357    C
358    C
359    C
360    C
361    C
362    C
363    C
364    C
365    C
366    C
367    C
368    C
369    C
370    C
371    C
372    C
373    C
374    C
375    C
376    C
377    C
378    C
379    C
380    C
381    C
382    C
383    C
384    C
385    C
386    C
387    C
388    C
389    C
390    C
391    C
392    C
393    C
394    C
395    C
396    C
397    C
398    C
399    C
400    C
401    C
402    C
403    C
404    C
405    C
406    C
407    C
408    C
409    C
410    C
411    C
412    C
413    C
414    C
415    C
416    C
417    C
418    C
419    C
420    C
421    C
422    C
423    C
424    C
425    C
426    C
427    C
428    C
429    C
430    C
431    C
432    C
433    C
434    C
435    C
436    C
437    C
438    C
439    C
440    C
441    C
442    C
443    C
444    C
445    C
446    C
447    C
448    C
449    C
450    C
451    C
452    C
453    C
454    C
455    C
456    C
457    C
458    C
459    C
460    C
461    C
462    C
463    C
464    C
465    C
466    C
467    C
468    C
469    C
470    C
471    C
472    C
473    C
474    C
475    C
476    C
477    C
478    C
479    C
480    C
481    C
482    C
483    C
484    C
485    C
486    C
487    C
488    C
489    C
490    C
491    C
492    C
493    C
494    C
495    C
496    C
497    C
498    C
499    C
500    C
501    C
502    C
503    C
504    C
505    C
506    C
507    C
508    C
509    C
510    C
511    C
512    C
513    C
514    C
515    C
516    C
517    C
518    C
519    C
520    C
521    C
522    C
523    C
524    C
525    C
526    C
527    C
528    C
529    C
530    C
531    C
532    C
533    C
534    C
535    C
536    C
537    C
538    C
539    C
540    C
541    C
542    C
543    C
544    C
545    C
546    C
547    C
548    C
549    C
550    C
551    C
552    C
553    C
554    C
555    C
556    C
557    C
558    C
559    C
560    C
561    C
562    C
563    C
564    C
565    C
566    C
567    C
568    C
569    C
570    C
571    C
572    C
573    C
574    C
575    C
576    C
577    C
578    C
579    C
580    C
581    C
582    C
583    C
584    C
585    C
586    C
587    C
588    C
589    C
590    C
591    C
592    C
593    C
594    C
595    C
596    C
597    C
598    C
599    C
600    C
601    C
602    C
603    C
604    C
605    C
606    C
607    C
608    C
609    C
610    C
611    C
612    C
613    C
614    C
615    C
616    C
617    C
618    C
619    C
620    C
621    C
622    C
623    C
624    C
625    C
626    C
627    C
628    C
629    C
630    C
631    C
632    C
633    C
634    C
635    C
636    C
637    C
638    C
639    C
640    C
641    C
642    C
643    C
644    C
645    C
646    C
647    C
648    C
649    C
650    C
651    C
652    C
653    C
654    C
655    C
656    C
657    C
658    C
659    C
660    C
661    C
662    C
663    C
664    C
665    C
666    C
667    C
668    C
669    C
670    C
671    C
672    C
673    C
674    C
675    C
676    C
677    C
678    C
679    C
680    C
681    C
682    C
683    C
684    C
685    C
686    C
687    C
688    C
689    C
690    C
691    C
692    C
693    C
694    C
695    C
696    C
697    C
698    C
699    C
700    C
701    C
702    C
703    C
704    C
705    C
706    C
707    C
708    C
709    C
710    C
711    C
712    C
713    C
714    C
715    C
716    C
717    C
718    C
719    C
720    C
721    C
722    C
723    C
724    C
725    C
726    C
727    C
728    C
729    C
730    C
731    C
732    C
733    C
734    C
735    C
736    C
737    C
738    C
739    C
740    C
741    C
742    C
743    C
744    C
745    C
746    C
747    C
748    C
749    C
750    C
751    C
752    C
753    C
754    C
755    C
756    C
757    C
758    C
759    C
760    C
761    C
762    C
763    C
764    C
765    C
766    C
767    C
768    C
769    C
770    C
771    C
772    C
773    C
774    C
775    C
776    C
777    C
778    C
779    C
780    C
781    C
782    C
783    C
784    C
785    C
786    C
787    C
788    C
789    C
790    C
791    C
792    C
793    C
794    C
795    C
796    C
797    C
798    C
799    C
800    C
801    C
802    C
803    C
804    C
805    C
806    C
807    C
808    C
809    C
810    C
811    C
812    C
813    C
814    C
815    C
816    C
817    C
818    C
819    C
820    C
821    C
822    C
823    C
824    C
825    C
826    C
827    C
828    C
829    C
830    C
831    C
832    C
833    C
834    C
835    C
836    C
837    C
838    C
839    C
840    C
841    C
842    C
843    C
844    C
845    C
846    C
847    C
848    C
849    C
850    C
851    C
852    C
853    C
854    C
855    C
856    C
857    C
858    C
859    C
860    C
861    C
862    C
863    C
864    C
865    C
866    C
867    C
868    C
869    C
870    C
871    C
872    C
873    C
874    C
875    C
876    C
877    C
878    C
879    C
880    C
881    C
882    C
883    C
884    C
885    C
886    C
887    C
888    C
889    C
890    C
891    C
892    C
893    C
894    C
895    C
896    C
897    C
898    C
899    C
900    C
901    C
902    C
903    C
904    C
905    C
906    C
907    C
908    C
909    C
910    C
911    C
912    C
913    C
914    C
915    C
916    C
917    C
918    C
919    C
920    C
921    C
922    C
923    C
924    C
925    C
926    C
927    C
928    C
929    C
930    C
931    C
932    C
933    C
934    C
935    C
936    C
937    C
938    C
939    C
940    C
941    C
942    C
943    C
944    C
945    C
946    C
947    C
948    C
949    C
950    C
951    C
952    C
953    C
954    C
955    C
956    C
957    C
958    C
959    C
960    C
961    C
962    C
963    C
964    C
965    C
966    C
967    C
968    C
969    C
970    C
971    C
972    C
973    C
974    C
975    C
976    C
977    C
978    C
979    C
980    C
981    C
982    C
983    C
984    C
985    C
986    C
987    C
988    C
989    C
990    C
991    C
992    C
993    C
994    C
995    C
996    C
997    C
998    C
999    C
1000   C

```

SUBROUTINE PROPER 74/74 OPT=0 ROUND=**/ TRACE FTN 4.6+460 04/07/81 13.10.59 PAGE 2

RETURN
 905 FORMAT(1H1,*, TABLE OF OPEN WATER CURVE FOR *,8A10)
 906 FORMAT(1H0,*,
 902 FORMAT (5X,F10.2,F10.4,F10.5,F10.3)
 END

PROPER
 PROPER
 PROPER
 PROPER
 PROPER

59
 60
 61
 62
 63

60

```

1      SUBROUTINE PROPP(IS,TITLE)
C
C      THIS ENTRY PRINTS ALL DATA IN COMMON BLOCK "CPROP"
C
5      COMMON/CPROP/IPTYP(3),DIAM(3),VADES(3),RPMDES(3),TDES(3),QDES(3),
1      EFFDES(3),OMWTD(3),CAV(3),Z(3),H(3),BAR(3),PD(3),J(3),
2      IERROR(3),KT(3),KQ(3)
C
10     DIMENSION TITLE(8)
C
15     INTEGER Z
C
20     REAL J,KT,KQ
C
25     WRITE(6,700) TITLE
30     WRITE(6,701) VADES(IS),TDES(IS),DIAM(IS),QDES(IS),RPMDES(IS)
1, EFFDES(IS),BAR(IS),KT(IS),CAV(IS),PD(IS),KQ(IS),Z(IS),J(IS),
2 IPTYP(IS),IERROR(IS)
C
35     PRINT ERROR MESSAGE
C
C      IR=IERROR(IS)
C      CALL ERROR(IR)
C      RETURN
C
700  FORMAT(1H1,8A10)
701  FORMAT(12X,31H  PROPULSOR CHARACTERISTICS
1//1X,18HVELOCITY-(KNOTS) -,F9.2,7X,16HTHRUST-(LBS) -,F10.0/1X,
218HDIAMETER-(FEET) -,F9.2,7X,16HTORQUE-(FT-LBS)-,F10.0/1X,
318HRPM -,F8.1/1X,
411HEFFICIENCY-,F7.3,7H BAR-,F6.3,7H KT-,F7.4/1X,
511HCAV OR TIP-,F6.1,8H P/D-,F6.3,7H KQ-,F8.5/1X,
611HNO. BLADES-,13.7X,4HJ -,F6.3//
71X,17HPROPULSOR TYPE -,14
2/ 1X,17HERROR TYPE -,14)
END

```

```

1 SUBROUTINE ERROR(IE)
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```


13.10.59

04/07/81

FTN 4.6+460

74/74 OPT=0 ROUND=*/ TRACE

SUBROUTINE TROOST

```

C
40 CONTINUE
DO 50 I=1,2
  PD(1)=PD(I+1)
  EFF(I)=EFF(I+1)
CONTINUE
PD(3)=PD(2)+DEL PD
IF(PD(3).GT.1.4) GO TO 97
  M=M+1
CALL ROOT(PD(3).Z,BAR,COEFF(IPJ),IPJ,ITQ,KT,KQ,EFFIN,J)
EFF(3)=EFFIN
IF(M.LE.2.OR.EFF(3).GE.EFF(2)) GO TO 40

C
70 ITERATE TO MAXIMUM EFFICIENCY
C THIS METHOD IS REFERENCED IN--
C "INTRODUCTION TO NUMERICAL ANALYSIS" BY F.B. HILDEBRAND
C MC GRAW-HILL 1956, PAGES 50 AND 56.
C
75 PD(4)=PD(2)-DELPD/2.0
PD(5)=PD(2)+DELPD/2.0
CALL ROOT(PD(4).Z,BAR,COEFF(IPJ),IPJ,ITQ,KT,KQ,EFF(4),J)
CALL ROOT(PD(5).Z,BAR,COEFF(IPJ),IPJ,ITQ,KT,KQ,EFF(5),J)
PDNEW=PD(2)-0.0175*(-EFF(1)-0.5*EFF(4)+0.5*EFF(5)+EFF(3))/
  (EFF(1)-0.5*EFF(4)-EFF(2)-0.5*EFF(5)+EFF(3))
IF(PDNEW.LT.PD(1).OR.PDNEW.GT.PD(3)) GO TO 96

C
80 1 FINAL COMPUTED VALUES OF KT AND KQ ARE COMMON WITH
C SUBROUTINE ROOT AND HAVE ALREADY BEEN CALCULATED
C
85 CALL ROOT(PDNEW,Z,BAR,COEFF(IPJ),IPJ,ITQ,KT,KQ,EFFNEW,J)
GO TO 960
CONTINUE

C
90 ITERATION MALFUNCTION WITH PDNEW OUTSIDE THE LIMITS
C OF PD(1) OR PD(3). THE VALUE AT PD(2) HAS BEEN
C USED.
C
95 IERROR = 2
PDNEW=PD(2)
CALL ROOT(PDNEW,Z,BAR,COEFF(IPJ),IPJ,ITQ,KT,KQ,EFFNEW,J)
CONTINUE
GO TO 100
CONTINUE

C
100 AT THIS POINT P/D=1.4 HAS BEEN USED AND THE SEARCH FOR
C THE MAXIMUM HAS BEEN TERMINATED
C
105 IERROR = 1
EFFNEW=EFF(2)
PDNEW=PD(2)
CONTINUE
IF(IPJ.LE.3) NT=VA1-101.268/(D*J)
IF(IPJ.GE.4) D=VA1-1.6878/(N1/60.0)*J)
P=KQ-RHQ*N1-D**5/3600.
WRITE (6,988) VA1,J,PDNEW
GO TO 30
CONTINUE

C
21 CONTINUE

```

```

115 PDX=0.0
    DPD=0.2
    PDY=1.72
    J=VA/(N*D)
    KT=T/(RHO*N*D*D*D)
    XY=CALC(J,BAR,PDY,Z,1)
    WRITE(6,989)
    WRITE(6,990) VA1,J,KT,PDY
    IF(KT.GT.XY) GO TO 28
    PDX=PDY
    CONTINUE
22 PDX=PDY-DPD
    XX=CALC(J,BAR,PDX,Z,1)
    WRITE(6,990) VA1,J,KT,PDY,PDX,XX
    IF(KT.LT.XX) GO TO 22
    CONTINUE
23 DPD=(PDY-PDX)*((KT-XY)/(XY-XX))
    XX=XY
    PDX=PDY
    PDY=PDY+DPD
    XY=CALC(J,BAR,PDY,Z,1)
    WRITE(6,990) VA1,J,KT,PDY,PDX,XX,XY
    IF(ABS(KT-XY).GT.0.0001) GO TO 23
    PDNEW=PDY
    CALL TROST2(KT,KQ,J,PDNEW,BAR,Z,EFFNEW)
    IERROR=0
    GO TO 29
    CONTINUE
28 C
    C
    C
145 AT THIS POINT P/D=1.4 HAS BEEN USED
    PDNEW = 1.4
    IERROR = 1
    CALL TROST2(KT,KQ,J,PDNEW,BAR,Z,EFFNEW)
29 CONTINUE
    P=KQ-RHO*N*D*D*D
    CONTINUE
30 RETURN
    988 FORMAT(1H,* FIRST LOOP - ROOT CALL*,8F10.3)
    989 FORMAT(1H,* SECOND LOOP - CALC CALL*)
    990 FORMAT(20X,6(F10.3,2X))
    END
TROOST 116
TROOST 117
TROOST 118
TROOST 119
TROOST 120
TROOST 121
TROOST 122
TROOST 123
TROOST 124
TROOST 125
TROOST 126
TROOST 127
TROOST 128
TROOST 129
TROOST 130
TROOST 131
TROOST 132
TROOST 133
TROOST 134
TROOST 135
TROOST 136
TROOST 137
TROOST 138
TROOST 139
TROOST 140
TROOST 141
TROOST 142
TROOST 143
TROOST 144
TROOST 145
TROOST 146
TROOST 147
TROOST 148
TROOST 149
TROOST 150
TROOST 151
TROOST 152
TROOST 153
TROOST 154
TROOST 155
TROOST 156
TROOST 157

```

```

1      SUBROUTINE TROST2(KT,KQ,J,POTR2,BAR,Z,EFFTR2)
2      TROST2
3      TROST2
4      TROST2
5      TROST2
6      TROST2
7      TROST2
8      TROST2
9      TROST2
10     TROST2
11     TROST2
12     TROST2
13     TROST2
14     TROST2
15     TROST2

      THIS SUBROUTINE IS USED TO CALCULATE A SINGLE KT, KQ AND
      EFFICIENCY VALUE FOR A GIVEN PROPELLER

      REAL KT,KQ,J
      INTEGER Z

      KT=CALC(J,BAR,POTR2,Z,1)
      KQ=CALC(J,BAR,POTR2,Z,2)
      EFFTR2=KT*J/(KQ*6.2831853)
      RETURN
      END

```


211

13.10.59

04/07/81

FTN 4.6*460

74/74 OPT=0 ROUND=*/ TRACE

SUBROUTINE ROOT

```

515      TQ1=TQ1-COEFF-J2**IPJ
        CONTINUE
        IF (ABS(TQ1)-EPS.LE.0.) GO TO 52
        DE=DEL*TQ1/(TQ2-TQ1)
        IF (ABS(DE).LT.ABS(DEL)) GO TO 518
        DEL=SIGN(DEL,DE)
        GO TO 519
518      CONTINUE
        DEL=DE
519      CONTINUE
        I=I+1
        GO TO 51
52      CONTINUE
        IF (IJK.EQ.2) GO TO 53
        KT=CALC(J2,BAR,PD,Z,1)
        KQ=CALC(J2,BAR,PD,Z,2)
        EFFIN=KT*J2/(KQ-6.2831853)
53      CONTINUE
        RETURN
        END

```

```

ROOT
ROOT
ROOT
ROOT
ROOT
ROOT
ROOT
ROOT
ROOT
ROOT
ROOT
ROOT
ROOT
ROOT
ROOT
ROOT
ROOT
ROOT
ROOT
ROOT

```

```

59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78

```

```

1      C
2      C
3      C
4      C
5      C
6      C
7      C
8      C
9      C
10     C
11     C
12     C
13     C
14     C
15     C
16     C
17     C
18     C
19     C
20     C
21     C
22     C
23     C
24     C
25     C
26     C
27     C
28     C
29     C
30     C
31     C
32     C
33     C
34     C
35     C
36     C
37     C
38     C
39     C
40     C
41     C
42     C
43     C
44     C
45     C
46     C
47     C
48     C
49     C
50     C
51     C
52     C
53     C
54     C
55     C
56     C
57     C
58     C

SUBROUTINE BURIL( T,D,H,VA,N,CAVI,BAR,IERROR,PD,NRET)
THIS SUBROUTINE USES BURRILL'S CAVITATION CRITERIA TO DETERMINE
IF THE BAR USED IS SUFFICIENT AND, IF IT IS NOT, TO ESTIMATE
WHAT BAR IS NECESSARY

COMMON/BARCK/IE

REAL N
NRET = 0
BARC=BARB
BARB=BAR
THRUST=T
AD=0.7853975*D*D
DELT=14.45+0.45*H
QSUBT=(VA/7.12)**2*(N*D/329.0)**2
SIGN=DELT/QSUBT
TAU=0.17096999E-1
1+.23641524*SIGN
1-.52794965E-2*CAVI
1+.78910550E-1*SIGN-CAVI
1+.42061131E-5*SIGN-CAVI**3
1-.98059874E-1*SIGN**2-CAVI
1-.40621894E-2*SIGN**2-CAVI**2
1-.18207486E-7*SIGN**2-CAVI**5
1+.75936652E-2*SIGN**3-CAVI
1+.92023581E-2*SIGN**3-CAVI**2
1-.17649923E-3*SIGN**5-CAVI**2
1-.34804349E-4*SIGN**5-CAVI**4
1+.99981727E-6*SIGN**5-CAVI**5
AP=THRUST/(TAU-QSUBT*144.0)
PAR=AP/AQ
EAR=PAR/(1.067-0.229*PD)
IF(ABS(BAR-EAR).LE.0.0005) GO TO 90
IF((IE.EQ.1.AND.ABS(EAR-BAR).LE.0.01)) GO TO 95
BAR=EAR
IF(BAR.GT.0.95) GO TO 70
IF(BAR.LT.0.4) GO TO 80
NRET = 1
GO TO 95
CONTINUE

70    C
C    C
C    C
EAR IS GREATER THAN 0.95, USE 0.95 FOR EAR

BAR = 0.95
IERROR = 3
NRET = 1
GO TO 95
CONTINUE

80    C
C    C
C    C
EAR IS LESS THAN 0.4 USE 0.4 FOR BAR

BAR=0.4
IERROR = 4
NRET = 1
GO TO 95

```

04/07/81 13.10.59

FTN 4.6+460

OPT=0 ROUND=0/ TRACE

74/74

SUBROUTINE BURIL

BURIL 59
 BURIL 60
 BURIL 61
 BURIL 62
 BURIL 63

90 CONTINUE
 IERROR = 0
 95 CONTINUE
 RETURN
 END

60

SUBROUTINE POLCOF

74/74

OPT=0 ROUND

TRACE

FTN 4.6+460

04/07/81

13.10.59

PAGE

1

1		SUBROUTINE POLCOF(PD,BAR,Z)	2	POLCOF
C			3	POLCOF
C		DETERMINE THE COEFFICIENTS OF THE POLYNOMIALS WHICH APPROXIMATE	4	POLCOF
C		KT AND KQ CURVE	5	POLCOF
5	C	COMMON/OFFDES/PJAY(4)	6	POLCOF
	C		7	POLCOF
	C	COMMON/PCOF/CT(4),CQ(4)	8	POLCOF
	C		9	POLCOF
10	C	REAL KT,KQ,J	10	POLCOF
	C		11	POLCOF
	C	INTEGER Z	12	POLCOF
	C		13	POLCOF
	C		14	POLCOF
15		J=0.0	15	POLCOF
		KT=CALC1(J,BAR,PD,Z,1)	16	POLCOF
		DO 5 I=1,4	17	POLCOF
		CT(I)=PJAY(I)	18	POLCOF
	5	CONTINUE	19	POLCOF
		KQ=CALC1(J,BAR,PD,Z,2)	20	POLCOF
20		DO 7 I=1,4	21	POLCOF
		CQ(I)=PJAY(I)	22	POLCOF
	7	CONTINUE	23	POLCOF
		RETURN	24	POLCOF
		END	25	POLCOF

04/07/81 13.10.59

FTN 4.6+460

74/74 CPT=0 ROUND=// TRACE

SUBROUTINE BESSJ

```

1      SUBROUTINE BESSJ(X,N,VJ)
2
3      C
4      C
5      C
6      C
7      C
8      C
9      C
10     C
11     C
12     C
13     C
14     C
15     C
16     C
17     C
18     C
19     C
20     C
21     C
22     C
23     C
24     C
25     C
26     C
27     C
28     C
29     C
30     C
31     C
32     C
33     C
34     C
35     C
36     C
37     C
38     C
39     C
40     C
41     C
42     C
43     C
44     C
45     C
46     C
47     C
48     C
49     C
50     C
51     C
52     C
53     C
54     C
55     C
56     C
57     C
58     C

```

EVALUATE BSESL FUNCTION FROM ORDER 0 TO ORDER N
 REAL J, JL, JLL, JLLL
 DIMENSION VJ(N)
 DATA PI /3.1415926535898/
 1 EPS /1.0E-30/
 2 EXPON /2.7182818284590/
 DETERMINE THE MAXIMUM N = NU
 D = ALOG(EPS)/X
 C = ALOG(EXPON/2.0)
 B = D-2.0-C
 ALPHA = (-B+SQRT(B**2-4.0*D*(2.0-C)))/(2.0*(2.0-C))
 CLN = ALOG(2.0*PI*X)
 IF(ALPHA.LT.1.5) ALPHA = 1.5
 DO 1000 I=1,20
 ALN = ALOG(ALPHA)
 F = -(CLN+ALN)/(2.0*X)+ALPHA*(C-ALN)-D
 FP = -1.0/(2.0*X+ALPHA)+C-1.0-ALN
 DEL = F/FP
 ALPHA = ALPHA-DEL
 IF(ABS(DEL) .LE. 0.01) GO TO 1100
 1000 CONTINUE
 GO TO 8888B
 1100 CONTINUE
 GNU = ALPHA-X
 NU = IFIX(GNU)+1
 IF(NU.LT.N+10) NU = N+10
 IF((NU/2)*2.NE.NU) NU=NU+1
 RECURSE IN PLACE FOR M.GT.N
 SUM = 0.0
 JLLL = 0.0
 JLL = EPS
 NUM1 = NU-1
 NP = N
 IF((NP/2)*2.EQ.NP) NP = NP-1
 DO 1200 MM = NP,NUM1,2
 M = NUM1-MM+NP
 JL = 2.0*FLOAT(M+2)-JLL/X-JLLL
 J = 2.0*FLOAT(M+1)-JL/X-JLL
 SUM = SUM+2.0*JL
 JLLL = JL
 JLL = J
 1200 CONTINUE
 IF(NP.NE.N) GO TO 1300
 JL = 2.0*FLOAT(N)-JLL/X-JLLL
 SUM = SUM+2.0*JL
 JLLL = JLL
 JLL = JL
 1300 CONTINUE
 I = N+1
 VJ(I) = JLLL

```

60      VJ(N) = JLL
      NP = N-1
      DO 1400 MM = 1,NP
      M = NP-MM
      K = M+1
      VJ(K) = 2.0*FLOAT(K)*VJ(M+2)/X-VJ(M+3)
      IF((M/2)*2.EQ.N) SUM = SUM+2*VJ(M+1)
65      1400 CONTINUE
      SUM = SUM-VJ(1)
      K = N+1
      DO 1500 M = 1,K
      VJ(M) = VJ(M)/SUM
70      1500 CONTINUE
      99999 CONTINUE
      RETURN
      88888 CONTINUE
      WRITE(6,100) X, ALPHA, DEL
75      100 FORMAT(28H NO CONVERGENCE, X, ALPHA, DEL/3E14.7)
      END
      BESSJ 59
      BESSJ 60
      BESSJ 61
      BESSJ 62
      BESSJ 63
      BESSJ 64
      BESSJ 65
      BESSJ 66
      BESSJ 67
      BESSJ 68
      BESSJ 69
      BESSJ 70
      BESSJ 71
      BESSJ 72
      BESSJ 73
      BESSJ 74
      BESSJ 75
      BESSJ 76
      BESSJ 77

```

04/07/81 13.10.59

FTN 4.6+460

74/74 OPT=0 ROUND=+/- TRACE

SUBROUTINE PCHEB

```

1      SUBROUTINE PCHEB (AS, BS, AB, BB, NN, TITLE)
2      PCHEB
3      PCHEB
4      PCHEB
5      PCHEB
6      PCHEB
7      PCHEB
8      PCHEB
9      PCHEB
10     PCHEB
11     PCHEB
12     PCHEB
13     PCHEB
14     PCHEB
15     PCHEB
16     PCHEB
17     PCHEB
18     PCHEB
19     PCHEB
20     PCHEB
21     PCHEB
22     PCHEB
23     PCHEB
24     PCHEB
25     PCHEB
26     PCHEB
27     PCHEB
28     PCHEB
29     PCHEB
30     PCHEB
31     PCHEB
32     PCHEB
33     PCHEB
34     PCHEB
35     PCHEB
36     PCHEB
37     PCHEB
38     PCHEB
39     PCHEB
40     PCHEB
41     PCHEB
42     PCHEB
43     PCHEB
44     PCHEB
45     PCHEB
46     PCHEB
47     PCHEB
48     PCHEB
49     PCHEB
50     PCHEB
51     PCHEB
52     PCHEB
53     PCHEB
54     PCHEB
55     PCHEB
56     PCHEB
57     PCHEB
58     PCHEB

      PLOT BODY SECTIONAL AREA CURVE AND STRUT WATERPLANE OUTLINE CURVE
      INPUTS
      AS AND BS ARE, RESPECTIVELY, THE VECTORS OF SYMMETRIC-
      AND OF ANTISYMMETRIC-CHEBYCHEV COEFFICIENTS FOR THE STRUT.
      AB AND BB ARE, RESPECTIVELY, THE VECTORS OF SYMMETRIC-
      AND OF ANTISYMMETRIC-CHEBYCHEV COEFFICIENTS FOR THE BODY.
      NN = DIMENSION OF AS, BS, AB, BB
      MAXIMUM VALUE OF NN IS NMAX
      NMAX = 10

      INITIALIZE THE GRID GEOMETRY, THE PLOT VARIABLES AND
      THE ABSCISSA VARIABLE.
      COMMON/PLOT/NFIRST,NLAST,NPOINT,XMAX,XMIN,NSCL1,NCHAR,NSCALE(4),
1      PCHEB(2)
      COMMON/XRPLLOT/II,JJ,KK,LL,NHL,NSBH,NVL,NSBV,HCHAR,VCHAR,
1      ISX,ISY,V,H
      LOGICAL NSCL1

      DIMENSION AS(NN), BS(NN), AB(NN), BB(NN), TITLE(8)
      DIMENSION STAT(101),YS(101),YB(101),LABEL(3)

      GENERATE ORDINATE VALUES FOR STRUT AND BODY POINTS
      DO 2000 I=1, 101
      STAT(I)=2.0*(FLOAT(I-1)*.01-.5)
      THETA=ASIN(STAT(I))
      YS(I)=0.0
      YB(I)=0.0
      DO 1000 NCO=1,NN
      UM=COS(FLOAT(2*NCO-1)*THETA)
      VM=SIN(FLOAT(2*NCO)*THETA)
      YS(I)=YS(I)+AS(NCO)*UM+BS(NCO)*VM
      YB(I)=YB(I)+AB(NCO)*UM+BB(NCO)*VM
      1000 CONTINUE
      2000 CONTINUE

      DETERMINE THE MINIMUM AND MAXIMUM ORDINATE VALUES
      YMAX = -9999999.0
      YMIN = +9999999.0
      DO 3000 J=1,101
      IF(YMAX.LT. YB(J)) YMAX=YB(J)
      IF(YMAX.LT. YS(J)) YMAX=YS(J)
      IF(YMIN.GT. YB(J)) YMIN=YB(J)
      IF(YMIN.GT. YS(J)) YMIN=YS(J)
      3000 CONTINUE

      CALL PLOT1 TO SET UP GRID SPACING AND DETERMINE THE AXIS VALUES

```


04/07/81 13.10.59

FTN 4.6+460

74/74 OPT=0 ROUNDO= / TRACE

SUBROUTINE PLOT2

```

1      SUBROUTINE PLOT2(XMAX,XMIN,YMAX,YMIN,NSCL1)
C
C      EXAMINE THE MINIMUM AND MAXIMUM VALUES OF ABSCISSA AND ORDINATE
C      AND ESTABLISH AN INTERNAL FORMULA FOR COMPUTING THE LOCATION
C      IN THE IMAGE REGION CORRESPONDING TO THE POINT TO BE PLOTTED
C
C      REAL LIN
C      INTEGER VCT,VLC,T,HCT,HLCT
C      LOGICAL NSCL1,V,H
C
C      DIMENSION LIN(110)
C
C      COMMON/XPLOTG/XL,XH,YL,YH,XI,YI,XMOV,YMOV
C      COMMON/XPLOTG/GRAF(11,204)
C      COMMON/XPLOTG/II,JJ,KK,LL,NHL,NSBH,NVL,NSBV,HCHAR,VCHAR,IX,IY,V,H
C
C      IF(NSCL1) GOTO 1
C      II=0
C      JJ=3
C      KK=0
C      LL=3
C      NHL=6
C      NVL=11
C      NSBH=9
C      NSBV=9
C      HCHAR=1L-
C      VCHAR=1LI
C
C      1 CONTINUE
C      HLCT=0
C      HCT=NSBH
C      DO 2 I=1,204
C      IF(HLCT.EQ.NHL) GOTO 3
C      IY=I
C      IX=J
C      VLC=0
C      VCT=NSBV
C      DO 4 J=1,110
C      IF(VLCT.EQ.NVL) GO TO 5
C      IX=J
C      IF(VCT.EQ.NSBV) GOTO 6
C      LIN(J)=1L
C      GOTO 40
C      6 CONTINUE
C      LIN(J)=VCHAR
C      VCT=-1
C      VLC=VLC+1
C      40 CONTINUE
C      VCT=VCT+1
C      4 CONTINUE
C      5 CONTINUE
C      VLC=0
C      IF(HCT.NE.NSBH) GOTO 7
C      HCT=-1
C      HLCT=HLCT+1
C      DO 8 J=1,IX
C      LIN(J)=HCHAR
C      8 CONTINUE

```

04/07/81 13.10.59

FTN 4.6+460

SUBROUTINE PLOT2 74/74 OPT=0 ROUND=0/ TRACE

```

60      7  CONTINUE
          HCT=HCT+1
          IQ=IX+1
          DO 10 J=IQ,110
              LIN(J)=1L
          10 CONTINUE
          ENCODE(110,9,GRAF(1,1)) (LIN(J),J=1,110)
65      2  CONTINUE
          3  CONTINUE
              XL=XMIN
              XH=XMAX
              YL=YMIN
              YH=YMAX
              XI=(XH-XL)/FLOAT(IX-1)
              YI=(YH-YL)/FLOAT(IY-1)
              XMOV=1.0-XL/XI
              YMOV=1.0-YL/YI
              V=.TRUE.
              H=.TRUE.
              RETURN
          9  FORMAT(110A1)
              END
70
75

```

```

PLOT2 59
PLOT2 60
PLOT2 61
PLOT2 62
PLOT2 63
PLOT2 64
PLOT2 65
PLOT2 66
PLOT2 67
PLOT2 68
PLOT2 69
PLOT2 70
PLOT2 71
PLOT2 72
PLOT2 73
PLOT2 74
PLOT2 75
PLOT2 76
PLOT2 77
PLOT2 78
PLOT2 79
PLOT2 80

```



```

1      C      SUBROUTINE PLOT4(MCHAR,NCHAR)
2      C
3      C      PRINT THE IMAGE OF THE COMPLETED GRAPH ON THE PRINTER,
4      C      INCLUDING THE VALUES OF THE ABSCISSA AND ORDINATE AT THE
5      C      GRID LINES OUTSIDE THE BOTTOM AND LEFT EDGES OF THE GRAPH
6      C
7      C
8      C
9      C      INTEGER SA, SB, SC
10     C      INTEGER CT,PDQ
11     C      LOGICAL V,M
12     C
13     C      DIMENSION HNUM(15), MCHAR(3), LCHAR(30), VGFM(3), HLFMT(3)
14     C
15     C      COMMON/XRPLTQ/XL,XH,YL,YH,XI,YI,XMOV,YMOV
16     C      COMMON/XRPLTG/GRAF(11,204)
17     C      COMMON/XRPLTQ/I,J,K,L,NHL,NSBH,NVL,NSBV,HCHAR,VCHAR,ISX,ISY,V,H
18     C
19     C      SA=NSBV
20     C      SC=NCHAR
21     C      SB=NVL
22     C      KQ=1
23     C      CALL QPLOT25(PDQ)
24     C      ENCODE(26,2,VGFM(1))J
25     C      IF(NSBV.LE.0) NSBV=10
26     C
27     C      4 CONTINUE
28     C      IF(NSBV.GT. 10) GO TO 3
29     C      NSBV=NSBV+2
30     C      NVL=NVL
31     C      NVL=XVL/2.0+.5
32     C      KO=KQ+1
33     C      GO TO 4
34     C
35     C      3 CONTINUE
36     C      NNN=NVL-1
37     C      MMM=NSBV-8
38     C      ENCODE(25,5,HLFMT(1)) PDQ,L,NNN,MMM,L
39     C      IF (NCHAR.NE.0) GOTO 14
40     C      NCHAR=1
41     C      LCHAR(1)=1L
42     C      GO TO 15
43     C
44     C      14 CONTINUE
45     C      DECODE(10,17,MCHAR(1)) (LCHAR(IJ),IJ=1,10)
46     C      DECODE(10,17,MCHAR(2)) (LCHAR(IJ),IJ=11,20)
47     C      DECODE(10,17,MCHAR(3)) (LCHAR(IJ),IJ=21,30)
48     C
49     C      15 CONTINUE
50     C      MCT=0
51     C      NSPC=(ISY-NCHAR)/2
52     C      LCT=0
53     C      CT=NSBH
54     C      DO 6 MMNX=1,ISY
55     C      N=ISY-MMNX + 1
56     C      IF (MMNX.LE.NSPC.OR.LCT.GT.NCHAR) GO TO 7
57     C      LCT=LCT+1
58     C      LCR=LCHAR(LCT)
59     C      GO TO 8
60     C
61     C      7 CONTINUE
62     C      LCR=1L
63     C
64     C      8 CONTINUE

```

13.10.59

04/07/81

FTN 4.6+460

74/74 OPT=0 ROUND=0/ TRACE

SUBROUTINE PLOT4

```

60      IF (CT.EQ.NSBH.AND.H) GO TO 9
        PRINT 1,LCR,(GRAF(I,N),I=1,11)
        GO TO 13
9        CONTINUE
        CT=1
        QNN=(YH-MCT+YI)*10.**I
        PRINT VGFMT,LCR,QNN,(GRAF(I,N),I=1,11)
        MCT=MCT+NSBH+1
13       CONTINUE
        CT=CT+1
6        CONTINUE
        MCT=0
        IF(.NOT.V) GO TO 12
        DO 10 N=1,NVL
            HNUM(N)=(XL+MCT+XI)*10.**K
            MCT=MCT+NSBV+KQ
10        CONTINUE
        PRINT 11
        PRINT HL,FMT,(HNUM(N),N=1,NVL)
12       CONTINUE
        NSBV=SA
        NVL=SB
        NCHAR=SC
        RETURN
1       FORM=I(1H,A1,16X,11A10)
2       FORMAT(15H(1H,A1,3X,F10.,I1,10H,3X,11A10))
5       FORMAT(1H(,12,6HX,F10.,I1,1H,I2,1H(,12,6HX,F10.,I1,2H)))
11      FORMAT(1H0)
17      FORMAT(10A1)
        END
59      PLOT4
60      PLOT4
61      PLOT4
62      PLOT4
63      PLOT4
64      PLOT4
65      PLOT4
66      PLOT4
67      PLOT4
68      PLOT4
69      PLOT4
70      PLOT4
71      PLOT4
72      PLOT4
73      PLOT4
74      PLOT4
75      PLOT4
76      PLOT4
77      PLOT4
78      PLOT4
79      PLOT4
80      PLOT4
81      PLOT4
82      PLOT4
83      PLOT4
84      PLOT4
85      PLOT4
86      PLOT4
87      PLOT4
88      PLOT4

```

```

1      C      SUBROUTINE QPLOTZS(PDQ)
          INTEGER A,B,C,D,PDQ
          LOGICAL V,H
5      C      COMMON/XRPLDTF/XL,XH,YL,YH,XI,YI,XM,YM
          COMMON/XRPLDTQ/II,JJ,KK,LL,A,B,C,D,E,F,M,N,V,H
          C
          X=ABS(XL)
          IF(X.LT.XH) X=XH
          K=0
          DO 1 I=1,11
              XU=10**((I-1)
              IF(X.GT.XU) GOTO 2
              K=K+1
          1 CONTINUE
          2 CONTINUE
          PDQ=1
          IF(LL.EQ.0) PDQ=0
          QQ=PDQ+KK+LL+K
          PDQ=14.0-(10.0-QQ)/2.0
          RETURN
          END
20

```

```

1      C
      C
      C
      FUNCTION FORMDR(VL)
      EVALUATE FORM DRAG COEFFICIENT
      DIMENSION X(9), Y(9)
      DATA X / 1.325, 1.38, 1.43, 1.48, 1.55, 1.68, 1.9, 2.2, 2.5 /
      DATA Y / .51, .71, .84, .85, .83, .75, .65, .57, .50 /
      IF (VL.GE.1.325.AND.VL.LT.2.5) GO TO 4
      FORMDR=0.5
      GO TO 5
      4 FORMDR = YINTP (VL,X,Y,9)
      5 RETURN
      END
10
2      FORMDR
3      FORMDR
4      FORMDR
5      FORMDR
6      FORMDR
7      FORMDR
8      FORMDR
9      FORMDR
10     FORMDR
11     FORMDR
12     FORMDR
13     FORMDR
14     FORMDR

```

```

1      C
      C
      C
      FUNCTION YINTP (XA,X,Y,N)
      INTERPOLATING THROUGH A SET OF DISCRETE DATA
      DIMENSION X(1), Y(1)
      1 DO 10 I=1,N
      IF (X(I)-XA) 10,10,2
      2 IN=I-2
      IF(IN) 4,4,6
      4 IN=1
      GO TO 12
      6 NN=N-3
      IF (IN-NN) 12,12,8
      8 IN=NN
      GO TO 12
      10 CONTINUE
      12 IO=IN+3
      YINTP=0.
      DO 20 I=IN,IO
      PROD=Y(I)
      DO 16 J=IN,IO
      IF (I-J) 15,16,15
      15 PROD=PROD*(XA-X(J))/(X(I)-X(J))
      16 CONTINUE
      20 YINTP=YINTP+PROD
      21 RETURN
      END

```

DTNSRDC ISSUES THREE TYPES OF REPORTS

1. DTNSRDC REPORTS, A FORMAL SERIES, CONTAIN INFORMATION OF PERMANENT TECHNICAL VALUE. THEY CARRY A CONSECUTIVE NUMERICAL IDENTIFICATION REGARDLESS OF THEIR CLASSIFICATION OR THE ORIGINATING DEPARTMENT.

2. DEPARTMENTAL REPORTS, A SEMIFORMAL SERIES, CONTAIN INFORMATION OF A PRELIMINARY, TEMPORARY, OR PROPRIETARY NATURE OR OF LIMITED INTEREST OR SIGNIFICANCE. THEY CARRY A DEPARTMENTAL ALPHANUMERICAL IDENTIFICATION.

3. TECHNICAL MEMORANDA, AN INFORMAL SERIES, CONTAIN TECHNICAL DOCUMENTATION OF LIMITED USE AND INTEREST. THEY ARE PRIMARILY WORKING PAPERS INTENDED FOR INTERNAL USE. THEY CARRY AN IDENTIFYING NUMBER WHICH INDICATES THEIR TYPE AND THE NUMERICAL CODE OF THE ORIGINATING DEPARTMENT. ANY DISTRIBUTION OUTSIDE DTNSRDC MUST BE APPROVED BY THE HEAD OF THE ORIGINATING DEPARTMENT ON A CASE-BY-CASE BASIS.

