No. 229
October 1980

Added Mass and Damping of Vibrating Propellers

Michael G. Parsons
William S. Vorus
Edward M. Richard

This research was sponsored by the
Department of Commerce, Maritime Administration
University Research Program, Contract No. MA-3-79-SAC-B0012

Department of Naval Architecture
and Marine Engineering
College of Engineering
The University of Michigan
Ann Arbor, Michigan 48109

50272-101

REPORT DOCUMENTATION PAGE	1. REPORT NO. MA-RD-940-81037	2.	3. Recipient's Accession No.

4. Title and Subtitle

ADDED MASS AND DAMPING OF VIBRATING PROPELLERS

5. Report Date: October 1980

6.

7. Author(s)

Michael Parsons, William Vorus, Edward Richard

8. Performing Organization Rept. No. No. 229

9. Performing Organization Name and Address

Department of Naval Architecture and Marine Engineering
College of Engineering
The University of Michigan
Ann Arbor, Michigan 48109

10. Project/Task/Work Unit No.

11. Contract(C) or Grant(G) No.
(C) MA-79-SAC-B0012
(G)

12. Sponsoring Organization Name and Address

Office of Maritime Technology
Maritime Administration
U. S. Department of Commerce
Washington, D.C. 20230

13. Type of Report & Period Covered: Final

14.

15. Supplementary Notes

This research was conducted under the Maritime Administration University Research Program - 1978.

16. Abstract (Limit: 200 words)

A general analysis of the added mass and damping of a marine propeller in all modes of rigid body vibration is presented. This derivation assumes that the pressure distribution on a propeller blade due to a unit heave oscillation or unit pitch oscillation about the midchord point can be obtained by two-dimensional thin foil theory, lifting-line theory, or lifting-surface theory. These results show that eighteen coefficients are needed to completely characterize the added mass and damping of a propeller. The use of two-dimensional thin foil theory and lifting-line theory to obtain the pressure distribution on a propeller blade due to a unit heave oscillation or unit pitch oscillation are reviewed. These results have been utilized in the PRAMAD computer program which calculates the added mass and damping of a vibrating propeller. This program incorporates approximate lifting-surface corrections which were obtained by regressing the ratio of lifting-surface and lifting-line results for a matrix of four-bladed Wageningen B-Series propellers. Results are presented for sensitivity studies performed using the PRAMAD program to show the effects of advance coefficient, vibration frequency, blade number, and skew on propeller added mass and damping. Preliminary design equations are presented for the estimation of all eighteen added mass and damping coefficients of 4-, 5-, 6-, and 7-bladed Wageningen B-Series propellers.

17. Document Analysis a. Descriptors

Propeller Vibration
Propeller Added Mass
Propeller Damping

b. Identifiers/Open-Ended Terms

c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report)	21. No. of Pages
Approved for Release National Technical Information Service Springfield, VA 22151	Unclassified	132
	20. Security Class (This Page) Unclassified	22. Price

(See ANSI-Z39.18)

See Instructions on Reverse

OPTIONAL FORM 272 (4-77)
(Formerly NTIS-35)
Department of Commerce

ABSTRACT

A general analysis of the added mass and damping of a marine propeller in all modes of rigid body vibration is presented. This derivation assumes that the pressure distribution on a propeller blade due to a unit heave oscillation or unit pitch oscillation about the midchord point can be obtained by two-dimensional thin foil theory, lifting-line theory, or lifting-surface theory. These results show that eighteen coefficients are needed to completely characterize the added mass and damping of a propeller. The use of two-dimensional thin foil theory and lifting-line theory to obtain the pressure distribution on a propeller blade due to a unit heave oscillation or unit pitch oscillation are reviewed. These results have been utilized in the PRAMAD computer program which calculates the added mass and damping of a vibrating propeller. This program incorporates approximate lifting-surface corrections which were obtained by regressing the ratio of lifting-surface and lifting-line results for a matrix of four-bladed Wageningen B-Series propellers. Results are presented for sensitivity studies performed using the PRAMAD program to show the effects of advance coefficient, vibration frequency, blade number, and skew on propeller added mass and damping. Preliminary design equations are presented for the estimation of all eighteen added mass and damping coefficients of 4-, 5-, 6-, and 7-bladed Wageningen B-Series propellers.

Table of Content

	page
Abstract	i
Table of Content	ii
List of Figures	iii
List of Tables	iv
1. Introduction	1
2. Derivation of Propeller Added Mass and Damping	6
2.1 General Approach	6
2.2 Coordinate System Definitions	8
2.3 Heave and Pitch Due to Oscillations in Overall Coordinates	11
2.4 Forces due to Unit Oscillations	14
3. Pressure Distributions Due to Unit Oscillations	22
3.1 Two-Dimensional Thin Foil Theory	22
3.2 Lifting-Line	27
3.3 Lifting-Surface	31
4. The PRAMAD Computer Program Results	32
4.1 Comparison of Two-Dimensional, Lifting-Line, and Lifting-Surface Results	32
4.2 Sensitivity Studies	38
5. Lifting-Surface Corrections	44
5.1 Definition of Corrections	44
5.2 Regression Analyses for Corrections	47
6. Preliminary Design Equations for Wageningen B-Series Propellers	52
7. Design Example	58
8. Closure	62
9. References	63
Appendices:	
A. User's Instruction for PRAMAD	A-1
B. Programmer's Documentation for PRAMAD	B-1
C. Listing of PRAMAD	C-1

List of Figures

Figure No.		page
1.	Nomenclature for Propulsion Shafting Vibration	2
2.	Coordinate System in Projected Plane of Propeller	9
3.	Coordinate System in Local Plane of Propeller Blade	10
4.	Coordinate System for Two Dimensional Thin Foil	23
5.	Lifting-Line Propeller Model	27
6.	Typical Pressure Distribution due to a Unit Heave	36
7.	Typical Pressure Distribution due to a Unit Pitch	36
A-1	Blade Segmentation for Data Input	A-6
B-1	PRAMAD Macro Flowchart	B-2

List of Tables

Table No.		page
1.	C Coefficients for Blade Normal Displacement Eq. (29)	12
2.	D Coefficients for Blade Local Pitch Eq. (33)	14
3.	Complex Hydrodynamic Force Integrals	22
4.	Nondimensionalization of Coefficients	32
5.	Comparison of Results for B4-70-80 Propeller ($J_{\ell\ell}$, $\omega = 4\Omega$)	34
6.	Comparison of Quasi-Steady and Lifting-Surface Torsional/Axial Damping	37
7.	Effect of Advance Coefficient J on Results for B4-75-90 Propeller ($\omega = 4\Omega$)	39
8.	Effect of Vibration Frequency on Results for B4-75-90 Propeller (.75 $J_{\ell\ell}$)	41
9.	Effect of Number of Blades Z on Results for BZ-75-90 Propellers (.75 $J_{\ell\ell}$, blade rate $\omega = Z\Omega$)	42
10.	Effect of Linear Skew Distribution with Tip Skew θ_s on Results for B4-75-90 Propeller (.75 $J_{\ell\ell}$, $\omega = 4\Omega$)	43
11.	Ratio of Lifting-Surface to Lifting-Line Results used in Regression Analyses for Lifting-Surface Corrections	46
12.	Extrapolated Data used in Regression Analyses for Lifting-Surface Corrections	48
13.	Regression Data for Lifting-Surface Correction Equations	51
14.	Regression Equation Coefficients for B4 Propellers	54
15.	Regression Equation Coefficients for B5 Propellers	55
16.	Regression Equation Coefficients for B6 Propellers	56
17.	Regression Equation Coefficients for B7 Propellers	57
18.	Characteristics of DTNSRDC Propeller 4381 Example	59
19.	Results for DTNSRDC Propeller 4381	60
A-1	Data Set Definition for PRAMAD	A-3
A-2	Suggested Time Limits for PRAMAD	A-5

1. Introduction

The increase in the installed power in commercial ships in recent years has made propulsion shafting vibration one of the most important problems in marine engineering. It is now very important, if not mandatory[1,2], to be able to analyze shafting systems in torsional, axial, and lateral vibration in preliminary as well as detailed design. In the past 20 years, the growth of computers has made practical a complete analysis of marine propulsion shafting vibration. Needed numerical techniques have also been developed during this period. Research effort in this country has significantly improved our ability to establish the propeller excitation of the propulsion shafting.[3,4,5] No parallel effort has been underway during this period, however, to provide comparable techniques and data for the estimation of the propeller added mass and damping. These data are also needed to perform the propulsion shafting analyses. The collective term "added mass and damping" is used here to include the coefficients which characterize all the hydrodynamic forces and moments which act on a propeller due to its translational and rotational vibration in a fluid.

In general, a marine propeller vibrates in the six rigid body modes with displacements δ_i and rotations θ_i as defined in Fig. 1. The propeller will be assumed to vibrate as a rigid body in this discussion. The propeller operates in a *circumferentially varying* wake field and thus generates a vibratory component of lift which produces vibratory excitation forces F_i in the three coordinate directions and vibratory excitation moments Q_i about the three coordinate axes. The force F_x directly excites axial or longitudinal vibration. The moment Q_x directly excites torsional vibration. The forces F_y and F_z and the moments Q_y and Q_z excite lateral vibrations. As the propeller vibrates in water, it experiences additional hydrodynamic forces f_i and moments q_i (added mass and damping) due to its oscillatory motion. In linear foil theory these can be viewed as the forces and moments

generated by the propeller's oscillatory motion in a *uniform* wake field.

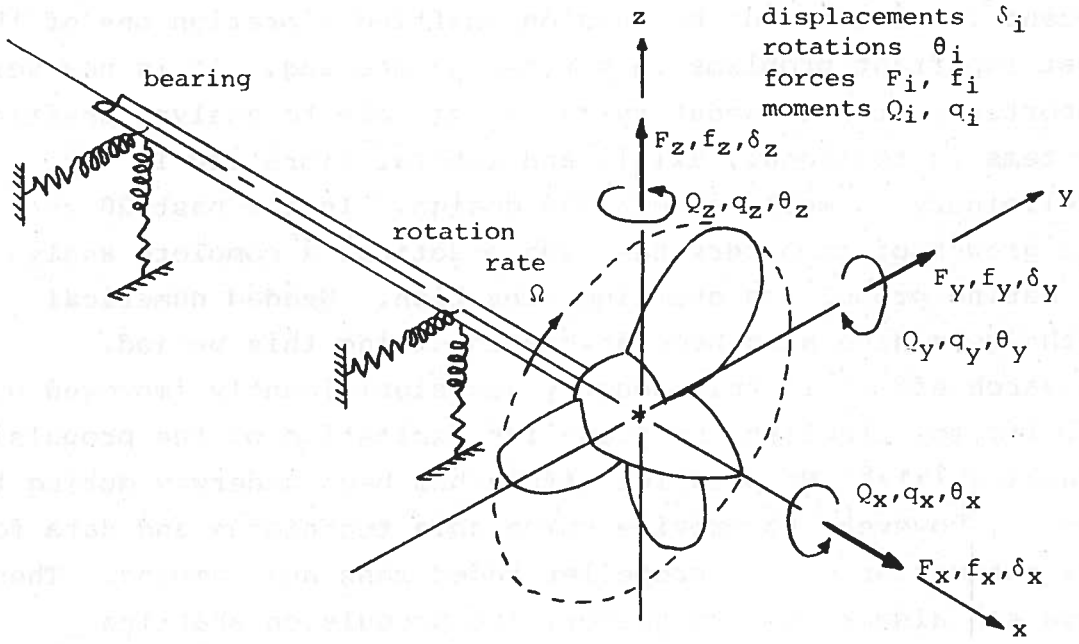


Figure 1. Nomenclature for Propulsion Shafting Vibration

The equations of motion for the vibrating propeller can be written as follows in matrix form:

$$\underline{M}\ddot{\underline{x}} = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & J_{xx} & 0 & 0 \\ 0 & 0 & 0 & 0 & J_{yy} & 0 \\ 0 & 0 & 0 & 0 & 0 & J_{zz} \end{bmatrix} \begin{bmatrix} \ddot{\delta}_x \\ \ddot{\delta}_y \\ \ddot{\delta}_z \\ \ddot{\theta}_x \\ \ddot{\theta}_y \\ \ddot{\theta}_z \end{bmatrix} = \underline{f}_e + \underline{f}_h + \underline{f}_s \ , \quad (1)$$

where m is the propeller mass, J_{xx} is the mass moment of inertia of the propeller about its axis of rotation and $J_{yy}=J_{zz}$ are diametral mass moments of inertia. The displacement vector $\underline{x}$, vibratory excitation vector $\underline{f}_e$, and additional hydrodynamic force vector $\underline{f}_h$ are defined as follows:

$$\underline{x} = [\delta_x, \delta_y, \delta_z, \theta_x, \theta_y, \theta_z]^T , \tag{2}$$

$$\underline{f}_e = [F_x, F_y, F_z, Q_x, Q_y, Q_z]^T , \tag{3}$$

$$\underline{f}_h = [f_x, f_y, f_z, q_x, q_y, q_z]^T . \tag{4}$$

The vector $\underline{f}_s$ includes the forces and moments exerted on the propeller by the shafting and couples the propeller motion with that of the rest of the propulsion shafting system. The additional hydrodynamic forces $\underline{f}_h$ depends on the displacement, velocity, and acceleration of the propeller and can be represented as follows:

$$\underline{f}_h = -M_a\ddot{\underline{x}} - C_p\dot{\underline{x}} - K_p\underline{x} . \tag{5}$$

The matrix K_p will be zero for a fully-immersed propeller leaving,

$$\underline{f}_h = -M_a\ddot{\underline{x}} - C_p\dot{\underline{x}} . \tag{6}$$

Substituting eq. (6) into eq. (1) and rearranging yields the traditional form of the equations of motion; i.e.,

$$(M + M_a)\ddot{\underline{x}} + C_p\dot{\underline{x}} - \underline{f}_s = \underline{f}_e . \tag{7}$$

The matrix M_a is the added mass matrix for the propeller. The first three terms on the main diagonal are the added mass in the three coordinate directions, the second three terms on the main diagonal are the added mass moments of inertia, and the off-diagonal terms are usually called inertia coupling. The matrix C_p is the damping matrix for the propeller. The first three terms on the main diagonal are the linear damping, the second three are the rotational damping and the off-diagonal terms are usually called velocity coupling. Notice that negative signs are included in the definition eq. (6) so that the added mass and damping matrices appear with plus signs on the left side of the equations of motion, eq. (7), where they are normally used.

The added mass and damping matrices contain a total of 72 coefficients. Only eighteen non-zero coefficients, however, are

actually needed to fully characterize the added mass and damping properties of a marine propeller. We will show below that these matrices have the following form:

$$M_a = \begin{bmatrix} m_{11} & 0 & 0 & m_{41} & 0 & 0 \\ 0 & m_{22} & -m_{32} & 0 & m_{52} & -m_{62} \\ 0 & m_{32} & m_{22} & 0 & m_{62} & m_{52} \\ m_{41} & 0 & 0 & m_{44} & 0 & 0 \\ 0 & m_{52} & -m_{62} & 0 & m_{55} & -m_{65} \\ 0 & m_{62} & m_{52} & 0 & m_{65} & m_{55} \end{bmatrix} ; \quad C_p = \begin{bmatrix} c_{11} & 0 & 0 & c_{41} & 0 & 0 \\ 0 & c_{22} & -c_{32} & 0 & c_{52} & -c_{62} \\ 0 & c_{32} & c_{22} & 0 & c_{62} & c_{52} \\ c_{41} & 0 & 0 & c_{44} & 0 & 0 \\ 0 & c_{52} & -c_{62} & 0 & c_{55} & -c_{65} \\ 0 & c_{62} & c_{52} & 0 & c_{65} & c_{55} \end{bmatrix} . \tag{8}$$

The two matrices have an identical form. The zeros in the first and fourth rows and columns indicate that the added mass and damping introduce no coupling between the torsional/axial motion (coordinates 1 and 4) and the lateral motion (coordinates 2, 3, 5, and 6). The symmetry of the propeller in the lateral plane results in the equality of selected terms in the lateral coordinates; e.g. $m_{22} = m_{33}$, $m_{55} = m_{66}$, and $m_{52} = m_{63}$. A reciprocity relationship results in the equality of other terms; e.g. $m_{14} = m_{41}$ and $m_{25} = m_{52}$. Note that the matrices are symmetrical *except* for four sign changes. These sign changes result from the handedness of the propeller; i.e., for a right-hand turning propeller the +y-axis immediately follows the +z-axis as the propeller rotates but not vice versa. Six added mass and damping terms are needed for the coupled torsional/axial motion; i.e., m_{11} , m_{44} , m_{41} , c_{11} , c_{44} , and c_{41} . Twelve terms are needed for lateral motion. These can be separated into one group with the forces and moments in the lateral directions which are the same as the direction of the motion; i.e., m_{22} , m_{55} , m_{52} , c_{22} , c_{55} , and c_{52} . These we term the coefficients for *forces parallel to the motion*. A second group has forces and moments in the lateral directions which are normal to the motion; i.e., m_{32} , m_{65} , m_{62} , c_{32} , c_{65} , and c_{62} . These we term the coefficients for *forces perpendicular to the motion*. This nomenclature follows that introduced recently by Hylarides and van Gent.[6]

In free vibration analyses, which are performed to establish the natural frequencies of the propulsion shafting system, damping and velocity coupling are usually neglected. Data is needed for the added mass and inertia coupling but the latter is also commonly neglected. In forced vibration analyses, which are performed to establish the magnitude of vibration, both the added mass and damping are required. Limited data exists for propeller added mass in torsional and axial vibration[7,8,9,10]; even less exists for lateral vibration[8,11,12,13]. The quasi-steady approach commonly used in the United States to estimate propeller damping in torsional and axial motion can lead to serious errors;[14,15,16] essentially no data exists for propeller damping in lateral vibration. The notable exception to this general situation is the recent paper by Hylarides and van Gent[6] which gave complete added mass and damping results for a matrix of nine 4-bladed Wageningen B-Series propellers. The present paper presents new results on the fundamental character of propeller added mass and damping, provides a practical technique for obtaining the added mass and damping of a propeller in detailed design, and presents design equations which can be used to estimate the added mass and damping early in preliminary design. We feel these results will permit a major improvement in the state-of-the-art of marine propulsion shafting vibration analyses.

2. Derivation of Propeller Added Mass and Damping

In this section, we present a general analysis of the added mass and damping of a marine propeller. Here it is assumed that the pressure distribution on a propeller blade due to a unit local pitch and a unit local heave oscillation at a specified frequency are known. Techniques for obtaining these pressure distributions will be discussed separately in Section 3.

2.1. General Approach

Equation (6) states that the added mass and damping forces and moments on a propeller are the result of the velocity and acceleration of its vibratory motion in a uniform wake field. Suppose we subject a propeller to a general harmonic oscillation at frequency ω, which can be expressed in complex form as,

$$\underline{x} = \mathrm{Re}\{\underline{X}e^{i\omega t}\} \quad , \tag{9}$$

and then calculate the resulting vector of six hydrodynamic forces and moments on the propeller, $\underline{f}_h$. The foil behavior is linear so the resulting hydrodynamic force will be harmonic at the same frequency; i.e.,

$$\underline{f}_h = \mathrm{Re}\{\underline{F}_h e^{i\omega t}\} \quad , \tag{10}$$

where the complex magnitude vector $\underline{F}_h$ reflects the phase shifts which exist between the excitation and response. The displacement magnitude vector $\underline{X}$ would also be complex if there were phase shifts in the various components of the displacement with respect to the zero-time reference point. Equation (9) can be differentiated to obtain the velocity and acceleration and the results can be substituted into eq. (6) to yield,

$$\underline{f}_h = \mathrm{Re}\{(\omega^2 M_a - i\omega C_p)\underline{X}e^{i\omega t}\} \quad . \tag{11}$$

Comparing eq. (10) and eq. (11), the force magnitude vector $\underline{F}_h$ is given by,

$$\underline{F}_h = F\underline{X} = (\omega^2 M_a - i\omega C_p)\underline{X} \quad , \tag{12}$$

where F is a matrix with complex elements.

Now if we subject the propeller to a *unit* oscillation at frequency ω in a single coordinate ℓ, the displacement magnitude vector $\underline{X}$ is a unit vector in coordinate ℓ and the force magnitude vector becomes column ℓ of matrix F; i.e.,

$$\underline{F}^*_{h_\ell} = F \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \text{row } \ell \ , \tag{13}$$

where the asterisk denotes a unit oscillation. The resulting force or moment in coordinate (row) j in $\underline{F}^*_{h_\ell}$ is then element (j,ℓ) of the matrix F. We therefore have from eq. (12),

$$F^*_{h_{j\ell}} = f_{j\ell} = \omega^2 m_{j\ell} - i\omega c_{j\ell} \ , \tag{14}$$

giving,

$$m_{j\ell} = \frac{1}{\omega^2} \mathrm{Re}\{F^*_{h_{j\ell}}\} \ , \tag{15}$$

and,

$$c_{j\ell} = -\frac{1}{\omega} \mathrm{Im}\{F^*_{h_{j\ell}}\} \ . \tag{16}$$

These state that if we (1) subject the propeller to a unit oscillation at frequency ω in coordinate ℓ, (2) calculate the resulting force or moment in coordinate j, and (3) designate this complex magnitude as $F^*_{h_{j\ell}}$, the elements (j,ℓ) of the propeller added mass and damping matrices can be calculated by eqs. (15) and (16), respectively. This is the technique used in the following. The process must be performed once for each of the nine pairs of elements (j,ℓ) needed in eq. (8).

2.2. Coordinate System Definitions

In this analysis, we use three separate coordinate systems. The first is the *overall* coordinate system defined in Fig. 1 for a right-hand propeller. The displacement vector in these coordinates is given by eq. (2) and component ℓ of this vector can be denoted x_ℓ. The second coordinate system of interest is in the *projected plane of the propeller* as shown in Fig. 2. The coordinates are aft (unit vector $\underline{i}$) along the x-axis in Fig. 1, radially outward ($\underline{e}_r$), and tangential ($\underline{e}_\theta$) directed opposite to the direction of rotation. The Z blades are indexed by the integer k=0,1,...,Z-1; the blade spacing is $2\pi/Z$. The position vector to any point on a particular blade k is defined by the coordinates (x_1, r_1, θ_1) where the angular position in the projected plane θ_1 is given by,

$$\theta_1 = \theta + \frac{2\pi k}{Z} + \theta_s(r_1) + \alpha' \quad , \tag{17}$$

where the first component,

$$\theta = -\Omega t \quad , \tag{18}$$

is due to the rotation of the propeller. Angle θ is defined to the generator line (radial reference line through the midchord point at the hub plus rake in x-direction) for the index (k=0) blade. The first two terms in eq. (17) give the angular position of the generator line for blade k. The projected skew angle θ_s defines the angular displacement of the midchord point at radius r_1 from the generator line. Coordinate α' is a local angular coordinate in the projected plane of the blade. It is negative at the leading edge, zero at midchord, and positive at the trailing edge.

The third coordinate system of interest is a local coordinate system centered at the midchord point of the propeller blade section at radius r_1 as shown in Fig. 3. Coordinate n is normal to the blade and generally directed aft (positive on the blade face). Coordinate m is along the blade

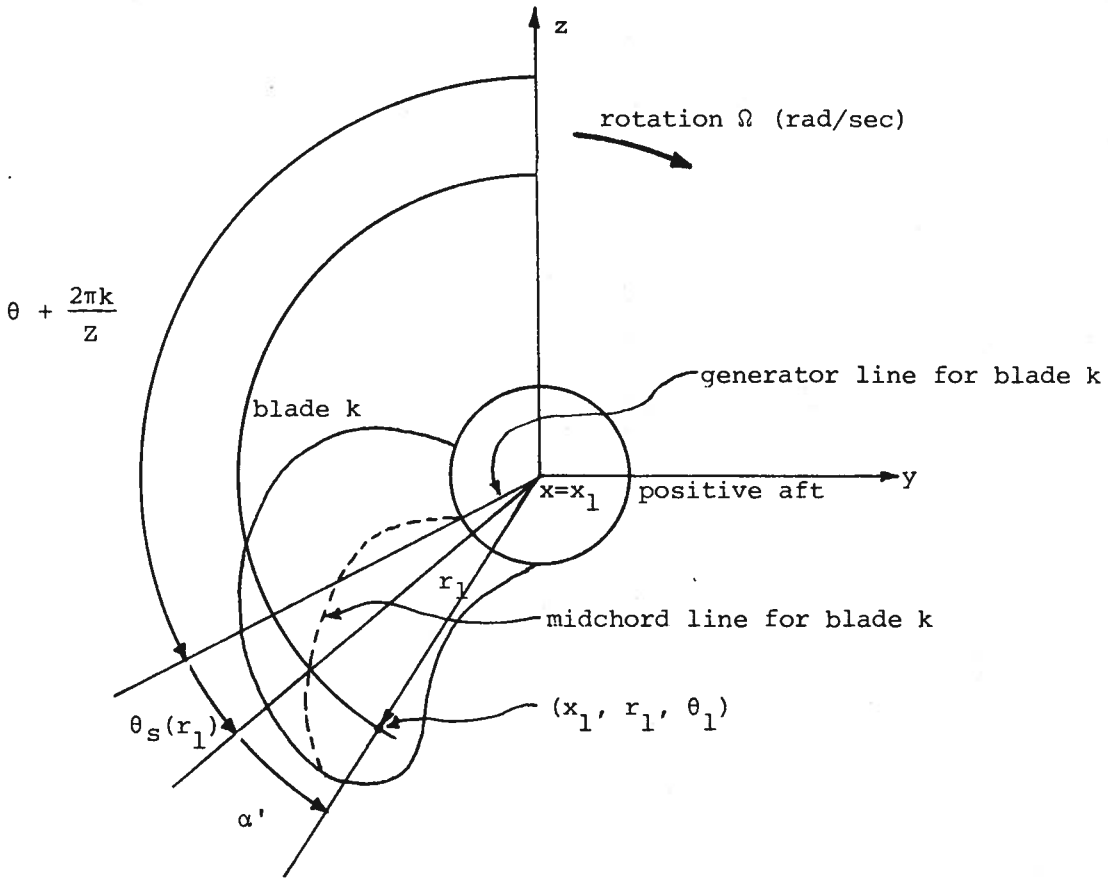


Figure 2. Coordinate System in Projected Plane of Propeller.

chord; positive toward the trailing edge. Coordinate r is radially outward ($\underline{e}_r$) as in the projected plane. These local coordinates are oriented at the geometric pitch angle β_g, with respect to the transverse (y-z) plane of the propeller. The chordwise linear coordinate m is related to the local angular coordinate in the projected plane α' by,

$$m = \frac{r_1 \alpha'}{\cos\beta_g} \quad . \tag{19}$$

The axial position in the overall coordinate system, x, and in the projected-plane coordinates, x_1, are similarly given by,

$$x = x_1(r_1,\alpha') = r_1\theta_r(r_1) + r_1(\theta_s(r_1)+\alpha')\tan\beta_g \quad , \tag{20}$$

where θ_r is the rake of the propeller blade at radius r_1 .

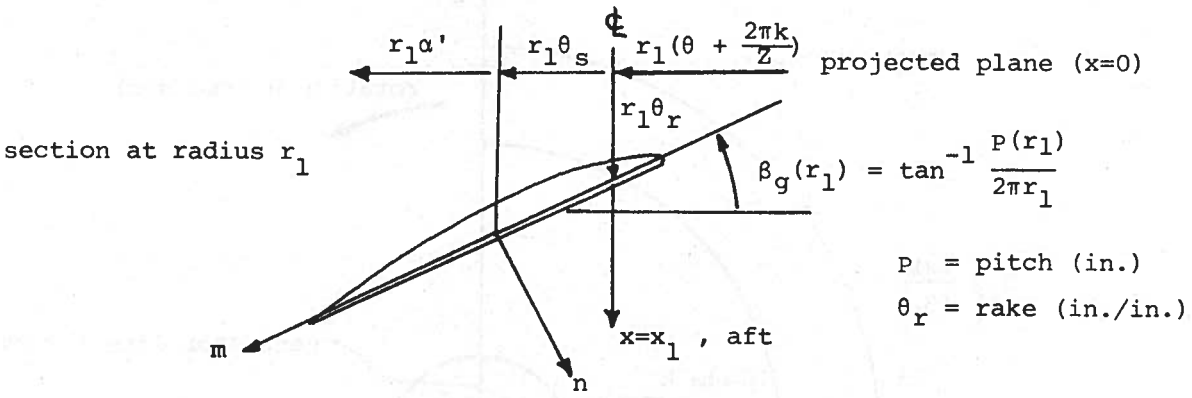


Figure 3. Coordinate System in Local Plane of Propeller Blade

Small displacements x_ℓ in the overall coordinate system shown in Fig. 1 will produce a vector displacement in the projected-plane coordinate system shown in Fig. 2 given by,

$$\underline{\delta} = \sum_{\ell=1}^{6} \underline{Y}_\ell x_\ell \quad , \tag{21}$$

where,

$$\underline{Y}1 = \underline{i} \quad ,$$

$$\underline{Y}2 = -\sin\theta_1\underline{e}_r - \cos\theta_1\underline{e}_\theta \quad ,$$

$$\underline{Y}3 = \cos\theta_1\underline{e}_r - \sin\theta_1\underline{e}_\theta \quad ,$$

$$\underline{Y}4 = r_1\underline{e}_\theta \quad ,$$

$$\underline{Y}5 = r_1\cos\theta_1\underline{i} - x_1\cos\theta_1\underline{e}_r + x_1\sin\theta_1\underline{e}_\theta \quad ,$$

$$\underline{Y}6 = r_1\sin\theta_1\underline{i} - x_1\sin\theta_1\underline{e}_r - x_1\cos\theta_1\underline{e}_\theta \quad .$$

The unit normal $\underline{n}$ in the local coordinate system shown in Fig. 3 is similarly related to the projected-plane unit vectors by,

$$\underline{n} = \cos\beta_g\underline{i} - \sin\beta_g\underline{e}_\theta \quad . \tag{22}$$

If a propeller undergoes a small general motion in the overall

coordinates, eqs. (21) and (22) can be used to obtain the resulting displacement normal to the blade section; i.e.,

$$\delta n = \underline{\delta}\cdot\underline{n} = (\delta_x + r_1\theta_y\cos\theta_1 + r_1\theta_z\sin\theta_1)\cos\beta_g + (\delta_y\cos\theta_1 + \delta_z\sin\theta_1 - r_1\theta_x - x_1\theta_y\sin\theta_1 + x_1\theta_z\cos\theta_1)\sin\beta_g. \quad (23)$$

The heave motion of the local blade section is given by eq. (23) evaluated at the midchord point, i.e., $\alpha' = 0$ in eq. (17) and eq. (20). The pitch motion of the local blade section about its midchord point is given by,

$$\delta n'\Big|_{\alpha'=0} = \frac{\partial \delta n}{\partial m}\bigg|_{\alpha'=0} = \frac{\partial \delta n}{\partial \alpha'}\frac{\partial \alpha'}{\partial m}\bigg|_{\alpha'=0} = \frac{\cos\beta_g}{r_1}\frac{\partial \delta n}{\partial \alpha'}\bigg|_{\alpha'=0} . \quad (24)$$

This notation signifies that local angle α' is set to zero only after all derivatives have been taken.

2.3. Heave and Pitch Due to Oscillations in Overall Coordinates

To utilize equations (15) and (16), we need the force or moment in coordinate j resulting from a unit oscillation in a single coordinate ℓ ; i.e., $F^*_{h_{j\ell}}$. It is convenient, however, to derive the expressions for a small general displacement $\underline{\delta}$ and then specialize this later to a unit displacement in coordinate ℓ . Consider oscillations in each coordinate ℓ at frequency ω ,

$$x_\ell = \mathrm{Re}\{A_\ell e^{i\omega t}\} . \quad (25)$$

These can be substituted into eq. (23) to obtain the resulting oscillatory motion normal to the blade section. The complex form can be used for $\sin\theta'$ and $\cos\theta'$ in the terms involving these functions in eq. (23) to give expressions of the form,

$$x_\ell\cos\theta_1 = \frac{1}{2}\mathrm{Re}\{A_\ell e^{iB_e i(\omega-\Omega)t} + A_\ell e^{-iB_e i(\omega+\Omega)t}\} , \quad (26)$$

$$x_\ell\sin\theta_1 = -\frac{1}{2}\mathrm{Re}\{A_\ell i e^{iB_e i(\omega-\Omega)t} - A_\ell i e^{-iB_e i(\omega+\Omega)t}\} , \quad (27)$$

where,

$$B(r_1,\alpha',k) = \frac{2\pi k}{Z} + \theta_s(r_1) + \alpha' \quad . \tag{28}$$

The resulting oscillatory motion normal to the blade section can be expressed in the compact form,

$$\delta n(x_1,r_1,\alpha',k,t) = \mathrm{Re}\{\sum_{\ell=1}^{6} A_\ell [C_\ell^- e^{i(\omega-\Omega)t} + C_\ell e^{i\omega t} + C_\ell^+ e^{i(\omega+\Omega)t}]\}. \tag{29}$$

Note that this motion has components at three frequencies. Recall that ω is the vibration frequency and Ω is the propeller rotation frequency. These three frequencies are analogous to the frequencies $(Z-1)\Omega$, $Z\Omega$, and $(Z+1)\Omega$, which occur in a similar manner in the blade-rate propeller vibratory excitation problem. The C coefficients are given in Table 1. The zeros in Table 1 reflect the lack of coupling between torsional/axial motion and motion in the lateral plane.

ℓ	C_ℓ^-	C_ℓ	C_ℓ^+
1	0	$\cos\beta_g$	0
2	$\frac{1}{2}e^{iB}\sin\beta_g$	0	$\frac{1}{2}e^{-iB}\sin\beta_g$
3	$-\frac{i}{2}e^{iB}\sin\beta_g$	0	$-\frac{i}{2}e^{-iB}\sin\beta_g$
4	0	$-r_1\sin\beta_g$	0
5	$(\frac{r_1}{2}\cos\beta_g + \frac{x_1 i}{2}\sin\beta_g)e^{iB}$	0	$(\frac{r_1}{2}\cos\beta_g - \frac{x_1 i}{2}\sin\beta_g)e^{-iB}$
6	$(-\frac{r_1 i}{2}\cos\beta_g + \frac{x_1}{2}\sin\beta_g)e^{iB}$	0	$(\frac{r_1 i}{2}\cos\beta_g + \frac{x_1}{2}\sin\beta_g)e^{-iB}$

Table 1. C Coefficients for Blade Normal Displacement Eq. (29)

The local oscillatory heave of the propeller blade section due to small oscillations in the overall coordinates can be obtained by substituting $\alpha' = 0$ into eqs. (20) and (28), and then using eq. (29). This yields,

$$\delta n(0) = \delta n\Big|_{\alpha'=0} = \mathrm{Re}\{\sum_{\ell=1}^{6} A_\ell [C_\ell^-(0)e^{i(\omega-\Omega)t} + C_\ell(0)e^{i\omega t} + C_\ell^+(0)e^{i(\omega+\Omega)t}]\}, \tag{30}$$

where the $C(0)$ coefficients are from Table 1 with $\alpha' = 0$ which gives,

$$B = B_o = B(r_1,0,k) = \frac{2\pi k}{Z} + \theta_s(r_1) \quad , \tag{31}$$

$$x_1 = x_{10} = x_1(r_1,0) = r_1\theta_r(r_1) + r_1\theta_s(r_1)\tan\beta_g \quad . \tag{32}$$

The local oscillatory pitch of the blade section about its midchord point due to small oscillations in the overall coordinates can be obtained by using eq. (29) in eq. (24). The result can be expressed in the compact form,

$$\delta n'(0) = \mathrm{Re}\{\sum_{\ell=1}^{6} A_\ell [D_\ell^- e^{i(\omega-\Omega)t} + D_\ell e^{i\omega t} + D_\ell^+ e^{i(\omega+\Omega)t}]\} \quad , \tag{33}$$

where the D coefficients are obtained from the corresponding C coefficients using,

$$D = \frac{\cos\beta_g}{r_1} \frac{\partial C}{\partial \alpha'}\Big|_{\alpha'=0} \quad . \tag{34}$$

The D coefficients are given in Table 2 where eq. (31) and eq. (32) are used in these expressions for B and x_1, respectively. The zeros in Table 2 reflect the lack of coupling between torsional/axial motion and motion in the lateral plane and indicate that there is no local pitch of the blade section in torsional or axial motion.

ℓ	D_ℓ^-	D_ℓ	D_ℓ^+
1	0	0	0
2	$\frac{i}{2r_1}e^{iB}\sin\beta_g\cos\beta_g$	0	$-\frac{i}{2r_1}e^{-iB}\sin\beta_g\cos\beta_g$
3	$\frac{1}{2r_1}e^{iB}\sin\beta_g\cos\beta_g$	0	$\frac{1}{2r_1}e^{-iB}\sin\beta_g\cos\beta_g$
4	0	0	0
5	$(\frac{i}{2}-\frac{x_1}{2r_1}\sin\beta_g\cos\beta_g)e^{iB}$	0	$(-\frac{i}{2}-\frac{x_1}{2r_1}\sin\beta_g\cos\beta_g)e^{-iB}$
6	$(\frac{1}{2}+\frac{x_1 i}{2r_1}\sin\beta_g\cos\beta_g)e^{iB}$	0	$(-\frac{1}{2}-\frac{x_1 i}{2r_1}\sin\beta_g\cos\beta_g)e^{-iB}$

Table 2. D Coefficients for Blade Local Pitch Eq. (33).

2.4. Forces Due to Unit Oscillations

Equations (30) and (33) indicate that oscillation in the overall coordinates at a frequency ω will produce local heave and pitch, respectively, of the propeller blade section at the three frequencies $\omega - \Omega$, ω, and $\omega + \Omega$. Assume we are able to calculate the pressure distribution on the propeller blade due to a unit heave or unit pitch at any frequency ω'. These results could be written in a complex form as follows for heave and pitch, respectively:

$$\text{heave:} \quad p(r_1,\alpha',k,\omega',t) = \mathrm{Re}\{P(r_1,\alpha',k)e^{i\omega' t}\} \quad , \tag{35}$$

$$\text{pitch:} \quad p'(r_1,\alpha',k,\omega',t) = \mathrm{Re}\{P'(r_1,\alpha',k)e^{i\omega' t}\} \quad . \tag{36}$$

We can designate the complex pressure amplitudes due to a unit heave at frequencies $\omega - \Omega$, ω, and $\omega + \Omega$ as P_-, P, and P_+, respectively. Similarly, we can designate the complex pressure amplitudes due to a unit pitch at frequencies $\omega - \Omega$, ω, and $\omega + \Omega$ as P'_-, P', and P'_+, respectively. Section 3 will

discuss how these pressure distributions can be calculated.

Equation (30) gives the magnitude of the heave of the local blade section which occur simultaneously at the frequencies $\omega - \Omega$, ω, and $\omega + \Omega$ as the propeller undergoes a general oscillation at frequency ω. The resulting pressure distribution can be obtained by summing the products of the magnitude of heave at each frequency times the pressure distribution due to a unit heave at that frequency. Equation (33) gives the magnitude of the pitch at the three frequencies which also occur as the propeller undergoes a general oscillation at frequency ω. The pitching will contribute three more pressure components which can be obtained by multiplying the pitch magnitude times the pressure distribution due to a unit pitch at the corresponding frequency. The total pressure at projected-plane coordinates r_1 and α' on blade k is therefore given by the sum of the six possible pressure components; i.e.,

$$p(r_1,\alpha',k,\omega,t) = \mathrm{Re}\{\sum_{\ell=1}^{6} A_\ell [(C_\ell^-(0)P_- + D_\ell^-(0)P_-')e^{i(\omega-\Omega)t} + (C_\ell(0)P + D_\ell(0)P')e^{i\omega t} + (C_\ell^+(0)P_+ + D_\ell^+(0)P_+')e^{i(\omega+\Omega)t}]\}. \quad (37)$$

To use eq. (15) and eq. (16), we need the pressure distribution due to a unit oscillation in coordinate ℓ so $A_\ell = 1$ and $A_i = 0$; $i=1,2,\ldots6$, $i \neq \ell$. Equation (37) then yields,

$$p_\ell(r_1,\alpha',k,\omega,t) = \mathrm{Re}\{(C_\ell^-(0)P_- + D_\ell^-(0)P_-')e^{i(\omega-\Omega)t} + (C_\ell(0)P + D_\ell(0)P')e^{i\omega t} + (C_\ell^+(0)P_+ + D_\ell^+(0)P_+')e^{i(\omega+\Omega)t}\}. \quad (38)$$

To use eq. (15) and eq. (16), we need the force or moment component in coordinate j produced by this pressure; i.e., $F^*_{hj\ell}$. This can be obtained by integrating the correct component of the force produced by the pressure eq. (38) over the blade and summing over all Z blades. Mathematically this becomes,

$$\mathrm{Re}\{F^{*}_{h_{j\ell}} e^{i\omega t}\} = \mathrm{Re}\{\sum_{k=0}^{Z-1} \int_{r_h}^{r_t} \int_{-\alpha_L(r_1)}^{+\alpha_L(r_1)} p_\ell(r_1,\alpha',k,\omega,t)(\underline{n}\cdot\underline{\gamma}_j)\frac{r_1 d\alpha'}{\cos\beta_g} dr_1\} \quad , \tag{39}$$

where the radius is integrated from hub to tip and α_L is the projected angle of the semichord at radius r_1. The vector dot products which produce the appropriate components of the force are available from eq. (21) and eq. (22); i.e.,

$$\begin{aligned}
(\underline{n}\cdot\underline{\gamma}_1) &= \cos\beta_g \ , \\
(\underline{n}\cdot\underline{\gamma}_2) &= \cos\theta_1 \sin\beta_g \ , \\
(\underline{n}\cdot\underline{\gamma}_3) &= \sin\theta_1 \sin\beta_g \ , \\
(\underline{n}\cdot\underline{\gamma}_4) &= -r_1 \sin\beta_g \ , \\
(\underline{n}\cdot\underline{\gamma}_5) &= r_1\cos\theta_1\cos\beta_g - x_1\sin\theta_1\sin\beta_g \ , \\
(\underline{n}\cdot\underline{\gamma}_6) &= r_1\sin\theta_1\cos\beta_g + x_1\cos\theta_1\sin\beta_g \ .
\end{aligned} \tag{40}$$

In eq. (40), α' is not zero in general and eq. (17) and eq. (20) must be used for θ_1 and x_1, respectively.

Equation (39) indicates that even though the pressure on the propeller blade is at three frequencies, the force this produces is at the single frequency ω. This result can be illustrated mathematically by presenting two examples of the use of eq. (39). These examples will also show how we reduce this expression to a form convenient for computer solution. Substituting eq. (38) into eq. (39) and taking the summation inside the integrals we obtain,

$$\begin{aligned}
\mathrm{Re}\{F^{*}_{h_{j\ell}} e^{i\omega t}\} = \mathrm{Re}\{ \int_{r_h}^{r_t} \int_{-\alpha_L(r_1)}^{\alpha_L(r_1)} & [(G^{-}_{j\ell}P_{-}+H^{-}_{j\ell}P^{-}_{-}) \\
& + (G_{j\ell}P+H_{j\ell}P') + (G^{+}_{j\ell}P_{+}+H^{+}_{j\ell}P'_{+})]\frac{r_1 d\alpha'}{\cos\beta_g} dr_1 e^{i\omega t}\} \quad ,
\end{aligned} \tag{41}$$

where,

$$G^{\pm}_{j\ell} = e^{\mp i\Omega t} \sum_{k=0}^{Z-1} C^{\pm}_{\ell}(0)(\underline{n}\cdot\underline{\gamma}_j) \ , \tag{42}$$

$$H^{\pm}_{j\ell} = e^{\mp i\Omega t} \sum_{k=0}^{Z-1} D^{\pm}_{\ell}(0)(\underline{n}\cdot\underline{\gamma}_j) \ , \tag{43}$$

$$G_{j\ell} = \sum_{k=0}^{Z-1} C_{\ell}(0)(\underline{n}\cdot\underline{\gamma}_j) \ , \tag{44}$$

$$H_{j\ell} = \sum_{k=0}^{Z-1} D_{\ell}(0)(\underline{n}\cdot\underline{\gamma}_j) \ . \tag{45}$$

Note that the negative exponential is used with the plus superscript (G^+ and H^+) in eq. (42) and (43) and vice versa.

Consider first the axial added mass and axial damping case (ℓ=1, j=1) . From Tables 1 and 2, only $C_1(0) = \cos\beta_g$ is non-zero so only G_{11} is non-zero. We therefore have from eq. (41),

$$F^*_{h11} = \int_{r_h}^{r_t} \int_{-\alpha_L(r_1)}^{\alpha_L(r_1)} G_{11}(r_1,\alpha')P(r_1,\alpha')\frac{r_1 d\alpha'}{\cos\beta_g} dr_1 \ , \tag{46}$$

and using $(\underline{n}\cdot\gamma_1)$ from eq. (40) in eq. (44) we have,

$$G_{11}(r_1,\alpha') = \sum_{k=0}^{Z-1} \cos^2\beta_g = Z\cos^2\beta_g \ . \tag{47}$$

The reduction of the summation in eq. (47) to just the multiplier Z indicates that all Z blades contribute equally to the axial added mass and axial damping. Equation (46) can be rearranged to yield,

$$F^*_{h11} = Z \int_{r_h}^{r_t} \left[\int_{-\alpha_L}^{\alpha_L} P(r_1,\alpha')\frac{d\alpha'}{\cos\beta_g} \right] \cos^2\beta_g r_1 dr_1 \ . \tag{48}$$

For programming convenience we define the term in brackets as

$$PHX(r_1) = \int_{-\alpha_L(r_1)}^{\alpha_L(r_1)} P(r_1,\alpha')\frac{d\alpha'}{\cos\beta_g(r_1)} \quad , \tag{49}$$

a chordwise integral of the pressure due to a unit heave at frequency ω . Equation (48) then becomes simply,

$$F^*_{h_{11}} = Z \int_{r_h}^{r_t} PHX(r_1)\cos^2\beta_g \, r_1 dr_1 \quad , \tag{50}$$

and the axial added mass m_{11} can be obtained using eq. (15) and the axial damping c_{11} can be obtained using eq. (16).

The analysis is not so simple in the lateral case. As an example of the other extreme, consider the lateral (diametral) added mass moment of inertia and lateral rotational damping case (ℓ=5, j=5). The C and D coefficients from Tables 1 and 2, respectively, are used with B_o and X_{10} from eq. (31) and (32), respectively. Since $C_5 = D_5 = 0$, eq. (44) and (45) yield $G_{55} = H_{55} = 0$. The complex form can be used for $\cos\theta_1$ and $\sin\theta_1$ in eq. (40) to give,

$$(\underline{n}\cdot\underline{Y}_5) = \frac{r_1}{2}(e^{iB}e^{-i\Omega t}+e^{-iB}e^{i\Omega t})\cos\beta_g + \frac{x_1 i}{2}(e^{iB}e^{-i\Omega t}-e^{-iB}e^{i\Omega t})\sin\beta_g, \tag{51}$$

where α' is not zero in general so we must use,

$$B = B_o + \alpha' \quad , \text{ and}$$

$$x_1 = x_{10} + r_1\alpha'\tan\beta_g \quad ,$$

in this expression. Now C_5^- , D_5^- , and eq. (51) can be substituted in eq. (42) to evaluate G_{55}^- . If we take advantage of the fact that,

$$\sum_{k=0}^{Z-1} e^{\pm 2iB_o} = e^{\pm 2i\theta_s} \sum_{k=0}^{Z-1} e^{\pm\frac{4ik}{Z}} = 0 \quad , \quad Z > 2 \quad , \tag{52}$$

we obtain the following expression for G_{55}^- :

$$G^{-}_{55} = \frac{Z}{4}[(r_1^2\cos^2\beta_g + x_{10}^2\sin^2\beta_g)e^{-i\alpha'}$$
$$+ (-r_1^2 i\sin^2\beta_g + x_{10}r_1\sin^2\beta_g\tan\beta_g)\alpha' e^{-i\alpha'}] \quad . \qquad (53)$$

Similar expressions can be obtained for G^{+}_{55} , H^{-}_{55} , and H^{+}_{55} . The zero summation in eq. (52) indicates that the contributions from the Z blades exactly cancel one another due to propeller symmetry. Substituting all these results into eq. (41) yields the following spanwise integral for $F^{*}_{h_{55}}$:

$$F^{*}_{h_{55}} = \frac{Z}{4}\int_{r_h}^{r_t}[(r_1^3\cos^2\beta_g + x_{10}^2 r_1\sin^2\beta_g)PHLM$$
$$+ (-r_1^3 i\sin^2\beta_g + x_{10}r_1^2\sin^2\beta_g\tan\beta_g)PHLMA$$
$$+ (r_1^2 i\cos\beta_g + x_{10}r_1\sin^3\beta_g + x_{10}^2 i\sin^2\beta_g\cos\beta_g)PPLM$$
$$+ (r_1^2\sin\beta_g\tan\beta_g + x_{10}r_1 i\sin^3\beta_g)PPLMA$$
$$+ (r_1^3\cos^2\beta_g + x_{10}^2 r_1\sin^2\beta_g)PHLP$$
$$+ (r_1^3 i\sin^2\beta_g + x_{10}r_1^2\sin^2\beta_g\tan\beta_g)PHLPA$$
$$+ (-r_1^2 i\cos\beta_g + x_{10}r_1\sin^3\beta_g - x_{10}^2 i\sin^2\beta_g\cos\beta_g)PPLP$$
$$+ (r_1^2\sin\beta_g\tan\beta_g - x_{10}r_1 i\sin^3\beta_g)PPLPA]dr_1 \qquad (54)$$

where,

$$PHL\begin{bmatrix}P\\M\end{bmatrix}(r_1) = \int_{-\alpha_L(r_1)}^{\alpha_L(r_1)} P_{\pm}(r_1,\alpha')e^{\pm i\alpha'}\frac{d\alpha'}{\cos\beta_g} \quad , \qquad (55)$$

$$PHL\begin{bmatrix}P\\M\end{bmatrix}A(r_1) = \int_{-\alpha_L(r_1)}^{\alpha_L(r_1)} P_{\pm}(r_1,\alpha')\alpha' e^{\pm i\alpha'}\frac{d\alpha'}{\cos\beta_g} \quad , \qquad (56)$$

$$PPL\begin{bmatrix}P\\M\end{bmatrix}(r_1) = \int_{-\alpha_L(r_1)}^{\alpha_L(r_1)} P'_{\pm}(r_1,\alpha')e^{\pm i\alpha'}\frac{d\alpha'}{\cos\beta_g} \quad , \qquad (57)$$

$$PPL\begin{bmatrix}P\\M\end{bmatrix}A(r_1) = \int_{-\alpha_L(r_1)}^{\alpha_L(r_1)} P'_{\pm}(r_1,\alpha')\alpha' e^{\pm i\alpha'} \frac{d\alpha'}{\cos\beta_g} \quad , \qquad (58)$$

with the plus sign used with P and the minus sign used with the M in the chordwise pressure integral name. Comparing eqs. (53) and (54) shows that the term $\bar{G}_{55}P_-$ in eq. (41) produces the first two terms in the integrand in eq. (54).

Equation (41) was evaluated for all 36 combinations of i and j. These results yield only nine unique, nonzero results and the added mass and damping matrix properties shown in eq. (8). The complete expressions for the nine complex hydrodynamic forces $F^*_{h_{j\ell}}$ are given in Table 3. All terms within the integrals are functions of the variable of integration r_1 . The chordwise pressure integrals are defined by eq. (49) and eqs. (55) through (58). In the computer program PRAMAD which calculates the propeller added mass and damping using these results, the chordwise pressure integrals for a unit heave at frequency ω , eq. (49), and for a unit pitch and a unit heave at frequencies $\omega - \Omega$ and $\omega + \Omega$, eq. (55) through (58), are first calculated and stored as a function of radius. These results are then combined into the appropriate radial integrands as given in Table 3 and integrated to give the desired complex hydrodynamic forces. The associated added mass and damping are then obtained using eq. (15) and eq. (16), respectively.

$$F^*_{h_{11}} = Z \int_{r_h}^{r_t} PHX\cos^2\beta_g r_1 dr_1$$

$$F^*_{h_{22}} = \frac{Z}{4} \int_{r_h}^{r_t} r_1\sin\beta_g [PHLM + \frac{i\cos\beta_g}{r_1} PPLM + PHLP - \frac{i\cos\beta_g}{r_1} PPLP] dr_1$$

$$F^*_{h_{32}} = \frac{Z}{4} \int_{r_h}^{r_t} r_1\sin^2\beta_g [iPHLM - \frac{\cos\beta_g}{r_1} PPLM - iPHLP - \frac{\cos\beta_g}{r_1} PPLP] dr_1$$

$$F^*_{h_{41}} = -Z \int_{r_h}^{r_t} PHX\cos\beta_g \sin\beta_g r_1^2 dr_1$$

$$F^*_{h_{44}} = Z \int_{r_h}^{r_t} PHX\sin^2\beta_g r_1^3 dr_1$$

$$F^*_{h_{52}} = \frac{Z}{4} \int_{r_h}^{r_t} \sin\beta_g [(a-ib)PHLM - iePHLMA + (c+id)PPLM + fPPLMA$$
$$+ (a+ib)PHLP + iePHLPA + (c-id)PPLP + fPPLPA] dr_1$$

$$F^*_{h_{55}} = \frac{Z}{4} \int_{r_h}^{r_t} [gPHLM + (aa-ibb)PHLMA + (cc+idd)PPLM + (e+icc)PPLMA] dr_1$$
$$+ gPHLP + (aa+ibb)PHLPA + (cc-idd)PPLP + (e-icc)PPLPA] dr_1$$

$$F^*_{h_{62}} = \frac{Z}{4} \int_{r_h}^{r_t} \sin\beta_g [(b+ia)PHLM + ePHLMA - (d-ic)PPLM + ifPPLMA$$
$$+ (b-ia)PHLP + ePHLPA - (d+ic)PPLP - ifPPLPA] dr_1$$

$$F^*_{h_{65}} = \frac{Z}{4} \int_{r_h}^{r_t} [igPHLM + (bb+iaa)PHLMA - (dd-icc)PPLM - (cc-ie)PPLMA$$
$$- igPHLP + (bb-iaa)PHLPA - (dd+icc)PPLP - (cc+ie)PPLPA] dr_1$$

where

$a = r_1^2\cos\beta_g$

$b = x_{10}r_1\sin\beta_g$

$c = x_{10}\sin\beta_g\cos\beta_g$

$d = r_1\cos^2\beta_g$

$e = r_1^2\sin\beta_g\tan\beta_g$

$f = r_1\sin^2\beta_g$

$g = r_1^3\cos^2\beta_g + x_{10}^2 r_1\sin^2\beta_g$

$aa = x_{10}r_1^2\sin^2\beta_g\tan\beta_g$

$bb = r_1^3\sin^2\beta_g$

$cc = x_{10}r_1\sin^3\beta_g$

$dd = x_{10}^2\sin^2\beta_g\cos\beta_g + r_1^2\cos\beta_g$

Table 3. Complex Hydrodynamic Force Integrals

3. Pressure Distributions Due to Unit Oscillation

In this section, we review techniques which can be used to establish the chordwise pressure distribution acting on a propeller blade at some radius r_1 as it undergoes a unit oscillation at some frequency ω' in heave or pitch about its midchord point. These pressure distributions are denoted p and p' in eq. (35) and eq. (36). The associated complex magnitudes $P(r_1, \alpha', \omega')$ and $P'(r_1, \alpha', \omega')$ are utilized in the chordwise pressure integrals eq. (49) and eqs. (55) through (58). The available techniques parallel those used in steady propeller theory. In the order of increasing accuracy, complexity, and cost, they are two-dimensional thin foil theory, lifting-line theory, and lifting-surface theory. To date, our work has been limited to the first two approaches.

3.1. Two-Dimensional Thin Foil Theory

The complete solution for small lateral oscillations of a thin foil in a uniform stream of incompressible fluid was first published in this country by Theodorsen.[17] Here we utilize the presentation of Theodorsen's method given by Bisplinghoff, Ashley, and Halfman.[18] They utilize the coordinate system shown in Fig. 4. The thin foil undergoes heave and pitch oscillations due to a vertical oscillatory motion,

$$z_a(x,t) = \mathrm{Re}\{\bar{z}_a(x)e^{i\omega t}\} \quad . \tag{59}$$

The vertical fluid velocity at the foil surface is,

$$w_a(x,t) = \mathrm{Re}\{\bar{w}_a(x)e^{i\omega t}\} \quad . \tag{60}$$

Under the usual thin foil approximations, the linearized boundary condition is,

$$w_a(x,t) = \frac{\partial z_a}{\partial t} + U\frac{\partial z_a}{\partial x} \quad , \tag{61}$$

which yields,

$$\bar{w}_a(x) = i\omega\bar{z}_a(x) + \frac{\partial \bar{z}_a}{\partial x} \quad . \tag{62}$$

Note that in Fig. 4, positive heave is in the negative z-direction.

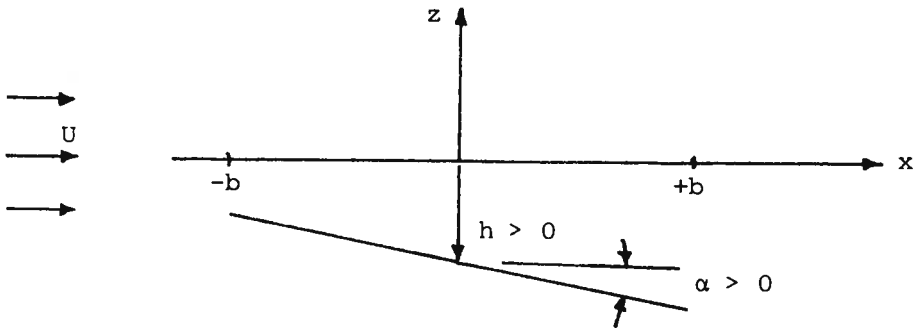


Figure 4. Coordinate System for Two-Dimensional Thin Foil

The resulting pressure distribution, or more specifically the difference between the pressure on the upper and lower surfaces ($z=0^{\pm}$) of the foil, will also be oscillatory; i.e.,

$$\Delta p(x,t) = p(x,0^+,t) - p(x,0^-,t) = \mathrm{Re}\{\Delta\bar{P}_a(x)e^{i\omega t}\} \quad , \tag{63}$$

and this will produce an oscillatory lift,

$$\ell(t) = -\int_{-b}^{b} \Delta p(x,t)dx = \mathrm{Re}\{Le^{i\omega t}\} \quad . \tag{64}$$

The pressure distribution is given by,

$$-\frac{\Delta\bar{P}_a(x^*)}{\rho U} = \frac{2}{\pi}\left\{[1-C(k)]\sqrt{\frac{1-x^*}{1+x^*}}\int_{-1}^{1}\sqrt{\frac{1+\xi^*}{1-\xi^*}}\,\bar{w}_a(\xi^*)d\xi^*\right.$$

$$\left. + \int_{-1}^{1}\left[\sqrt{\frac{1-x^*}{1+x^*}}\sqrt{\frac{1+\xi^*}{1-\xi^*}}\,\frac{1}{x^*-\xi^*} - ik\Lambda(x^*,\xi^*)\right]\bar{w}_a(\xi^*)d\xi^*\right\}, \tag{65}$$

where,

$$x^* = \frac{x}{b} \quad ,$$

$$\xi^* = \frac{\xi}{b} \quad , \quad \text{and}$$

$$k = \frac{\omega b}{U} \quad , \quad \text{the reduced frequency.} \tag{66}$$

The function C(k) is the Theodorsen function,

$$C(k) = \frac{J_1(k)-iY_1(k)}{J_1(k)-iY_1(k)+iJ_0(k)+Y_0(k)} \quad , \tag{67}$$

where J_i and Y_i are Bessel functions of the first and second kinds, respectively. The function Λ is given by,

$$\Lambda(x^*,\xi^*) = \frac{1}{2} \ln \left[\frac{1-x^*\xi^*+\sqrt{1-\xi^{*2}}\sqrt{1-x^{*2}}}{1-x^*\xi^*-\sqrt{1-\xi^{*2}}\sqrt{1-x^{*2}}} \right] \quad . \tag{68}$$

Using eq. (63) and eq. (64), the complex lift magnitude is given by,

$$L = \int_{-b}^{b} [-\Delta\bar{P}_a(x)]dx = b \int_{-1}^{1} [-\Delta\bar{P}_a(x^*)]dx^* \quad . \tag{69}$$

If we define a complex lift coefficient $C_L = L/\rho Ub$ we then get,

$$C_L = \int_{-1}^{+1} \left[-\frac{\Delta\bar{P}_a(x^*)}{\rho U} \right] dx^* \quad , \tag{70}$$

which is a chordwise integration of the pressure distribution given by eq. (65).

We can now specialize these results to heave and pitch of the foil. In heave, we have,

$$z_a(x,t) = -h(t) = -\mathrm{Re}\{\bar{h}e^{i\omega t}\} \quad , \tag{71}$$

so eq. (62) yields,

$$\bar{w}_a(x) = -i\omega\bar{h} \quad . \tag{72}$$

Bisplinghoff, et al[18] show that the resulting integrated lift gives the complex lift coefficient,

$$C_L = \pi \bar{h}\omega[2i\ C(k)-k] \quad . \tag{73}$$

The first part of this expression is the circulatory part due to the interaction of the heave and the circulation present on the operating propeller. The second part of this expression is the noncirculatory part and would be present even if the propeller were not rotating. In pitch, we have,

$$z_a(x,t) = -\alpha(t)x = -Re\{x\bar{\alpha}e^{i\omega t}\} \quad , \tag{74}$$

so eq. (62) yields,

$$\bar{w}_a(x) = -\ i\omega\bar{\alpha}x - U\bar{\alpha} = -U\bar{\alpha}(1+ikx^*) \quad . \tag{75}$$

Bisplinghoff, et al[18] show that the resulting integrated lift gives the complex lift coefficient,

$$C_L = \pi U\bar{\alpha}[ik+C(k)(2+ik)] \quad . \tag{76}$$

In this expression the first part is the noncirculatory part; the second part is the circulatory part.

The two-dimensional thin foil theory results can now be specialized to the propeller vibration problem of interest here. We need the pressure distribution for a unit heave $\delta n(0)=1$ and for a unit pitch about the midchord point of the blade section $\delta n'(0)=1$. Comparing the coordinate systems in Fig. 3 and Fig. 4 we have,

$$\text{heave:} \quad \delta n(0) = -h \quad , \quad \text{so} \quad \bar{h} = -1 \quad , \tag{77}$$

$$\text{pitch:} \quad \delta n'(0) = -\alpha \quad , \quad \text{so} \quad \bar{\alpha} = -1 \quad . \tag{78}$$

The results for a unit heave oscillation from eq. (72), eq. (73), and eq. (77) then become,

$$\bar{w}_a(x) = i\omega \quad , \tag{79}$$

$$C_L = \pi\omega[k-2iC(k)] \quad . \tag{80}$$

The results for a unit pitch oscillation from eq. (75), eq. (76), and eq. (78) then become,

$$\bar{w}_a(x) = U(1+ikx^*) \quad , \tag{81}$$

$$C_L = -\pi U[ik+C(k)(2+ik)] \quad . \tag{82}$$

The linearized boundary condition magnitudes eq. (79) and eq. (81) are used in eq. (65) to obtain the pressure distribution for the unit heave and the unit pitch, respectively.

This completes the development of the two-dimensional thin foil results. The pressure distribution given by eq. (65) is singular at the leading edge $x^*=-1$ as is typical in the linearized theory. Care must also be taken in numerically evaluating the integrals on ξ^* due to the singularities at $\xi^*=x^*$ and at the leading and trailing edges. In our work, we evaluate the integrals on ξ^* using a 100 segment rectangular rule with the integrand evaluated at the <u>center</u> of each segment. This avoids the leading edge singularity directly. The integrands at the center of the trailing edge segment are obtained by assuming $\bar{w}_a$ constant at its value at the trailing edge and then integrating the remainder of the integrand analytically. If the integrand with the $(x^*-\xi^*)^{-1}$ singularity is evaluated within 0.001 of $\xi^*=x^*$, the integrand is set equal to zero since the singularity can be shown to integrate to zero in the limit; i.e. the Cauchy principal value integral is indicated in eq. (65). The final numerical difficulty is in obtaining the pressure at the leading edge $x^*=-1$. The pressure is known at the trailing edge to be $\Delta\bar{P}_a(1)=0$. The pressure is calculated at nine additional, equally-spaced points along the chord length. Equation (70) is then used to establish the *equivalent finite leading edge pressure* which will yield the correct integrated lift as required by either eq. (80) or eq. (82). Simpson's rule is used to perform the integration in eq. (70) and this equation is then solved for the equivalent leading edge pressure $\Delta\bar{P}_a(-1)$.

The two-dimensional results ignore the induced velocities caused by the spanwise and chordwise circulation distribution and

the finite span. The results will be shown below to generally over predict both the added mass and damping of the propeller. The solution is, however, very economical.

3.2 Lifting-Line Theory

Our lifting-line analysis models the propeller as an even number $4 \le MM \le 10$ of circumferential segments of equal radial width as shown in Fig. 5 for MM=4. The bound vorticity is assumed to have a constant value radially in each segment. This is essentially the approach used by Brown[19] except that skew has been added. The lifting-line segments are at a constant angle passing through the midchord point of the midpoint of each blade segment. The trailing vortex system has both radial and tangential components. The radial component is due to the time-varying nature of the bound vorticity. The tangential (streamwise) component is due to the radial variation of the bound vorticity. With the assumption of radially constant bound vorticity in each segment, the tangential component consists of discrete trailers beginning at the midchord point at the joints between the segments and at the ends of the blade. This lifting-line model reflects the spanwise distribution of circulation but neglects the chordwise distribution. The model will, therefore, be more appropriate for higher aspect ratio (lower expanded area ratio) propellers.

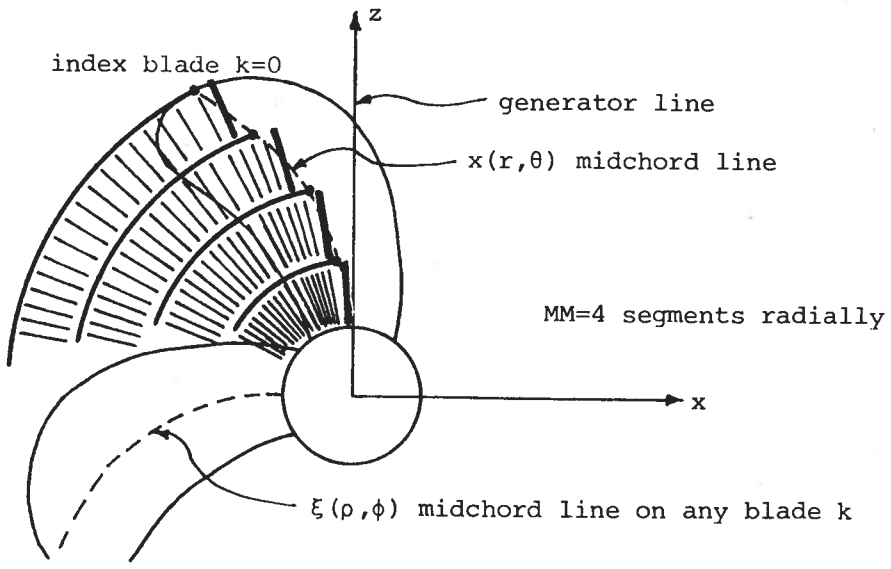


Figure 5. Lifting-Line Propeller Model

From Fig. 5, the angular coordinate to the midchord line of the index (k=0) blade at radius r when its reference line is vertical is just the projected skew angle,

$$\theta(r,k=0) = \theta_s(r) \quad . \tag{83}$$

The angular coordinate to the midchord line of any of the blades at radius ρ under the same condition is then,

$$\phi(\rho,k) = \frac{2\pi k}{Z} + \theta_s(\rho) \quad . \tag{84}$$

If we now let $w_a(r,t)$ be the normal velocity at point r on the midchord line of the index (k=0) blade, we have,

$$w_a(r,t) = \mathrm{Re}\{\bar{w}_a(r)e^{i\omega t}\} \quad , \tag{85}$$

where $\bar{w}_a(r)$ is the complex amplitude and ω is the vibration frequency. If we denote the fluid velocity normal to the blade as $w_{3\text{-}D}(r,t)$; i.e.,

$$w_{3\text{-}D}(r,t) = \mathrm{Re}\{\bar{w}_{3\text{-}D}(r)e^{i\omega t}\} \quad , \tag{86}$$

the kinematic boundary condition requires,

$$\bar{w}_a(r) = \bar{w}_{3\text{-}D}(r) \quad . \tag{87}$$

Employing a lifting-line model of the propeller with the bound vorticity along the midchord line, $\bar{w}_{3\text{-}D}(r)$ can be written,

$$\bar{w}_{3\text{-}D}(r) = \int_{r_h}^{r_t} \bar{\Gamma}(\rho)K_r(r,\rho)d\rho + \int_{r_h}^{r_t} \frac{d\bar{\Gamma}}{d\rho} K_t(r,\rho)d\rho \quad , \tag{88}$$

where $\bar{\Gamma}$ is the complex circulation and $d\bar{\Gamma}/d\rho$ is its radial derivative. The kernel functions K_r and K_t represent the normal velocity induced at r on the index blade by the radial and tangential vortex systems, respectively, of all Z blades. The integral over the tangential vortex system is evaluated in a Cauchy principal value sense just as in the steady theory. Its evaluation presents no particular difficulty. The integral over the radial vortex system has a more difficult singularity,

however, at $\rho=r$. Following Reissner[20] and Brown,[19] the difficulty in evaluating the radial vortex system integral can be eliminated by adding to and subtracting from eq. (88) an equivalent two-dimensional normal velocity,

$$w_{2-D}(r,t) = \mathrm{Re}\{\bar{w}_{2-D}(r)e^{i\omega t}\} \quad , \tag{89}$$

where the amplitude has the form,

$$\bar{w}_{2-D}(r) = \bar{\Gamma}(r)K_{2-D}(r) \quad . \tag{90}$$

This yields for eq. (87) and eq. (88),

$$\bar{w}_a(r) = \bar{w}_{2-D}(r) + \int_{r_h}^{r_t} [\bar{\Gamma}(\rho)K_r(r,\rho) - \bar{\Gamma}(r)K_{2-D}(r)]d\rho$$

$$+ \oint_{r_h}^{r_t} \frac{d\bar{\Gamma}}{d\rho} K_t(r,\rho)d\rho \quad . \tag{91}$$

The radial vortex system integral is now well-behaved at $\rho=r$ since $K_r(r,\rho)$ and $K_{2-D}(r)$ have the same singularity as $\rho\to r$.

The integral equation (91) can be solved numerically by modeling the bound vorticity as MM segments of constant circulation radially as shown in Fig. 5. We let K_{ij} be the combined kernel of both the radial and tangential vortex systems. With the assumption of piecewise radially constant $\bar{\Gamma}$, the radial derivative $d\bar{\Gamma}/d\rho$ is a sequence of delta functions at the joints between the segments. The strength of the discrete trailers are given by the difference in the circulation at the two adjacent segments. The singularity at $\rho=r$ in the tangential vortex system integral is avoided entirely since there is no trailing vortex at the midpoint of the segments where eq. (91) is satisfied.

Denoting i as the field point where eq. (91) is satisfied and denoting j as the integration points ρ, eq. (91) can be written as a system of MM simultaneous equations,

$$\bar{w}_{a_i} = \bar{w}_{2\text{-}Di} + \sum_{j=1}^{MM} [\bar{\Gamma}_j K_{ij} - \bar{\Gamma}_i K^*_{2\text{-}Di}] \quad , \quad i=1,\ldots,MM \ . \tag{92}$$

The complex circulation at each segment $\bar{\Gamma}_\ell$ can be written as,

$$\bar{\Gamma}_\ell = b_\ell \bar{w}_{2\text{-}D_\ell} G_\ell \quad , \tag{93}$$

where G_ℓ is the complex circulation obtained from Theodorsen's method for the two-dimensional thin foil and b_ℓ is the section semichord length. Our complex circulation G_ℓ is related to Bisplinghoff, et al's[18] nomenclature by,

$$G_\ell = \bar{\Omega} e^{-ik} \quad , \tag{94}$$

where they derive a complex reduced circulation $\bar{\Omega}$ given by,

$$\bar{\Omega} = \frac{4 \int_{-1}^{1} \sqrt{\frac{1+\xi^*}{1-\xi^*}}\, \bar{w}_a(\xi^*) d\xi^*}{\pi i k [J_1(k) - iY_1(k) + iJ_0(k) + Y_0(k)]} \ . \tag{95}$$

Substituting eq. (93) into eq. (92) produces a system of MM simultaneous equations,

$$\bar{w}_{a_i} = \bar{w}_{2\text{-}D_i} + \sum_{j=1}^{MM} [\bar{w}_{2\text{-}Dj} \bar{K}_{ij} - \bar{w}_{2\text{-}Di} \bar{K}^*_{2\text{-}Di}] \quad ; \quad i=1,\ldots,MM \quad , \tag{96}$$

in terms of the MM unknowns $\bar{w}_{2\text{-}D}$, the equivalent two-dimensional normal velocity at the midpoint of each blade segment. The left side of these equations is obtained by using eq. (79) or eq. (81) at the midpoint of each segment for the heave and pitch problems, respectively. The system of equations eq. (96) is solved for the $\bar{w}_{2\text{-}D}$ which are then used in place of w_a in eq. (65) to obtain the pressure distribution at the midpoint of each segment. The kernels $\bar{K}_{ij}$ and $\bar{K}^*_{2\text{-}Di}$ are essentially those used by Brown[20] except that the spatially oscillating exponential he includes for the nonuniform flow case is just 1 for the uniform flow in the added mass and damping problem and we include the skew angle in the definition of the midchord point.

This completes the development of the lifting-line theory. In summary, we use two-dimensional theory to obtain the complex circulation G_ℓ using eq. (94) and eq. (95). These results are used in the kernels $\bar{K}_{ij}$ and $\bar{K}^*_{2\text{-}Di}$ which form the system of MM simultaneous equations eq. (96). This system is then solved for the equivalent two-dimensional normal velocities $\bar{w}_{2\text{-}D}$ which are used in place of $\bar{w}_a$ in eq. (65) to obtain the chordwise pressure distribution $\Delta\bar{P}_a$. This theory neglects the chordwise distribution of the bound circulation and will therefore be less accurate for lower aspect ratio (higher expanded area ratio) propellers.

3.3. Lifting-Surface Theory

Lifting-surface theory models the chordwise as well as spanwise distribution of the bound vorticity. The development presented in Section 2 is equally valid when the pressure distributions for a blade undergoing a unit heave or unit pitch about its midchord point are obtained using an unsteady lifting-surface theory. Our work to date, however, has not utilized a lifting-surface theory. Kerwin and Lee[21] recently reviewed various lifting-surface theories and presented an unsteady lifting-surface theory for the unsteady flow problem. This approach could be adapted to the problem of the foil oscillating in a uniform flow. Hylarides and van Gent[6] have applied lifting-surface theory to the added mass and damping problem. They have published results for a series of four-bladed Wageningen B-Series propellers. These results will be discussed further below.

4. The PRAMAD Computer Program Results

The Propeller Added Mass and Damping program (PRAMAD) was written to calculate the added mass and damping of propellers using the theory developed in Section 2. Either the two-dimensional thin foil theory or the lifting-line theory results reviewed in Section 3, at the user's option, are used to calculate the pressure distributions. The program is a batch-type program written in FORTRAN IV. User's Instructions, Programmer's Documentation, and a listing of the program are included as Appendices A, B, and C, respectively. The program accepts the propeller geometry input data in a wide variety of forms at the user's option. Output is in English units, SI units, and a nondimensional form as defined in Table 4. Selected results obtained with this program are reviewed in this and following sections. The lifting-surface corrections which are included in the program are presented in Section 5.

Type of coefficient	coefficients	divisor
added mass moment of inertia	m_{44}, m_{55}, m_{65}	ρD^5
inertia coupling	m_{41}, m_{52}, m_{62}	ρD^4
added mass	m_{11}, m_{22}, m_{32}	ρD^3
rotational damping	c_{44}, c_{55}, c_{65}	$\rho n D^5$
velocity coupling	c_{41}, c_{52}, c_{62}	$\rho n D^4$
linear damping	c_{11}, c_{22}, c_{32}	$\rho n D^3$

Table 4. Nondimensionalization of Coefficients

4.1. Comparison of Two-Dimensional, Lifting-Line, and Lifting-Surface Results

The PRAMAD program evaluates the added mass and damping of a propeller using two-dimensional thin foil and lifting-line theories. Hylarides and van Gent[6] have published results which they obtained using a lifting-surface theory for a matrix of nine four-bladed Wageningen B-Series propellers. Their results were

presented in nondimensional form for a blade-rate vibration frequency ($\omega=4\Omega$, Ω=rotation frequency). The propellers were assumed to be "lightly-loaded;" i.e., operating at an advance coefficient $J_{\ell\ell}$ producing zero thrust. For comparison purposes we have duplicated their calculations for the B4-70-80 propeller. This propeller has a nominal expanded area ratio of .70 and a pitch-diameter ratio at the 0.7 radius of .80. The nondimensional results are shown in Table 5.

Torsional/Axial and Lateral: Parallel Results. There is reasonable agreement for the torsional/axial results and the lateral results with the forces parallel to the motion. In general, there is better agreement between the lifting-line and lifting-surface results for damping than for added mass. There is only a small difference between our two-dimensional and lifting-line results for added mass. This occurs because our lifting-line theory "corrects" only the circulatory part of the added mass and damping for finite aspect ratio. The added mass coefficients are dominated by the noncirculatory part which is effectively unchanged in the lifting-line analysis. In view of the differences between lifting-line and lifting-surface results, we have incorporated lifting-surface corrections for these coefficients into the PRAMAD program. The development of these corrections is described in Section 5.

Lateral: Perpendicular Results. There is little agreement for the lateral results with the forces perpendicular to the motion. Hylarides and van Gent present results for m_{35} and c_{35} which are different from m_{62} and c_{62}, respectively. We have shown by the theoretical derivation presented in Section 2 that these pairs of coefficients must be equal. The added mass results are generally quite small; i.e., m_{55} is thirty times m_{65}. The differences between the numerical results for added mass obtained by the lifting-line and lifting-surface theories are, therefore, not surprising. From an engineering standpoint these coefficients are negligible. Our results show damping values (c_{65},c_{62},c_{32}) which are comparable to the damping forces parallel

component	two dimensional from PRAMAD	lifting-line from PRAMAD	lifting-surface from Hylarides & van Gent[6]
torsional/axial			
m_{44}	.00166	.00163	.00119
m_{41}	-.01342	-.01316	-.00934
m_{11}	.10842	.10626	.07340
c_{44}	.02267	.00947	.01180
c_{41}	-.18063	-.07492	-.09260
c_{11}	1.44128	.59286	.72700
lateral:parallel			
m_{55}	.00470	.00486	.00358
m_{52}	.00624	.00639	.00467
m_{22}	.01381	.01381	.00866
c_{55}	.08915	.04397	.03790
c_{52}	.08905	.04016	.04410
c_{22}	.14586	.05936	.07010
lateral:perpendicular			
m_{65}	.00108	-.00014	.00015
m_{62}	.00090	-.00020	.00106
$m_{35} = m_{62}$	.00090	-.00020	-.00088
m_{32}	.00132	-.00017	-.00001
c_{65}	-.03743	-.05141	-.00518
c_{62}	-.04021	-.05181	.00610
$c_{35} = c_{62}$	-.04021	-.05181	-.01700
c_{32}	-.07609	-.09150	-.00100

Table 5. Comparison of Results for B4-70-80 Propeller ($J_{\ell\ell}$, $\omega=4\Omega$)

to the motion (c_{55},c_{52},c_{22}) and an order of magnitude or more larger than those obtained by Hylarides and van Gent. In view of the large differences in results, we have made no attempt at this time to develop lifting-surface corrections for the lateral results with the forces perpendicular to the motion.

Typical Pressure Distributions. Pressure distributions obtained using the lifting-line theory for the David Taylor NSRDC 4381 propeller described in Section 7 are shown in Fig. 6 and Fig. 7 for a unit heave and a unit pitch, respectively. These typical results are for the x=.65 nondimensional radius at a vibration frequency which corresponds to a reduced frequency k=2.91. As noted in Section 3.1, the finite leading edge pressure is an equivalent value derived so that the integrated pressure produces the necessary total lift. From Fig. 6, the heave produced lift is predominantly positive real with a phase angle of about -20° with respect to the heave displacement $\delta n(0)$. From Fig. 7, the pitch produced lift is predominantly negative imaginary with a phase angle of about -105° with respect to the pitch displacement $\delta n'(0)$. This is consistent with eq. (80) and eq. (82) since C(2.9) = .507-.041i.

Comparison with Quasi-Steady Damping. Long[16] suggests that a quasi-steady assumption should be used to obtain the torsional/axial damping. This approach has been used by Archer for torsional damping.[14,15] This approach assumes that the steady-state propeller characteristics (K_T,K_Q,J) are also applicable to a vibrating propeller. Under this assumption, the following results can be derived for the nondimensional damping coefficients:

$$c_{44} = \frac{1}{2\pi}\left(2K_Q - J\,\frac{dK_Q}{dJ}\right) , \tag{97}$$

$$c_{41} = -\frac{dK_Q}{dJ} , \tag{98}$$

$$c_{11} = -\frac{dK_T}{dJ} . \tag{99}$$

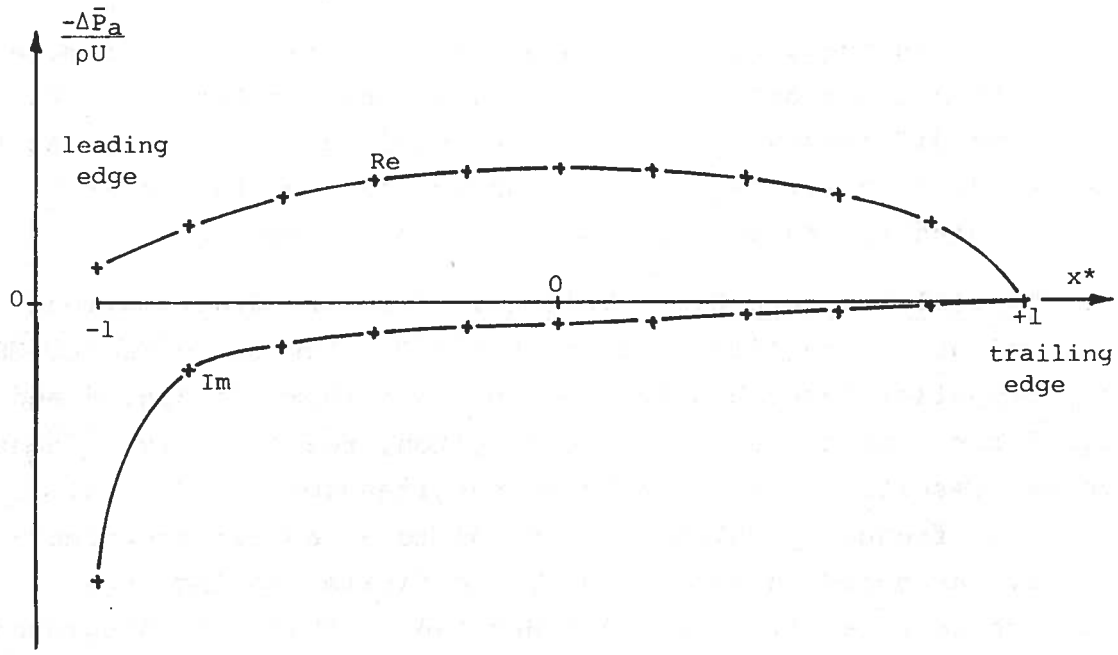


Figure 6. Typical Pressure Distribution Due to a Unit Heave

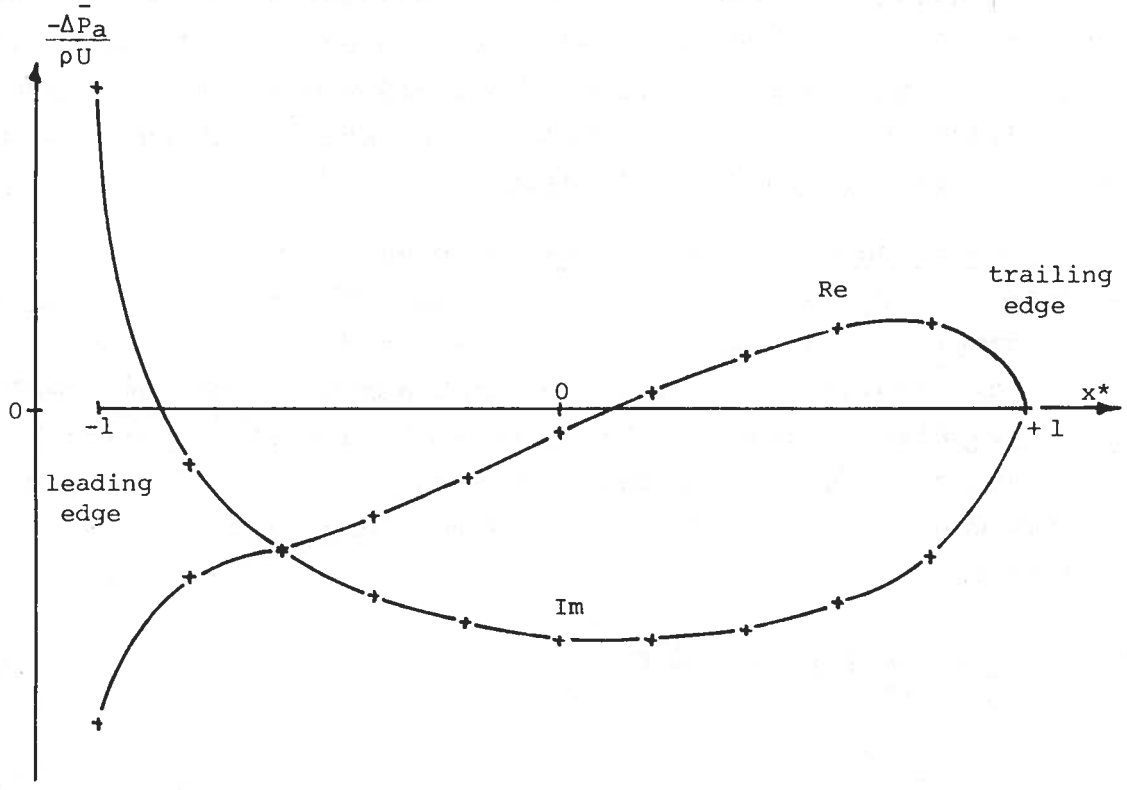


Figure 7. Typical Pressure Distribution Due to a Unit Pitch

These equations were used to estimate the quasi-steady damping coefficients of a B4-70-80 propeller at $J_{\ell\ell}$ (lightly loaded, K_T=0) using characteristics given by van Lammeren, van Manen, and Oosterveld.[22] This yields c_{44} = .00936, c_{41} = -.0537, c_{11} = .491. For this particular case the torsional damping is reasonably good, the axial damping is 30 percent low, and the velocity coupling is 40 percent low in magnitude. A comparison of quasi-steady results with lifting-surface results for all nine of the B4 propellers analyzed by Hylarides and van Gent is shown in Table 6. The damping is generally underpredicted by up to a factor of 2 compared with the lifting-surface results when the quasi-steady assumption is used.

propeller	torsional damping c_{44}		velocity coupling c_{41}		axial damping c_{11}	
	quasi-steady	lifting-surface[6]	quasi-steady	lifting-surface[6]	quasi-steady	lifting-surface[6]
B4-40-50	.00473	.00395	-.0343	-.0496	.348	.624
B4-40-80	.00934	.00965	-.0533	-.0758	.423	.595
B4-40-120	.01910	.01980	-.0806	-.1040	.462	.542
B4-70-50	.00459	.00530	-.0321	-.0666	.462	.837
B4-70-80	.00936	.01180	-.0537	-.0926	.491	.727
B4-70-120	.01970	.02300	-.0852	-.1200	.476	.630
B4-100-50	.00598*	.00463	-.0400*	-.0582	.492*	.853
B4-100-80	.01070	.01090	-.0653	-.0854	.463	.671
B4-100-120	.02040	.02230	-.0923	-.1170	.465	.612

Key: B4-70-80 4 blades A_e/A_o .70, P/D (x = 0.7) = .80

*estimated for P/D = .60

Table 6. Comparison of Quasi-Steady and Lifting-Surface Torsional/Axial Damping

4.2. Sensitivity Studies

To establish the effect of various propeller parameters on the added mass and damping in all modes of vibration, a number of sensitivity studies were performed using the lifting-line theory option of the PRAMAD program. For these studies, Wageningen B-series propellers with a nominal expanded area ratio of .75 and a pitch-diameter ratio of .90 were analyzed. The reference was the four-bladed B4-75-90 propeller.

Effect of Advance Coefficient. Table 7 presents results for the B4-75-90 propeller vibrating at its blade rate while operating at 25, 50, 75, and 100 percent of the lightly-loaded advance coefficient $J_{\ell\ell}(K_T=0)$. Most propellers are designed to operate at about .75 $J_{\ell\ell}$ which gives an advance coefficient slightly below that yielding the maximum open water efficiency. In general, the added mass terms, which are more strongly dependent on the noncirculatory contribution, are less sensitive than the damping terms to changes in the advance coefficient. Comparing the lightly-loaded results and the most typical $.75J_{\ell\ell}$ results, the added mass terms change very little. The lateral added mass results for forces perpendicular to the motion change significantly but the overall magnitudes are negligibly small. The torsional/axial and lateral:parallel damping results are 7 to 34 percent higher at the normal operating advance coefficient. The lateral:perpendicular damping results are 9 to 15 percent lower at the normal operating advance coefficient.

Effect of Vibration Frequency. Propulsion shafting vibration is primarily of concern at blade rate due to the large propeller excitation at this frequency. The propeller excitation at twice and three times blade rate are usually much smaller. Diesel engine propelled ships can also experience significant torsional excitation at integer multiples of the rotation frequency (two-stroke cycle engines) or integer multiples of one-half the rotation frequency (four-stroke cycle engines). Lateral vibration may be excited at the rotation frequency due to system imbalance. The PRAMAD program, therefore, accepts the vibration frequency and the rotation frequency as independent inputs.

component	$.25J_{\ell\ell}$	$.50J_{\ell\ell}$	$.75J_{\ell\ell}$	lightly-loaded $J_{\ell\ell}$
torsional/axial				
m_{44}	.00199	.00225	.00226	.00225
m_{41}	-.01431	-.01608	-.01620	-.01609
m_{11}	.10306	.11546	.11622	.11528
c_{44}	.02074	.01470	.01209	.01058
c_{41}	-.14653	-.10394	-.08513	-.07419
c_{11}	1.03669	.73572	.60015	.52009
lateral:parallel				
m_{55}	.00422	.00519	.00534	.00540
m_{52}	.00669	.00764	.00780	.00783
m_{22}	.01779	.01878	.01887	.01861
c_{55}	.06314	.04930	.04354	.04058
c_{52}	.07149	.05411	.04479	.03846
c_{22}	.13142	.10339	.07980	.05953
lateral:perpendicular				
m_{65}	.00028	.00002	-.00021	-.00040
m_{62}	.00024	-.00000	-.00028	-.00053
m_{32}	.00066	.00047	-.00015	-.00086
c_{65}	-.02859	-.04613	-.05219	-.05740
c_{62}	-.03067	-.04760	-.05532	-.06207
c_{32}	-.05419	-.08188	-.10239	-.12001

Table 7. Effect of Advance Coefficient J on Results for B4-75-90 Propeller ($\omega=4\Omega$)

Table 8 presents results for the B4-75-90 propeller operating at .75 $J_{\ell\ell}$ while vibrating at its rotation frequency and its blade rate and first two harmonics. In general, the added mass terms increase in magnitude with frequency while the damping terms decrease with frequency. The lateral damping terms for forces parallel to the motion are particularly sensitive to the frequency of vibration. The lateral added mass terms exhibit a sign reversal at low frequency with the m_{65}, m_{62}, and m_{32} terms reaching significant magnitudes at the rotation frequency.

Effect of Blade Number. Table 9 presents results for the blade-rate vibration ($\omega=Z\Omega$) for 4-, 5-, 6-, 7-bladed BZ-75-90 propellers operating at .75 $J_{\ell\ell}$. Since all propellers have the same expanded area ratio, these results indicate the effect of changes in the blade aspect ratio. The lifting-line theory would be expected to be more valid for the higher blade numbers. The analyses were performed for the same rotation frequency, so the results represent vibration at different frequencies. Comparing Tables 8 and 9, the trends can be seen to be opposite. Thus, the effect of the increasing aspect ratio dominates the influence of the increasing frequency as the blade number is increased. The added mass decreases with blade number. The torsional/axial and lateral:parallel damping results increase with blade number while the lateral:perpendicular results show an opposite trend.

Effect of Blade Skew. The Wageningen B-Series propellers are without skew. Table 10 presents the results for the blade-rate vibration of a B4-75-90 propeller operating at .75 $J_{\ell\ell}$ when it is redesigned with various linear skew distributions. Results are shown for a tip skew θ_S of 25, 50, 75, and 100 percent where 100 percent places the tip of one blade radially over the root of the adjacent blade. Positive skew is defined opposite to the direction of rotation. The 100 percent tip skew is $2\pi/Z = \pi/2$ radians on the 4-bladed propeller. The added mass results show little dependence on skew. The damping results either increase with skew or increase and then decrease with increasing skew. Maximum torsional/axial damping is exhibited by the design with a 50 percent skew distribution.

component	rotation rate $\omega = \Omega$	blade rate $\omega = 4\Omega$	$\omega = 8\Omega$	$\omega = 12\Omega$
torsional/axial				
m_{44}	.00177	.00226	.00237	.00242
m_{41}	-.01264	-.01620	-.01696	-.01726
m_{11}	.09055	.11622	.12157	.12370
c_{44}	.01319	.01209	.00806	.00684
c_{41}	-.09305	-.08513	-.05701	-.04851
c_{11}	.65733	.60015	.40377	.34441
lateral: parallel				
m_{55}	-.00207	.00534	.00571	.00586
m_{52}	-.00017	.00780	.00818	.00832
m_{22}	-.00151	.01887	.01970	.01999
c_{55}	.07579	.04354	.02343	.01249
c_{52}	.07956	.04479	.02325	.01389
c_{22}	.15027	.07980	.04547	.03260
lateral: perpendicular				
m_{65}	.00304	-.00021	-.00035	-.00029
m_{62}	.00225	-.00028	-.00042	-.00065
m_{32}	.00342	-.00015	-.00072	-.00065
c_{65}	-.05907	-.05219	-.04187	-.03894
c_{62}	-.05754	-.05532	-.04444	-.04187
c_{32}	-.11081	-.10239	-.08070	-.07684

Table 8. Effect of Vibration Frequency on Results for B4-75-90 Propeller ($.75J_{\ell\ell}$)

component	Z = 4	Z = 5	Z = 6	Z = 7
torsional/axial				
m_{44}	.00226	.00187	.00162	.00140
m_{41}	-.01620	-.01334	-.01132	-.00978
m_{11}	.11622	.09570	.07900	.06830
c_{44}	.01209	.01530	.01744	.01921
c_{41}	-.08513	-.10800	-.12173	-.13410
c_{11}	.60015	.76305	.84983	.93620
lateral: parallel				
m_{55}	.00534	.00440	.00374	.00324
m_{52}	.00780	.00646	.00551	.00479
m_{22}	.01887	.01551	.01401	.01217
c_{55}	.04354	.05160	.05632	.06062
c_{52}	.04479	.05620	.06219	.06789
c_{22}	.07980	.09910	.11146	.12170
lateral: perpendicular				
m_{65}	-.00021	.00009	.00016	.00017
m_{62}	-.00028	.00006	.00012	.00014
m_{32}	-.00015	.00033	.00031	.00029
c_{65}	-.05219	-.04483	-.03853	-.03296
c_{62}	-.05532	-.04798	-.04171	-.03597
c_{32}	-.10239	-.08635	-.07451	-.06408

Table 9. Effect of Number of Blades Z on Results for BZ-75-90 Propellers ($.75J_{\ell\ell}$, blade rate $\omega=Z\Omega$)

component	θ_s=0%	θ_s=25%	θ_s=50%	θ_s=75%	θ_s=100%
torsional/axial					
m_{44}	.00226	.00229	.00231	.00231	.00232
m_{41}	-.01620	-.01641	-.01655	-.01655	-.01655
m_{11}	.11622	.11773	.11869	.11860	.11850
c_{44}	.01209	.01338	.01416	.01356	.01288
c_{41}	-.08513	-.09410	-.09930	-.09478	-.08972
c_{11}	.60015	.66251	.69672	.66238	.62482
lateral: parallel					
m_{55}	.00534	.00535	.00547	.00565	.00587
m_{52}	.00780	.00790	.00796	.00804	.00815
m_{22}	.01887	.01936	.01935	.01912	.01883
c_{55}	.04354	.04891	.05525	.05738	.05825
c_{52}	.04479	.05170	.05623	.05540	.05416
c_{22}	.07980	.08454	.07671	.05922	.04352
lateral: perpendicular					
m_{65}	-.00021	-.00007	-.00000	-.00016	-.00040
m_{62}	-.00028	.00018	.00050	.00059	.00064
m_{32}	-.00015	-.00011	-.00050	-.00127	-.00187
c_{65}	-.05219	-.05318	-.05652	-.06204	-.07231
c_{62}	-.05532	-.05522	-.05463	-.05512	-.05801
c_{32}	-.10239	-.11101	-.11012	-.10548	-.09908

Table 10. Effect of Linear Skew Distribution with Tip Skew θ_s on Results for B4-75-90 Propeller ($.75J_{\ell\ell}$, $\omega=4\Omega$)

5. Lifting-Surface Corrections

As noted in Sections 3 and 4, our PRAMAD program utilizes either two-dimensional thin foil theory or lifting-line theory to establish the chordwise pressure distributions needed in eq. (49) and eqs. (55) through (58). The program can calculate the added mass and damping properties for any conventional propeller geometry. Our lifting-line results were compared with the lifting-surface results obtained by Hylarides and van Gent[6] for a four-bladed B4-70-80 Wageningen B-Series propeller in Table 5. In order to improve the practical utility of our program pending the incorporation of a lifting-surface theory option, we have used the results presented by Hylarides and van Gent to develop approximate lifting-surface corrections for our lifting-line results for the torsional/axial coefficients and the lateral results for forces parallel to the motion. Because of the fundamental inconsistencies between our results and those obtained by Hylarides and van Gent for the forces perpendicular to the motion, no attempt has been made to develop lifting-surface corrections for these coefficients.

5.1. Definition of Corrections

The results presented by Hylarides and van Gent are limited to the blade-rate vibration of a matrix of nine four-bladed B-Series propellers operating at a lightly-loaded advance coefficient $J_{\ell\ell}$. The nine propellers have combinations of three expanded area ratios A_e/A_o and three pitch-diameter ratios P/D. These two parameters are the two most important in determining the nondimensional added mass and damping. The Dutch results are presented in nondimensional form so the effects of propeller size and rotation rate are removed. The lifting-line option of the PRAMAD program was used to duplicate the analyses performed by Hylarides and van Gent. The ratio of the lifting-surface results to the lifting-line results for the torsional/axial and lateral: parallel coefficients for the nine propellers are given in Table 11. The ratios are closest to one for the small expanded area ratio (large aspect ratio) propellers where the lifting-line

model is expected to be most valid. As the expanded area ratio of the propeller increases, the ratio decreases for the added mass coefficients and increases for the damping coefficients. Our lateral damping coefficients c_{52} and c_{22} approach zero and even reverse sign for propellers with the two highest pitch ratios and the nominal expanded area ratio of one. The propeller blades have a geometric aspect ratio,

$$AR = \frac{\text{span}}{\text{mean chord}} = \frac{\text{span}^2}{\text{blade area}} \quad , \tag{100}$$

of only .893 in this case and the lifting-line modeling would be expected to be marginal.

For use in the PRAMAD program, we have defined lifting-surface corrections in the same spirit as presented for camber, ideal angle of attack due to loading, and ideal angle of attack due to thickness by Morgan, Silovic, and Denny.[23] For each of the twelve torsional/axial and lateral:parallel coefficients, a lifting-surface correction is defined as,

$$LSC(P/D,AR) = \frac{\text{lifting-surface result}}{\text{lifting-line result from PRAMAD}} \quad , \tag{101}$$

so that the PRAMAD lifting-line result can be multiplied by this approximate correction to yield on improved estimate. For the correction to be appropriate to propellers of any blade number, the appropriate parameter is the blade aspect ratio and not the expanded area ratio as this most closely correlates with the validity of the lifting-line modeling. For convenient computer implementation we desired a regression equation for each of the twelve lifting-surface corrections as functions of P/D and AR. The basic data for these corrections are the ratios given in Table 11 however these data could not be used alone. The PRAMAD program could be expected to be applied to propellers with up to seven blades and expanded area ratios as low as perhaps 0.5. At this extreme, the propeller blade would have a geometric aspect ratio of 3.125 which is well beyond the range of the data in Table 11. A direct regression of the data in Table 11 could not be expected

propeller	B4-40-50	B4-40-80	B4-40-120	B4-70-50	B4-70-80	B4-70-120	B4-100-50	B4-100-80	B4-100-120
nominal A_e/A_o	.400	.400	.400	.700	.700	.700	1.000	1.000	1.000
P/D @ x = 0.7	.500	.800	1.200	.500	.800	1.200	.500	.800	1.200
aspect ratio AR	2.232	2.232	2.232	1.275	1.275	1.275	.893	.893	.893
torsional/axial									
m_{44}	.80293	.83587	.86247	.73941	.72797	.73198	.59200	.59195	.60484
m_{41}	.77817	.81493	.84473	.71784	.70956	.71941	.57705	.57730	.59335
m_{11}	.75450	.79431	.82395	.69494	.69079	.70228	.55662	.56257	.57999
c_{44}	.95501	.99973	1.0050	1.1360	1.2457	1.2526	1.5668	3.8420	6.1572
c_{41}	.94096	.98771	.99775	1.1238	1.2360	1.2424	1.5753	4.0183	6.7206
c_{11}	.92749	.97396	.94914	1.1108	1.2263	1.2396	1.8489	4.2578	7.4760
lateral:parallel									
m_{55}	.0503	.99785	.95643	.72971	.73733	.73093	.57996	.61461	.62471
m_{52}	.93260	.91857	.89133	.70048	.72613	.74064	.57434	.63879	.67817
m_{22}	.83181	.80926	.80293	.60391	.62715	.65537	.30704	.53424	.57308
c_{55}	.82900	.81446	.82132	.75675	.86197	.94389	1.3166	2.2196	3.4220
c_{52}	.94383	.90361	.93048	.98600	1.0957	1.2933	2.6972	21.2039*	-8.0794*
c_{22}	.95637	.88561	.92674	1.1127	1.1809	1.4208	6.5398	-3.8052*	-2.7704*

*data not used in regression

Table 11. Ratio of Lifting-Surface to Lifting-Line Results used in Regression Analyses for Lifting Surface Corrections

to yield reliable results for aspect ratios above 2.232, the highest value for which data was available. Using the knowledge that all the corrections should approach 1.0 at high aspect ratio, the data of Table 11 was, therefore, manually extended with extrapolated data points at an aspect ratio of 4.0 as shown in Table 12. The extended data set given in Tables 11 and 12 could then be used to obtain regression equations which would be reliable over the range of expected use.

The lifting-surface corrections based on the data in Table 11 are strictly limited to blade-rate vibration of unskewed, four-bladed Wageningen B-Series propellers operating at $J_{\ell\ell}$. By expressing the corrections as functions of geometric aspect ratio AR, the corrections can be reasonably extended to propellers of other blade numbers. The ratio of the lifting-surface to lifting-line results would not be expected to be strongly dependent on vibration frequency, advance coefficient, skew, and detailed propeller geometry so the resulting corrections should be valid for engineering purposes. The lifting-line results and lifting-surface corrections are, however, output separately by the PRAMAD program so each user can determine whether or not to utilize the correction.

5.2. Regression Analyses for Corrections

To obtain the regression equations for the twelve lifting-surface corrections based on the data in Tables 11 and 12, the Michigan Interactive Data Analysis System (MIDAS)[24] was utilized. As one of its many capabilities, this program performs multiple linear regression using regression equations of the user's choice. The resulting regression equations for the lifting-surface corrections to the torsional/axial added mass and damping coefficients are as follows:

$$LSC(m_{44}) = .61046 + .34674(P/D) + .60294(AR)^{-1} - .56159(AR)^{-2} - .80696(P/D)(AR)^{-1} + .45806(P/D)(AR)^{-2} \quad , \qquad (102)$$

$$LSC(m_{41}) = .65348 + .28788(P/D) + .39805(AR)^{-1} - .42582(AR)^{-2} - .61189(P/D)(AR)^{-1} + .33373(P/D)(AR)^{-2} \qquad (103)$$

component	aspect ratio AR = 4		
	P/D = 0.50	P/D = 0.80	P/D = 1.20
torsional/axial			
m_{44}	.810	.870	.930
m_{41}	.805	.855	.910
m_{11}	.760	.810	.850
c_{44}	1.000	1.000	1.000
c_{41}	1.000	1.000	1.000
c_{11}	1.000	1.000	1.000
lateral: parallel			
m_{55}	1.000	1.000	1.000
m_{52}	1.000	1.000	1.000
m_{22}	.980	.950	.915
c_{55}	1.000	1.000	1.000
c_{52}	1.000	1.000	1.000
c_{22}	1.000	1.000	1.000

Table 12. Extrapolated Data Used in Regressions for Lifting-Surface Corrections

$$LSC(m_{11}) = .61791 + .23741(P/D) + .42253(AR)^{-1} - .43911(AR)^{-2} - .46697(P/D)(AR)^{-1} + .25124(P/D)(AR)^{-2} \quad , \tag{104}$$

$$LSC(c_{44}) = .82761 - .41165(AR)^{-2} + 1.2196(P/D)(AR)^{-1} + 6.3993(AR)^{-3} - 13.803(P/D)(AR)^{-3} - 6.9091(AR)^{-4} + 15.594(P/D)(AR)^{-4} \quad , \tag{105}$$

$$LSC(c_{41}) = .80988 - .63077(AR)^{-2} + 1.3909(P/D)(AR)^{-1} + 7.5424(AR)^{-3} - 15.689(P/D)(AR)^{-3} - 8.0097(AR)^{-4} + 17.665(P/D)(AR)^{-4} \quad , \tag{106}$$

$$LSC(c_{11}) = .82004 - .67190(AR)^{-2} + 1.3913(P/D)(AR)^{-1} + 7.7476(AR)^{-3} - 16.807(P/D)(AR)^{-3} - 8.2798(AR)^{-4} + 19.121(P/D)(AR)^{-4} \quad . \tag{107}$$

To improve the effectiveness of the equations over the range of expected use, the data values for the ratio of the coefficients c_{52} and c_{22} for the B4-100-80 and B4-100-120 propellers shown in Table 11 were excluded from the associated analyses. The resulting regression equations for the lifting-surface corrections to the lateral:parallel added mass and damping coefficients are as follows:

$$LSC(m_{55}) = - .13964 + .89760(AR) + .34086(P/D) - .15307(AR)^2 - .36619(P/D)(AR) + .70192(P/D)(AR)^2 \quad , \tag{108}$$

$$LSC(m_{52}) = .0010398 + .66020(AR) + .39850(P/D) - .10261(AR)^2 - .34101(P/D)(AR) + .060368(P/D)(AR)^2 \quad , \tag{109}$$

$$LSC(m_{22}) = .78170 + .36153(AR-2) - .19256(P/D)(AR-2) + .17908(P/D)(AR-2)^2 - .16110(AR-2)^2 - .061038(P/D)^2(AR-2) \quad , \tag{110}$$

$$LSC(c_{55}) = .78255 + .061046(AR) - 2.5056(AR)^{-3} + 1.6426(AR)^{-4} + 1.8440(P/D)(AR)^{-4} \quad , \tag{111}$$

$$LSC(c_{52}) = 1.0121 + .73647(AR)^{-2} - 3.8691(AR)^{-3} - 1.5129(P/D)(AR)^{-3} + 3.0614(AR)^{-4} + 3.0984(P/D)(AR)^{-4} \quad , \tag{112}$$

$$LSC(c_{22}) = .84266 + 6.7849(AR)^{-2} + .12809(P/D)(AR)^{-1} - 21.030(AR)^{-3} - 3.3471(P/D)(AR)^{-3} + 15.842(AR)^{-4} + 5.1905(P/D)(AR)^{-4} \quad . \qquad (113)$$

The multiple correlation coefficients R and the standard error SE for each of the regression equations (102) through (113) are shown in Table 13. The multiple correlation coefficient squared is the coefficient of multiple determination which indicates the proportion of the variation explained by the linear regression equation. The standard error is the square root of the error sum of squares SSE which is the sum of the squares of the residuals at the data points.[25] The regression equations were also evaluated for a B4-70-80 propeller and these results are shown in Table 13 with the data values from Table 11. The errors in the regression equations are less than 4 percent which is acceptable considering the approximations inherent in their development and use.

correction on component	mult. R	standard error	B4-70-80 Test		percent error
			regression	data	
torsional/axial					
m_{44}	.99935	.0056	.7345	.7280	0.9%
m_{41}	.99918	.0062	.7144	.7096	1.0%
m_{11}	.99748	.0099	.6999	.6908	1.0%
c_{44}	.99865	.1232	1.2050	1.2457	3.3%
c_{41}	.99921	.1038	1.1940	1.2360	3.4%
c_{11}	.99983	.0550	1.1850	1.2263	3.4%
lateral: parallel					
m_{55}	.99829	.0143	.7466	.7373	1.3%
m_{52}	.99914	.0088	.7256	.7261	0.1%
m_{22}	.98357	.0488	.6503	.6272	3.7%
c_{55}	.99627	.0856	.8309	.8620	3.6%
c_{52}	.99978	.0169	1.1100	1.0957	1.3%
c_{22}	.99991	.0405	1.2229	1.1809	3.6%

Table 13. Regression Data on Lifting-Surface Corrections

6. Preliminary Design Equations for Wageningen B-Series Propellers

One of the initial objectives of this research was to develop an improved capability to estimate the added mass and damping of a propeller in all modes of vibration early in preliminary design when vibration analyses must first be performed. At that point, the detailed design of the propeller is not available but its overall parameters such as diameter, blade number, rotation rate, expanded area ratio, and pitch are usually known. The Wageningen B-Series is the most commonly used standard series for commercial marine propellers so we used the PRAMAD program to calculate the added mass and damping of matrices of nine B-Series propellers having 4, 5, 6, and 7 blades. The calculations were performed using lifting-line theory with the propellers at an advance coefficient of $.75J_{\ell\ell}$ and a blade-rate vibration frequency. For each blade number, the propellers had expanded area ratios of .50, .75, and 1.0 and pitch-diameter ratios of 0.6, 0.9 and 1.2. The resulting nondimensional added mass and damping coefficients for each blade number Z were then fit by multiple linear regression equations using the MIDAS program.[24] The regression equations have the form:

$$\text{coefficient } m_{ij} \text{ or } c_{ij} = C_1 + C_2(A_e/A_o) + C_3(P/D) + C_4(A_e/A_o)^2 + C_5(P/D)^2 + C_6(A_e/A_o)(P/D) \quad , \qquad (114)$$

which is suitable for either calculator evaluation or computer implementation.

The regression equation coefficients for the added mass and damping coefficients for Wageningen B-Series propellers having 4, 5, 6, and 7 blades are given in Tables 14, 15, 16, and 17, respectively. The multiple correlation coefficient R and standard error SE are also shown for each regression equation. These equations represent lifting-line results so the lifting-surface corrections presented in Section 5 could be multiplied onto the torsional/axial and lateral:parallel results to improve the estimates. For the B-Series the blade geometric aspect ratio AR needed in eqs. (102) through (113) is given by:

$$AR = \frac{.22087\ Z}{A_e/A_o} \quad , \tag{115}$$

where Z is the blade number. The results given in Tables 7, 8, and 10 could also be used to correct the estimates for advance coefficient, vibration frequency, and skew, respectively, if desired. Sample results obtained using these equations are given in the next section.

component \ parameter	constant	$\frac{A_e}{A_o}$	$\frac{P}{D}$	$\left[\frac{A_e}{A_o}\right]^2$	$\left[\frac{P}{D}\right]^2$	$\frac{A_e}{A_o} * \frac{P}{D}$	mult.R	standard error
torsional/axial								
m_{44}	.30315E-2	-.80782E-2	-.40731E-2	.34170E-2	.43437E-3	.99715E-2	.99972	.00007
m_{41}	.12195E-2	.17664E-1	-.85938E-2	-.23615E-1	.94301E-2	-.26146E-1	.99970	.00038
m_{11}	-.62948E-1	.17980	.58719E-1	.17684	-.21439E-2	-.15395	.99993	.00132
c_{44}	-.35124E-1	.81977E-1	.32644E-1	-.41863E-1	.60813E-2	-.37170E-1	.99322	.00107
c_{41}	.13925	-.48179	-.14175	.27711	-.94311E-2	.17407	.99371	.00434
c_{11}	.32017	.29375E+1	-.90814	-.19719E+1	.53868	-.65404	.99644	.02503
lateral:parallel								
m_{55}	-.26636E-2	.61911E-2	.26565E-2	.77133E-2	-.66326E-3	-.40324E-2	.99996	.00004
m_{52}	-.19644E-2	-.47339E-2	.45533E-2	.89144E-2	-.44606E-2	.11823E-1	.99973	.00017
m_{22}	.17699E-1	-.59698E-1	-.18823E-1	.29066E-1	-.33316E-2	.73554E-1	.99973	.00052
c_{55}	-.63518E-2	.22851	-.31365E-1	-.14332	.25084E-1	-.49546E-1	.99629	.00142
c_{52}	-.11690	.36582	.10076	-.21326	.18676E-3	-.12515	.99342	.00312
c_{22}	-.35968	.87537	.29734	-.47961	.14001E-1	-.33732	.98930	.01017
lateral:perpendicular								
m_{65}	.12333E-3	.35676E-2	-.35561E-3	-.36381E-2	.65794E-3	-.17943E-2	.99961	.00003
m_{62}	-.17250E-2	.64561E-2	.19195E-2	-.40546E-2	.40439E-3	-.47506E-2	.99928	.00006
m_{32}	-.99403E-2	.23315E-1	.10895E-1	-.11360E-1	-.71528E-3	-.15713E-1	.99718	.00024
c_{65}	.59756E-1	-.18982	-.17653E-1	.82400E-1	.61804E-2	-.80790E-2	.99962	.00074
c_{62}	.78572E-1	-.18627	-.37105E-1	.11053	.17847E-1	-.55900E-1	.99876	.00164
c_{32}	.14397	-.32322	-.15348E-1	.24992	.14289E-1	-.21254	.99771	.00564

Table 14. Regression Equation Coefficients for B4 Propellers ($E \pm N = \times 10^{\pm N}$)

component \ parameter	constant	$\frac{A_e}{A_o}$	$\frac{P}{D}$	$\left[\frac{A_e}{A_o}\right]^2$	$\left[\frac{P}{D}\right]^2$	$\frac{A_e}{A_o} * \frac{P}{D}$	mult.R	standard error
torsional/axial								
m_{44}	.27835E-2	-.71650E-2	-.37301E-2	.30526E-2	.46275E-3	.85327E-2	.99966	.00006
m_{41}	-.26829E-3	.17208E-1	-.55064E-2	-.21012E-1	.72960E-2	-.22840E-1	.99971	.00031
m_{11}	-.47372E-1	.13499	.43428E-1	.15666	.41444E-3	-.12404	.99993	.00107
c_{44}	-.30935E-1	.69382E-1	.27392E-1	-.37293E-1	.63542E-2	-.21635E-1	.99536	.00099
c_{41}	.14558	-.44319	-.17025	.24558	.14798E-1	.12226	.99223	.00371
c_{11}	.16202	.30392E+1	-.59068	-.17372E+1	.37998	-.71363	.99674	.01816
lateral:parallel								
m_{55}	-.18541E-2	.40694E-2	.20342E-2	.72761E-2	-.47031E-3	-.33269E-2	.99998	.00003
m_{52}	-.20455E-3	-.73445E-2	.26857E-2	.95299E-2	-.34485E-2	.10863E-1	.99974	.00014
m_{22}	.17180E-1	-.54519E-1	-.17894E-1	.27151E-1	-.19451E-2	.62180E-1	.99967	.00048
c_{55}	-.25532E-2	.20018	-.22067E-1	-.10971	.18255E-1	.43517E-1	.99569	.00106
c_{52}	-.98481E-1	.28632	.10154	-.15975	-.10484E-1	-.79238E-1	.98925	.00244
c_{22}	-.27180	.61549	.24132	-.33370	.10475E-1	-.19101	.99075	.00798
lateral:perpendicular								
m_{65}	-.51073E-3	.36044E-2	.12804E-3	-.30064E-2	.25624E-3	-.11174E-2	.99948	.00002
m_{62}	-.18142E-2	.56442E-2	.15906E-2	-.35420E-2	.72381E-4	-.27848E-2	.99832	.00005
m_{32}	-.68895E-2	.16524E-1	.62244E-2	-.87754E-2	-.38255E-3	-.82429E-2	.99301	.00019
c_{65}	.33407E-1	-.99682E-1	-.68219E-2	.22669E-1	.56635E-2	-.22639E-1	.99994	.00035
c_{62}	.33507E-1	-.66800E-1	-.73992E-2	.33112E-1	.13061E-1	-.80862E-1	.99981	.00073
c_{32}	.15158E-1	-.10109E-1	.63232E-1	.50161E-1	.35720E-2	-.27201	.99983	.00161

Table 15. Regression Equation Coefficients for B5 Propellers ($E \pm N = \times 10^{\pm N}$)

component \ parameter	constant	$\frac{A_e}{A_o}$	$\frac{P}{D}$	$\left[\frac{A_e}{A_o}\right]^2$	$\left[\frac{P}{D}\right]^2$	$\frac{A_e}{A_o} * \frac{P}{D}$	mult.R	standard error
torsional/axial								
m_{44}	.23732E-2	-.62877E-2	-.30606E-2	.27478E-2	.29060E-3	.73650E-2	.99965	.00006
m_{41}	-.17748E-3	.14993E-1	-.51316E-2	-.18451E-1	.64733E-2	-.19096E-1	.99968	.00028
m_{11}	-.39132E-1	.10862	.37308E-1	.13359	-.33222E-3	-.10387	.99994	.00088
c_{44}	-.27873E-1	.61760E-1	.23242E-1	-.35004E-1	.70046E-2	-.11641E-1	.99696	.00092
c_{41}	.14228	-.41189	-.17770	.22644	.26626E-1	.83269E-1	.99456	.00334
c_{11}	.11113	.29831E+1	-.44133	-.15696E+1	.28560	-.66976	.99749	.01465
lateral:parallel								
m_{55}	-.16341E-2	.33153E-2	.19742E-2	.64129E-2	-.52004E-3	-.29555E-2	.99997	.00003
m_{52}	-.40692E-4	-.68309E-2	.24412E-2	.86298E-2	-.30852E-2	.92581E-2	.99970	.00013
m_{22}	.13668E-1	-.46198E-1	-.12970E-1	.24376E-1	-.29068E-2	.52775E-1	.99967	.00042
c_{55}	.63116E-3	.18370	-.17663E-1	-.92593E-1	.13964E-1	-.37740E-1	.99714	.00086
c_{52}	-.85805E-1	.24006	.10020	-.13176	-.16091E-1	-.52217E-1	.99214	.00202
c_{22}	-.24147	.52269	.23271	-.28526	.68691E-2	.13848	.99384	.00711
lateral:perpendicular								
m_{65}	-.46261E-3	.27610E-2	.11516E-3	-.21853E-2	.13978E-3	-.66927E-3	.99919	.00092
m_{62}	-.13553E-2	.40432E-2	.11434E-2	-.25422E-2	-.10052E-4	-.17182E-2	.99742	.00004
m_{32}	-.45682E-2	.10797E-1	.41254E-2	-.57931E-2	-.33138E-3	-.50724E-2	.99178	.00013
c_{65}	.21438E-1	-.54389E-1	-.73262E-2	-.54776E-2	.76065E-2	-.24165E-1	.99996	.00027
c_{62}	.14903E-1	-.10434E-1	-.40106E-2	-.31096E-2	.14738E-1	-.83101E-1	.99988	.00055
c_{32}	-.15579E-1	.83912E-1	.52866E-1	-.24903E-1	.93828E-2	-.24420	.99993	.00097

Table 16. Regression Coefficients for B6 Propellers ($E \pm N = \times 10^{\pm N}$)

component \ parameter	constant	$\frac{A_e}{A_o}$	$\frac{P}{D}$	$\left[\frac{A_e}{A_o}\right]^2$	$\left[\frac{P}{D}\right]^2$	$\frac{A_e}{A_o} * \frac{P}{D}$	mult.R	standard error
torsional/axial								
m_{44}	.21372E-2	-.56155E-2	-.27388E-2	.24553E-2	.26575E-3	.64805E-2	.99962	.00005
m_{41}	-.50233E-3	.13927E-1	-.41583E-2	-.16454E-1	.56027E-2	-.17030E-1	.99967	.00025
m_{11}	-.32908E-1	.88748E-1	.32596E-1	.11886	-.96860E-3	-.87831E-1	.99994	.00073
c_{44}	-.24043E-1	.51680E-1	.18585E-1	-.31175E-1	.75424E-2	-.10541E-2	.99813	.00083
c_{41}	.14003	-.37358	-.18904	.20133	.40056E-1	.45135E-1	.99705	.00300
c_{11}	.34070E-1	.29353E+1	-.24280	-.13929E+1	.17571	-.65123	.99882	.01067
lateral:parallel								
m_{55}	-.14132E-2	.26715E-2	.18052E-2	.56906E-2	-.52314E-3	-.24846E-2	.99996	.00003
m_{52}	.17646E-3	-.65252E-2	.19867E-2	.78350E-2	-.26907E-2	.82990E-2	.99969	.00012
m_{22}	.12144E-1	-.40599E-1	-.11616E-1	.21395E-1	-.24429E-2	.46140E-1	.99967	.00037
c_{55}	-.26383E-2	.17179	-.29958E-2	-.79085E-1	.50612E-2	-.33233E-1	.99904	.00055
c_{52}	-.78069E-1	.20492	.10121	-.10980	-.21608E-1	-.28950E-1	.99603	.00172
c_{22}	-.20348	.42553	.19690	-.23664	.11822E-1	-.69910E-1	.99635	.00616
lateral:perpendicular								
m_{65}	-.38383E-3	.19693E-2	.17327E-3	-.15326E-2	.26748E-4	-.36439E-3	.99913	.00001
m_{62}	-.96395E-3	.28059E-2	.80259E-3	-.17783E-2	-.37139E-4	-.10340E-2	.99649	.00002
m_{32}	-.29000E-2	.69281E-2	.24977E-2	-.37440E-2	-.12487E-3	-.30407E-2	.98983	.00008
c_{65}	.10617E-1	-.24040E-1	-.19931E-2	-.20636E-1	.60272E-2	-.26183E-1	.99995	.00028
c_{62}	.15152E-2	-.20965E-1	-.36583E-2	-.22580E-1	.11790E-1	-.80079E-1	.99990	.00048
c_{32}	-.36770E-1	.13433	.56417E-1	-.62648E-1	.74649E-2	-.22397	.99994	.00082

Table 17. Regression Equation Coefficients for B7 Propellers ($E \pm N = \times 10^{\pm N}$)

7. Design Example

As a final example, we have used the PRAMAD program to calculate the blade-rate added mass and damping of the five-bladed David Taylor NSRDC propeller 4381 which has been used by Boswell[26,27] as part of a study of the effects of skew on propeller performance. The characteristics of this propeller are given in Table 18. The propeller has an expanded area ratio of .725 and a design pitch of 1.210 at the x=0.7 nondimensional radius. The propeller is designed for uniform flow and is without rake or skew. To provide the input needed by the program, a 20 ft. diameter propeller was assumed to be turning at 110 rpm. The design advance coefficient was then used to obtain the ship speed. The propeller was modeled by MM=8 segments. The propeller section has a NACA 66 modified thickness on a NACA a=0.8 meanline. The propeller material weight density was assumed to be 0.314 lbf/in^3 and approximate hub dimensions were assumed to allow the estimation of the mass and torsional mass moment of inertia of the propeller.

The PRAMAD lifting-line theory results obtained for the DTNSRDC propeller 4381 are shown in Table 19. These results were obtained using 70.7 central processing unit (CPU) seconds on our Amdahl 470/V8 computer. The lifting-surface corrections and resulting products are also shown for the torsional/axial and lateral:parallel coefficients. To illustrate the effectiveness of the Wageningen B-Series design equations presented in Section 6 for preliminary estimates, the equations from Table 15 for a five-bladed propeller were evaluated using A_e/A_o=.725 and P/D=1.210 and these results are also shown in Table 19. Note that the 4381 propeller is not a Wageningen B-Series propeller. These estimates are generally within 10 percent; coefficients c_{65}, c_{22}, and m_{55} show larger deviations in this particular case. The lateral:perpendicular added mass coefficients are so small that there are large percentage differences and sign changes between the B-Series equations and the specific lifting-line output for the 4381 propeller.

$Z = 5$	$A_E/A_O = .725$
$D = 20$ ft. (assumed)	$P/D_{0.7} = 1.210$
$J = .889$	$M = 17$
$V_S = 32.6$ ft/sec.	$\omega = 5\Omega = 57.596$ rad/sec.
$N = 110$ rpm (assumed)	$\Omega = 11.519$ rad/sec.
$\rho_w = 1.9905$ lbfsec^2/ft^4	$\gamma_P = 0.314$ lbf/in^3 (assumed)

x	R	C/D	V_x/V_s	θ_s	P/D	θ_R	t/D
.200	24.00	.174	1.000	.000	1.332	.000	.0434
.250	30.00	.202			1.338		.0396
.300	36.00	.229			1.345		.0358
.350	42.00	.253			1.354		.0324
.400	48.00	.275			1.358		.0294
.450	54.00	.295			1.352		.0266
.500	60.00	.312			1.336		.0240
.550	66.00	.326			1.311		.0215
.600	72.00	.337			1.280		.0191
.650	78.00	.344			1.246		.0168
.700	84.00	.347			1.210		.0146
.750	90.00	.344			1.173		.0125
.800	96.00	.334			1.137		.0105
.850	102.00	.314			1.101		.0086
.900	108.00	.280			1.066		.0067
.950	114.00	.210			1.031		.0048
1.000	120.00	.000			.995		.0029

Table 18. Characteristics of DTNSRDC Propeller 4381 Example

component	lifting-line result	lifting-surface correction	product	B5 regression equation	per cent difference*
torsional/axial					
m_{44}	.00312	.7627	.00238	.00284	9.0
m_{41}	-.01594	.7492	-.01194	-.01485	6.8
m_{11}	.08186	.7372	.06035	.07718	5.7
c_{44}	.02263	1.0100	.02285	.02323	2.7
c_{41}	-.11830	.9778	-.11568	-.12749	7.8
c_{11}	.62443	.9345	.58351	.66790	7.0
lateral:parallel					
m_{55}	.00472	.7799	.00368	.00377	20.1
m_{52}	.00782	.7746	.00606	.00721	7.8
m_{22}	.02065	.7778	.01606	.02197	6.4
c_{55}	.04484	.9496	.04258	.04676	4.3
c_{52}	.06101	1.0653	.06499	.06314	3.5
c_{22}	.15762	1.0042	.15828	.13880	11.9
lateral:perpendicular					
m_{65}	-.00018	none	n.a.	.00007	-
m_{62}	-.00017	developed		-.00070	-
m_{32}	.00046			.00022	52.2
c_{65}	-.05376			-.04677	13.0
c_{62}	-.06284			-.05829	7.2
c_{32}	-.12606			-.12269	2.7

*compared with lifting-line result

Table 19 . Results for DTNSRDC Propeller 4381

The mass of the 4381 propeller defined in Table 18 was estimated to be 148.0 lbf.sec^2/in. and the torsional mass moment of inertia was estimated to be 393,400 lbf.in.sec^2. The dimensional axial added mass m_{11} of 80.0 lbf.sec^2/in. is therefore 54.1 percent of the mass. Expressing the added mass as a percentage of the mass is a theoretically invalid practice but is done here since many still use this basis to compare designs. The lateral added mass m_{22} of 21.3 lbf.sec^2/in. is 14.4 percent of the mass. The torsional added mass moment of inertia m_{44} of 182,000 lbf.in.sec^2 is 46.3 percent of the torsional mass moment of inertia. The lateral diametral added mass moment of inertia m_{55} of 281,300 lbf.in.sec^2 is 71.5 percent of the torsional mass moment of inertia.

8. Closure

A general analysis of the added mass and damping of a marine propeller is presented in Section 2. This analysis assumes that the pressure distribution on a propeller blade due to a unit heave or unit pitch about the midchord point can be obtained using two-dimensional thin foil theory, lifting-line theory, or lifting-surface theory. The complex force integrals used to calculate the added mass and damping matrices for a propeller as defined in eq. (8) are shown in Table 3. These represent new theoretical results.

We have developed the PRAMAD computer program which evaluates the added mass and damping of a marine propeller. This program uses either the two-dimensional thin foil theory or lifting-line theory reviewed in Section 3 to calculate the pressure distribution on the blade due to a unit heave or unit pitch about the midchord point. This program incorporates the lifting-surface corrections for the torsional/axial and lateral:parallel added mass and damping coefficients presented in Section 5. The effects of advance coefficient, vibration frequency, blade number, and skew on the added mass and damping calculated using the lifting-line option in PRAMAD are illustrated by sensitivity studies presented in Section 4.

To provide improved capability to estimate the added mass and damping of a propeller in preliminary design, we have presented design equations for all the added mass and damping coefficients of 4-, 5-, 6-, and 7-bladed Wageningen B-Series propellers. These equations are of the form shown in eq. (114) and were obtained by multiple linear regression of the lifting-line theory results obtained from the PRAMAD program. The coefficients for these equations are presented in Tables 14 through 18.

We would like to acknowledge the work of University of Michigan, Department of Naval Architecture and Marine Engineering graduate student Dean L. Couphos, who did some of the early programming for the PRAMAD program.

9. References

1. ..., Technical and Research Code C5, "Acceptable Vibration of Marine Steam and Heavy-Duty Gas-Turbine Main and Auxiliary Machinery Plants," SNAME, New York, Sept., 1976.

2. ..., Rules for Building and Classing Steel Vessels, American Bureau of Shipping, New York, 1978.

3. Lewis, F.M., "Propeller-Vibration Forces," Transactions of SNAME, Vol. 71, 1963, pp. 293-326.

4. Tsakonas, S., Breslin, J., and Miller, M., "Correlation and Application of an Unsteady Flow Theory for Propeller Forces," Transactions of SNAME, Vol. 75, 1967, pp. 158-193.

5. Vorus, W.S., "A Method for Analyzing the Propeller-Induced Vibratory Forces Acting on the Surface of a Ship Stern," Transactions of SNAME, Vol. 82, 1974, pp. 136-210.

6. Hylarides, S., and van Gent, W., "Hydrodynamic Reactions to Propeller Vibrations," Proceedings of the Conference on Operational Aspects of Propulsion Shafting Systems, London, May 21-22, 1979, pp. 44-55.

7. Kane, J.R., and McGoldrick, R.T., "Longitudinal Vibrations of Marine Propulsion-Shafting Systems," Transactions of SNAME, Vol. 57, 1949, pp. 193-252.

8. Panagopoulos, E., "Design-Stage Calculations of Torsional, Axial, and Lateral Vibrations of Marine Shafting," Transactions of SNAME, Vol. 58, 1950, pp. 329-384.

9. Burrill, L.C. and Robson, W., "Virtual Mass and Moment of Inertia of Propellers," Transactions of the North East Coast Institute of Engineers and Shipbuilders, Vol. 78, 1961-1962, pp. 235-250 and D92-D104.

10. Brooks, J.E., "Added Mass of Marine Propellers in Axial Translation," David Taylor Naval Ship Research and Development Center Report 76-0079, April, 1976.

11. Jasper, N.H., "A Design Approach to the Problem of Critical Whirling Speeds of Shaft-Disk Systems," International Shipbuilding Progress, Vol. 3, No. 17, Jan., 1956, pp. 37-60.

12. Toms, A.E., and Martyn, D.K., "Whirling of Line Shafting," Transactions of the Institute of Marine Engineers, Vol. 84, 1972, pp. 176-191.

13. Vassilopoulis, L., and Bradshaw, R., "Coupled Transverse Shaft Vibrations of Modern Ships," SNAME New England Section paper, Oct., 1973.

14. Archer, S., "Contribution to Improved Accuracy in the Calculation and Measurement of Torsional Vibration Stresses in Marine Propeller Shafting," Proceedings of the Institute of Mechanical Engineers, Vol. 164, 1951, pp. 351-366.

15. Archer, S., "Torsional Vibration Damping Coefficients for Marine Propellers," Engineering, Vol. 179, No. 4659, May 13, 1955, pp. 594-598.

16. Long, C.L., "Propellers, Shafting, and Shafting System Vibration Analysis," Ch. XI in Marine Engineering, SNAME, New York, 1971.

17. Theodorsen, T., "General Theory of Aerodynamic Instability and the Mechanism of Flutter," N.A.C.A. Report 496, 1935.

18. Bisplinghoff, R.L., Ashley, H., and Halfman, R.L., Aeroelasticity, Addison-Wesley, Cambridge, Mass., 1955, pp. 251-281.

19. Brown, N.A., "Periodic Propeller Forces in Non-Uniform Flow," MIT Report No. 64-7, June, 1964.

20. Reissner, E., "On the General Theory of Thin Airfoils for Nonuniform Motion," NACA Technical Note No. 946, Aug., 1944.

21. Kerwin, J.E., and Lee, C.-S., "Prediction of Steady and Unsteady Marine Propeller Performance by Numerical Lifting-Surface Theory," Transactions of SNAME, Vol. 86, 1978, pp. 218-253.

22. van Lammeren, W.P.A., van Manen, J.D., and Oosterveld, M.W.C., "The Wageningen B-Screw Series," Transactions of SNAME, Vol. 77, 1969, pp. 269-317.

23. Morgan, W.B., Silovic, V., and Denny, S.B., "Propeller Lifting-Surface Corrections," Transactions of SNAME, Vol. 76, 1968, pp. 309-347.

24. "Elementary Statistics using MIDAS," The University of Michigan Statistical Research Lab., Second Edition, Dec., 1976.

25. Walpole, R.E., and Myers, R.H., Probability and Statistics for Engineers and Scientists, Macmillan, New York, 1978, pp. 314-338.

27. Boswell, R.J., "Design, Cavitation Performance, and Open-Water Performance of a Series of Research Skewed Propellers," David W. Taylor NSRDC Report 3339, March, 1971.

28. Cumming, R.A., Morgan, W.B., and Boswell, R.J., "Highly Skewed Propellers," Transactions of SNAME, Vol. 80, 1972, pp. 98-135.

APPENDIX A. USER'S DOCUMENTATION FOR PRAMAD Rev. 1
Sept. 2, 1980

UNIVERSITY OF MICHIGAN

DEPARTMENT OF NAVAL ARCHITECTURE AND MARINE ENGINEERING

IDENTIFICATION: PRAMAD

PROGRAMMER: Associate Professors Michael G. Parsons and William S. Vorus and Edward M. Richard, Department of Naval Architecture and Marine Engineering, The University of Michigan, 1979-1980.

SPONSOR: PRAMAD was developed under Department of Commerce, Maritime Administration, University Research Contract No. MA-3-70-SAC-B0012. Selected subroutines were previously developed under American Bureau of Shipping support. Other subroutines are from the public files of the University of Michigan Computing Center.

PURPOSE: The Propeller Added Mass and Damping (PRAMAD) program is a batch-type program which evaluates the added mass, added mass moment of enertia, and inertia coupling and the linear damping, rotational damping, and velocity coupling of a propeller in torsional, axial, and lateral vibrations. The results may be obtained using a two-dimensional foil theory or using a lifting-line theory. Approximate lifting-surface corrections for the lifting-line results are given for the torsional/axial results and for the lateral vibration results for forces and moments which are parallel to the vibratory motion. The program accepts a variety of input forms. The output is in English, SI, and nondimensional forms.

METHOD: The PRAMAD program obtains the added mass and damping of a propeller in a two step process. First, either two-dimensional foil theory or the more expensive lifting-line theory is used, at the user's option, to obtain the chordwise pressure distribution on the propeller blade due to a unit heave or unit pitch of the blade section. If results are desired only for torsional/axial motion, the results for unit pitch are not needed. The second step uses the chordwise pressure distribution results to produce the added mass and damping of the propeller. When the displacement

vector for the propeller is given by $x = [\delta_x, \delta_y, \delta_z, \theta_x, \theta_y, \theta_z]^T$, the added mass and damping matrices take the following form:

$$M_a = \begin{bmatrix} m_{11} & 0 & 0 & m_{41} & 0 & 0 \\ 0 & m_{22} & -m_{32} & 0 & m_{52} & -m_{62} \\ 0 & m_{32} & m_{33} & 0 & m_{62} & m_{52} \\ m_{41} & 0 & 0 & m_{44} & 0 & 0 \\ 0 & m_{52} & -m_{62} & 0 & m_{55} & -m_{65} \\ 0 & m_{62} & m_{52} & 0 & m_{65} & m_{55} \end{bmatrix} ; \quad C_p = \begin{bmatrix} c_{11} & 0 & 0 & c_{41} & 0 & 0 \\ 0 & c_{22} & -c_{32} & 0 & c_{52} & -c_{62} \\ 0 & c_{32} & c_{22} & 0 & c_{62} & c_{52} \\ c_{41} & 0 & 0 & c_{44} & 0 & 0 \\ 0 & c_{52} & -c_{62} & 0 & c_{55} & -c_{65} \\ 0 & c_{62} & c_{52} & 0 & c_{65} & c_{55} \end{bmatrix}.$$

The torsional/axial results are uncoupled from the lateral results and are as usual associated with the x-direction which is directed aft along the axis of rotation of the propeller. For the purposes of the PRAMAD program the y- and z-directions can be any right-handed orthogonal directions in the transverse plane of the assumed right-handed propeller.

The program calculates the torsional/axial results (m_{44}, m_{41}, m_{11}, c_{44}, c_{41}, c_{11}), the lateral results with the added mass and damping forces and moments parallel to the motion (m_{55}, m_{52}, m_{22}, c_{55}, c_{52}, c_{22}), and the lateral results with the added mass and damping forces and moments perpendicular to the motion (m_{65}, m_{62}, m_{32}, c_{65}, c_{62}, c_{32}).

DATA INPUT: Propeller geometry, rotation rate, vibration frequency, ship speed, wake data, and program control parameters are read from a single line file (or set of data cards). This line file should be prepared as shown in Table A-1. Most of this data is self-explanatory. The user uses line 2 to tell the program what output is desired and what method to use to evaluate the blade pressure distributions. On line 3 the input variable M defines the number of radii for which blade geometry and wake data will be given on the M record type 6. The program requires that the propeller blade be divided into MM circumferential segments of equal radial width as shown in Fig. A-1. An even number of segments between 4 and 10 should be used; 8 are normally used. The propeller data must be input for the M=2*MM+1 radii which define the edges and center of each segment. Thus, the use of 8 segments requires input at 17 radii where R(1) is the hub radius and R(M=17) is the tip radius.

Record Type	Input	Format	Comments
1	TITLE	18A4	Any 72 character identification for the program output.
2	ITOR	5I5	ITOR=1; for torsion results
			=0; if not
	IAXL		IAXL=1; for axial results
			=0; if not
	ILAT		ILAT=1; for lateral results
			=0; if not
	METHOD		METHOD=0; two-dimensional foil theory
			=1; lifting-line theory
	KODE		KODE=0; normal output
			=1; additional output from PRES3D
3	M	2I5	number of radii for which geometry and wake data will be given on record type 6; blade is modeled as $4 \leq MM \leq 10$ even number of segments radially; then M=2*MM+1.
	NB		number of propeller blades (Z)
4	RPM	4F12.6,I5	propeller revolution rate (rev./min.)
	USI		ship speed V_S(ft./sec.) or (m./sec.) as given by IUNIT
	FREQ		vibration frequency (rad./sec.)
	RHOI		density of water (slugs/ft.^{3}) or (kg/m.^{3}) as given by IUNIT
	UNIT		IUNIT=0; USI and RHOI in English units
			=1; USI and RHOI in metric units

Table A-1. Data Set Definition (continued)

Record Type	Input	Format	Comments
5	IRAD	5I5	IRAD=1; if dimensional radii input (in.) =2; if dimensional radii input (m.) =3; if nondimensional radii input $x=R/R_t$ with $R(M)=R_t$ dimensional input (in.) =4; if nondimensional radii input $x=R/R_t$ with $R(M)=R_t$ dimensional input (m.)
	ICHORD		ICHORD=1; if chord input as projected semichord (rad.) =2; if chord input as nondimensional C/D ratio
	ISKEW		ISKEW=1; if skew input in radians =2; if skew input in degrees
	IPITCH		IPITCH=1, if dimensional pitch input (in.) =2; if dimensional pitch input (m.) =3; if pitch input as nondimensional P/D ratio
	IRAKE		IRAKE=1, if dimensional input (in./ft.) =2; if dimensional input (mm./m.) =3; if dimensional input (deg.)
6	RADIUS	6F12.6 repeated M times for each radius R	radius R input as indicated by IRAD
	CHORD		chord at radius R input as indicated by ICHORD
	WAKE		mean longitudinal inflow velocity at radius R nondimensionalized by ship speed, V_x/V_s
	SKEW		propeller projected skew at radius R input as indicated by ISKEW; positive opposite direction of rotation
	PCH		propeller pitch at radius R input as indicated by IPITCH
	RAKE		propeller rake at radius R as indicated by IRAKE; positive aft.

Table A-1. Data Set Definition (concluded)

The time limit XXX depends on the number of propeller blades (NB=Z) and the theory used to obtain the blade pressure distributions. Suggested limits are shown in Table A-2 when the results are desired for all cases. Approximately 20% of the time shown for the lifting-line theory is needed if only the axial and/or torsional results are desired; approximately 80% is needed if only the lateral results are desired.

number of blades	two-dim. foil theory	lifting-line theory
4	5	100
5	5	120
6	5	140
7	5	160

Table A-2. Suggested Time Limits for PRAMAD

SAMPLE OUTPUT: The following example is for a five-bladed propeller which is divided into MM=8 segments radially. The lifting-line theory is used to evaluate the added mass and damping for torsional, axial, and lateral vibrations. English units are used in the input. The propeller operates in a uniform wake and has no rake or skew. Chord and pitch are input as non-dimensional ratios. The radii are input in non-dimensional form with the last radius R(17) input in inches (IRAD=3). The input data file is shown below followed by the program output for KODE=0. The first part of the output is a verification of the input. The coupled torsional and axial results are output next. This is followed by the lateral results which are given as results for forces parallel to the motion and then forces perpendicular to the motion. Each output result is given dimensionally in English and then SI units followed by a non-dimensional result. The non-dimensional result is followed on the same line by the terms by which it must be multiplied to produce a dimensional result. For example, the torsional added mass moment of inertia result is non-dimensionalized by ρD^5. The results for the torsional/axial case and the lateral results for the force parallel to the motion are followed by lifting-surface corrections. The lifting-line results can be multiplied by this value as an approximate correction for the chordwise effects which are neglected in the lifting-line thoery. No correction is given for the lifting-line results in the lateral

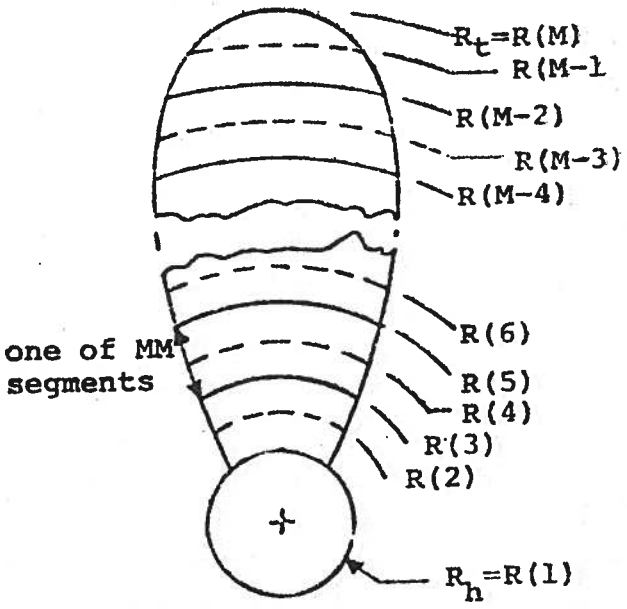


Figure A-1. Blade Segmentation for Data Input

Line 5 of the data file controls the variety of forms which can be used to input the propeller geometry. The radii can be input in inches, meters, or non-dimensional form. If the non-dimensional form is used the tip radius R(M) must be given in dimensional form to scale the input. Chord can be input as projected semichord in radians or as chord-diameter ratio. Skew can be input as degrees or radians. Pitch can be input in inches, meters, or non-dimensional pitch-diameter ratio. Rake can be input in inches/ft., mm/m, or degrees. Data record type 6 is repeated as M lines in order to input the data for each of the M radii.

MTS Run Information: PRAMAD may be run from a terminal using a preloaded data file or as a batch program using data cards. The object code is located in K8R6:PRAMAD.O (subject to change; see Professor Parsons for current location). To run the program from the terminal enter:

```
$RUN  K8R6:PRAMAD.O  4=datafile  T=XXX
```

To run the program as a batch program use:

```
$RUN  K8R6:PRAMAD.O  4=*SOURCE*  T=XXX
  (data cards)
$ENDFILE
```

case with the force perpendicular to the motion or for any case when the two-dimensional theory is used.

```
DTNSRDC PROPELLER 4381 EXAMPLE PROBLEM USING PRAMAD
    1    1    1    1    0
   17    5
       110.0        32.60       57.596       1.9905    0
    3    2    1    3    3
       0.200        0.174        1.000        0.000        1.332        0.000
       0.250        0.202        1.000        0.000        1.338        0.000
       0.300        0.229        1.000        0.000        1.345        0.000
       0.350        0.253        1.000        0.000        1.354        0.000
       0.400        0.275        1.000        0.000        1.358        0.000
       0.450        0.295        1.000        0.000        1.352        0.000
       0.500        0.312        1.000        0.000        1.336        0.000
       0.550        0.326        1.000        0.000        1.311        0.000
       0.600        0.337        1.000        0.000        1.280        0.000
       0.650        0.344        1.000        0.000        1.246        0.000
       0.700        0.347        1.000        0.000        1.210        0.000
       0.750        0.344        1.000        0.000        1.173        0.000
       0.800        0.334        1.000        0.000        1.137        0.000
       0.850        0.314        1.000        0.000        1.101        0.000
       0.900        0.280        1.000        0.000        1.066        0.000
       0.950        0.210        1.000        0.000        1.031        0.000
       120.0        0.000        1.000        0.000        0.995        0.000

#RUN  K8R6:PRAMAD.O  4=DATAFILE  T=120
#EXECUTION BEGINS


                    UNIVERSITY OF MICHIGAN
          DEPT. OF NAVAL ARCHITECTURE AND MARINE ENGINEERING
               PROPELLER ADDED MASS AND DAMPING PROGRAM

          DEVELOPED UNDER MARAD CONTRACT NO. MA-79-SAC-B0012


DTNSRDC PROPELLER 4381 EXAMPLE PROBLEM USING PRAMAD


***INPUT VERIFICATION***

ITOR=1      IAXL=1      ILAT=1      METHOD=1

M=17        NB=5

RPM=110.00      USI= 32.60      FREQ= 57.596      RHOI=1.9905

IUNIT=0  USI IN FEET PER SECOND, RHOI IN SLUGS PER CUBIC FOOT
```

IRAD=3 NONDIMENSIONAL RADII, TIP RADIUS IN INCHES

ICHORD=2 NONDIMENSIONAL CHORD INPUT AS C/D

ISKEW=1 SKEW INPUT IN RADIANS

IPITCH=3 NONDIMENSIONAL PITCH INPUT AS P/D

IRAKE=3 RAKE IN DEGREES

R/RT	RADIUS	CHORD	WAKE	SKEW	PITCH	RAKE
0.2000	0.200	0.1740	1.0000	0.0	1.332	0.0
0.2500	0.250	0.2020	1.0000	0.0	1.338	0.0
0.3000	0.300	0.2290	1.0000	0.0	1.345	0.0
0.3500	0.350	0.2530	1.0000	0.0	1.354	0.0
0.4000	0.400	0.2750	1.0000	0.0	1.358	0.0
0.4500	0.450	0.2950	1.0000	0.0	1.352	0.0
0.5000	0.500	0.3120	1.0000	0.0	1.336	0.0
0.5500	0.550	0.3260	1.0000	0.0	1.311	0.0
0.6000	0.600	0.3370	1.0000	0.0	1.280	0.0
0.6500	0.650	0.3440	1.0000	0.0	1.246	0.0
0.7000	0.700	0.3470	1.0000	0.0	1.210	0.0
0.7500	0.750	0.3440	1.0000	0.0	1.173	0.0
0.8000	0.800	0.3340	1.0000	0.0	1.137	0.0
0.8500	0.850	0.3140	1.0000	0.0	1.101	0.0
0.9000	0.900	0.2800	1.0000	0.0	1.066	0.0
0.9500	0.950	0.2100	1.0000	0.0	1.031	0.0
1.0000	120.000	0.0	1.0000	0.0	0.995	0.0

RESULTS FOR THE COUPLED TORSIONAL AND AXIAL CASE

```
TORSIONAL ADDED MASS = 0.238663E+06 LBF-IN-SEC**2
MOMENT OF INERTIA    = 0.269665E+05 N-M-S**2
     M44             = 0.312366E-02 RHO*D**5
       LS CORRECTION = 0.762703E+00

TORSIONAL/AXIAL      = -.507396E+04 LBF-SEC**2
INERTIA COUPLING     = -.225710E+05 N-S**2
     M41             = -.159381E-01 RHO*D**4
       LS CORRECTION = 0.749178E+00

AXIAL ADDED MASS     = 0.108591E+03 LBF-SEC**2/IN
                     = 0.190179E+05 N-S**2/M
     M11             = 0.818636E-01 RHO*D**3
       LS CORRECTION = 0.737245E+00
```

```
TORSIONAL DAMPING     = 0.316935E+07 LBF-IN-SEC
                      = 0.358105E+06 N-M-S
      C44             = 0.226260E-01 RHO*N*D**5
        LS CORRECTION = 0.101003E+01

TORSIONAL/AXIAL       = -.690483E+05 LBF-SEC
VELOCITY COUPLING     = -.307154E+06 N-S
      C41             = -.118304E+00 RHO*N*D**4
        LS CORRECTION = 0.977838E+00

AXIAL DAMPING         = 0.151854E+04 LBF-SEC/IN
                      = 0.265946E+06 N-S/M
      C11             = 0.624427E+00 RHO*N*D**3
        LS CORRECTION = 0.934475E+00
```

RESULTS FOR THE LATERAL CASE

*** FORCE PARALLEL TO MOTION ***

```
LATERAL ADDED MASS = 0.360624E+06 LBF-IN-SEC**2
MOMENT OF INERTIA  = 0.407469E+05 N-M-S**2
      M55          = 0.471991E-02 RHO*D**5
      LS CORRECTION = 0.779910E+00

LATERAL INERTIA    = 0.249039E+04 LBF-SEC**2
COUPLING           = 0.110783E+05 N-S**2
      M52          = 0.782269E-02 RHO*D**4
      LS CORRECTION = 0.774606E+00

LATERAL ADDED MASS = 0.273933E+02 LBF-SEC**2/IN
                   = 0.479746E+04 N-S**2/M
      M22          = 0.206510E-01 RHO*D**3
      LS CORRECTION = 0.777770E+00

LATERAL ROTATIONAL = 0.628100E+07 LBF-IN-SEC
DAMPING            = 0.709690E+06 N-M-S
      C55          = 0.448402E-01 RHO*N*D**5
      LS CORRECTION = 0.949618E+00

LATERAL VELOCITY   = 0.356071E+05 LBF-SEC
COUPLING           = 0.158394E+06 N-S
      C52          = 0.610075E-01 RHO*N*D**4
      LS CORRECTION = 0.106533E+01
```

```
LATERAL LINEAR      = 0.383302E+03 LBF-SEC/IN
DAMPING             = 0.671289E+05 N-S/M
     C22            = 0.157615E+00 RHO*N*D**3
     LS CORRECTION  = 0.100424E+01


*** FORCE PERPENDICULAR TO MOTION ***

LATERAL INERTIA     = -.140723E+05 LBF-IN-SEC**2
COUPLING            = -.159003E+04 N-M-S**2
     M65            = -.184181E-03 RHO*D**5

LATERAL INERTIA     = -.530728E+02 LBF-SEC**2
COUPLING            = -.236089E+03 N-S**2
     M62            = -.166710E-03 RHO*D**4

LATERAL INERTIA     = 0.607790E+00 LBF-SEC**2/IN
COUPLING            = 0.106444E+03 N-S**2/M
     M32            = 0.458195E-03 RHO*D**3

LATERAL VELOCITY    = -.753010E+07 LBF-IN-SEC
COUPLING            = -.850826E+06 N-M-S
     C65            = -.537575E-01 RHO*N*D**5

LATERAL VELOCITY    = -.366754E+05 LBF-SEC
COUPLING            = -.163147E+06 N-S
     C62            = -.628379E-01 RHO*N*D**4

LATERAL VELOCITY    = -.306552E+03 LBF-SEC/IN
COUPLING            = -.536875E+05 N-S/M
     C32            = -.126055E+00 RHO*N*D**3


*****************************

#EXECUTION TERMINATED
```

DIAGNOSTICS: PRAMAD includes a number of run time error returns which may appear to the user. These messages are preceeded by ***WARNING*** and are summarized as follows:

error message	explanation
CONVERGENCE NOT REACHED IN FORINT ERR=XXX	The evaluation of the integral over the trailing vortex in the wake failed to converge in 30 iterations in subroutine FORINT. A few errors ($\leq$20) of this type has been shown to produce no noticable effect on the numerical results.

CONVERGENCE NOT REACHED IN FETCH0 ERR=XXX	The evaluation of the integral over the trailing vortex in the wake failed to converge in 20 iternations in subroutine FETCH0. Errors of this type have not been experienced.
ERROR RETURN FROM BESJ IER = X	Error condition in subroutine BESJ. If IER=2, the requested argument for the Bessel Function was negative. If IER=3, the required accuracy of 0.001 was not achieved.
ERROR RETURN FROM BESY IER = X	Error condition in subroutine BESY. If IER=2, the requested argument for the Bessel Function was negative. If IER=3, the estimate exceeds 10**70.
SINGULAR COEFFICIENT MATRIX IN CIR	The coefficient matrix of the linear linear system of equations which must be solved by subroutine CIR to obtain the downwash influence coefficients is singular.
ITERATIVE REFINEMENT FAILED TO CONVERGE IN CIR ITERATIONS=XXX	The solution of the linear system of equations needed to obtain the downwash influence coefficients failed to converge in XXX interations in subroutine CIR.

APPENDIX B. PROGRAMMER'S DOCUMENTATION FOR PRAMAD

This programmer's documentation does not duplicate the User's Documentation for PRAMAD which is included in Appendix A. The reader should be familiar with that appendix before proceeding.

1. Program Organization

PRAMAD is a modular program consisting of a MAIN program and 22 subroutines. The relationship of these modules is shown in Fig. B-1. In Fig. B-1, a solid dot is used to signify parallel calls to subroutines and not a MENU or branching among alternative subroutines. Thus, subroutines INSUB, PRES3D, at least one of AMDTOR, AMDAXL, and AMDLAT, and PRINT are called in succession by the MAIN program. Subroutine INSUB reads all data, transforms all data to a common program set of units, calculates some data derived from the input, and prints the input verification. Subroutine PRES3D calculates selected chordwise integrals involving the pressure distribution due to a unit heave or a unit pitch of the blade section. This requires the use of 14 other subroutines when the lifting-line theory is used. If the two-dimensional theory is used only the subroutine FOIL2D branch is actively used. Subroutines AMDTOR, AMDAXL, and AMDLAT use the chordwise pressure integrals produced by PRES3D to evaluate the added mass and damping associated with torsional, axial, and lateral vibration, respectively. Subroutine AMDAXL evaluates the inertia and velocity coupling terms when coupled torsional/axial results are desired (ITOR=IAXL=1). These subroutines utilize subroutine MODSIM to perform the spanwise integration of the forces. Subroutine PRINT calculates the various dimensional and non-dimensional forms of the output and prints the results. If the lifting-line theory is used, PRINT calls subroutine LSCORR to obtain the lifting-surface corrections for the torsional/axial results and the lateral results with the force parallel to the motion.

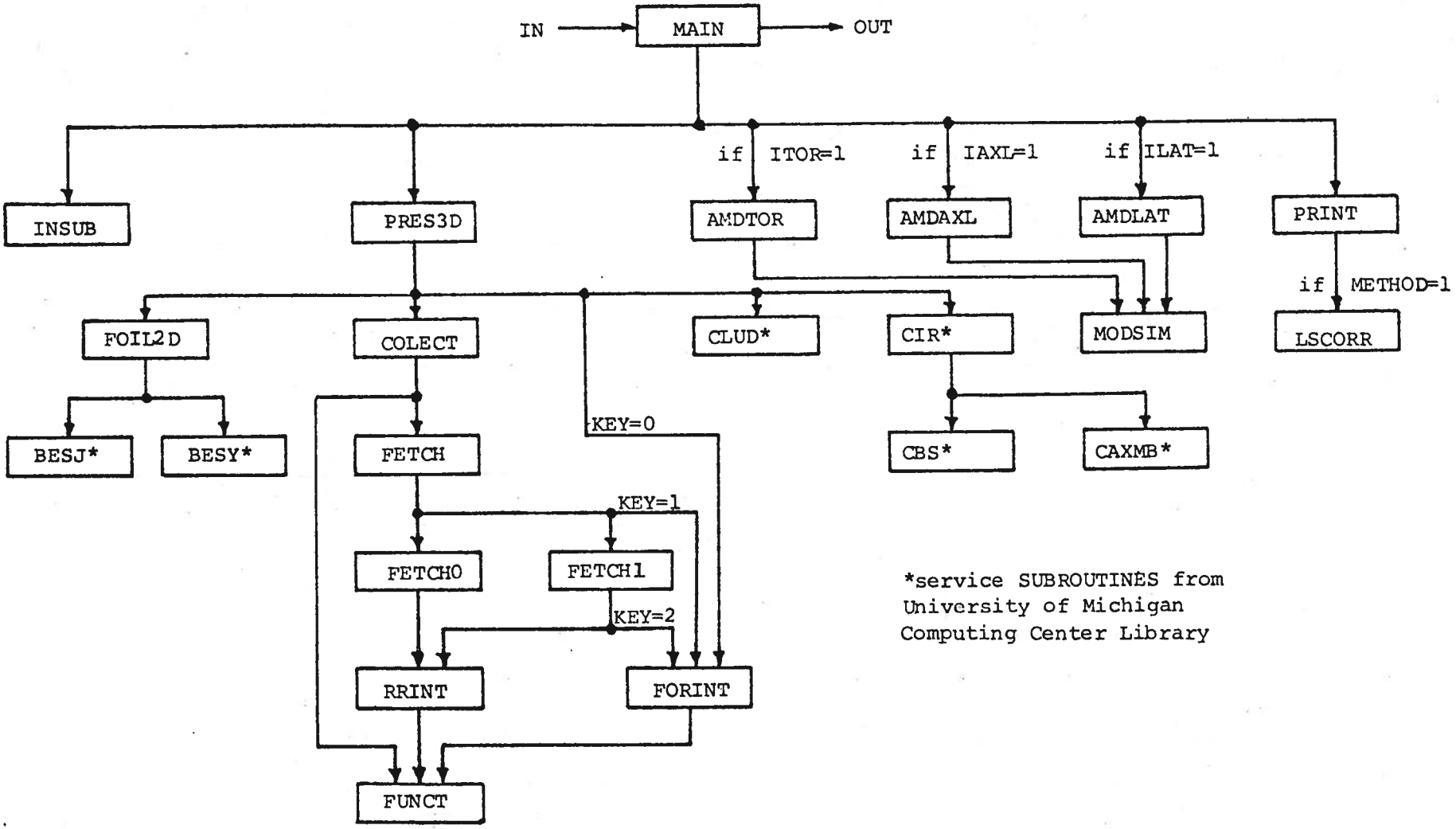


Figure B-1. PRAMAD Macro Flowchart

2. Common Block Definitions

PRAMAD utilizes an unlabeled common and 9 labeled common blocks. The variables included in these common blocks are defined in this section. Numbers in parentheses after variable names indicate the dimensions of arrays. Explicit mode declarations are noted.

2.1 Unlabeled COMMON

I	index for local segment in downwash influence coefficient development.
J	index for remote segment in downwash influence coefficients development.
II	2*I.
JJ	2*J.
RI	non-dimensional radius at center of local segment R(II)/R(M).
RK	non-dimensional radius at inner edge, outer edge, or center of remote segment.
N	vibration frequency divided by rotation frequency, FREQ/W (REAL); harmonic number when N is an integer.

2.2. COMMON A

EPS	convergence test condition for evaluation of wake integral in FORINT; set at 0.1 in FETCH1.
LIMIT	limit number of iterations for evaluation of wake integral in FORINT; set at 30 in FETCH1.
NORD	parameter of convergence acceleration scheme utilized in FORINT; set at 3 in FETCH1.

2.3. COMMON B

AA	axial limit of rectangular rule integration in trailing system; switch point between using RRINT and FORINT.
KONTRL	switch on the form of integrand in trailing vortex system; either 1 or 3.

2.4. COMMON CORR In the following all references to lateral results are for the case with the force parallel to the motion.

JXXLSC	lifting-surface correction for torsional added mass moment of inertia (REAL).
CXXLSC	lifting-surface correction for torsional/axial inertia coupling.
MXXLSC	lifting-surface correction for axial added mass (REAL).
BRXLSC	lifting-surface correction for torsional damping.
BCXLSC	lifting-surface correction for torsional/axial velocity coupling.
BXXLSC	lifting-surface correction for axial damping.
JYYLSC	lifting-surface correction for lateral added mass moment of inertia (REAL).
CYYLSC	lifting-surface correction for lateral inertia coupling.
MYYLSC	lifting-surface correction for lateral added mass (REAL).
BRYLSC	lifting-surface correction for lateral rotational damping.
BCYLSC	lifting-surface correction for lateral velocity coupling.
BYYLSC	lifting-surface correction for lateral linear damping.

2.5. COMMON D

SS1	projected angle between lifting-line segment JJ and segment II on index blade.
KON	switch with K=0 for radial trailer and K=3 for tangential trailer (set in PRES3D).
AB	projected angle between lifting-line segment II on index blade and segment JJ on any blade.
BB	axial distance between lifting-line segment II on index blade and segment JJ on any blade (n.d.).

2.6. COMMON DEVICE

INFILE	Input/output (I/O) channel for input; set in MAIN program to 4.
OUTDEV	I/O channel for output (INTEGER); set in MAIN program to 6.

2.7. COMMON INPUT

CO	(21)	cosine of the hydrodynamic pitch angle arctan(US*AO(I)/(W*R(I))) at each of M radii.
SI	(21)	sine of the hydrodynamic pitch angle at each of M radii.
BR	(21)	a local advance coefficient, $\lambda=V_a(r)/\pi nD$, at each of M radii.
TE	(21)	projected semi-chord at each of M radii(rad.).
R	(21)	radii (in.) defining edges and center of each blade segment; see Fig. A-1.
PITCH	(21)	pitch at each of M radii (in.).
A0	(21)	non-dimensional mean logitudinal inflow velocity (wake) at each of M radii (V_x/V_s).
AS	(21)	projected skew at each of M radii(rad.).
UT	(21)	hydrodynamic inflow velocity $(\sqrt{(W*R(I))**2+(US*A0(I))**2})$ at each of M radii (in/sec),
US		ship speed (in/sec).
W		propeller rotation frequency (rad/sec).
FREQ		vibration frequency (rad/sec).
RHO		water density (slugs/ft^3=lbfsec^2/ft^4).
M		number of radii for which propeller geometry and wake data is given.
MM		number of equal width segments into which blade is divided radially, M=2M+1.
NB		number of propeller blades (Z).
BETAG	(21)	geometric pitch angle at each of M radii, (rad), arctan(P(I)/2πR(I)).
AR	(21)	rake at each of M radii (in./in.).

2.8. COMMON OPTION

ITOR	control index; = 1 if torsional results desired, = 0 if not.
IAXL	control index; = 1 if axial results desired, = 0 if not.
ILAT	control index; = 1 if lateral results desired, = 0 is not.
METHOD	control index; = 1 for lifting-line theory, = 0 for two-dimensional foil theory.

2.9. COMMON OUTPUT

AMXX	(2,2)	matrix of final torsional/axial added mass results in English units; position (1,1) is torsional added mass moment of inertia, position (2,2) is axial added mass, and position (1,2) is torsional/axial inertia coupling.
DXX	(2,2)	matrix of final torsional/axial damping results in English units; position (1,1) is torsional damping, position (2,2) is axial damping, and position (1,2) is torsional/axial velocity coupling.
AMYZ	(2,4)	matrix of final lateral added mass results in English units; position (1,1) is lateral added mass moment of inertia, position (2,2) is lateral added mass, and position (1,2) is lateral inertia coupling for the case with forces parallel to motion. Positions (1,3), (2,4), and (1,4) carry the comparable results for the case with forces perpendicular to the motion.
DYZ	(2,4)	matrix of final lateral damping results in English units; position (1,1) is lateral rotational damping, position (2,2) is lateral linear damping, and position (1,2) is lateral velocity coupling for the case with forces parallel to motion. Positions (1,3), (2,4), and (1,4) carry the comparable results for the case with forces perpendicular to the motion.

2.10. COMMON PRINTG

PHX	(10)	chordwise integral involving the pressure due to unit heave at vibration frequency eq. (49) (COMPLEX) evaluated at the center of each of MM blade segments. Used in torsional/axial (X-direction) problem.
PHLP	(10)	chordwise integral involving the pressure due to unit heave at vibration plus rotation frequency eq. (55) (COMPLEX) evaluated at the center of each of MM blade segments. Used in lateral problem.
PHLM	(10)	chordwise integral involving the pressure due to unit heave at vibration minus rotation frequency eq. (55) (COMPLEX) evaluated at the center of each of MM blade segments. Used in lateral problem.
PPLP	(10)	chordwise integral involving the pressure due to unit pitch at vibration plus rotation frequency eq. (57) (COMPLEX) evaluated at the center of each of MM blade segments. Used in lateral problem.
PPLM	(10)	chordwise integral involving the pressure due to unit pitch at vibration minus rotation frequency eq. (57) (COMPLEX) evaluated at the center of each of MM blade segments. Used in lateral problem.
PHLPA	(10)	chordwise integral involving the pressure due to unit heave at vibration plus rotation frequency eq. (56) (COMPLEX) evaluated at the center of each of MM blade segments. Used in lateral problem.
PHLMA	(10)	chordwise integral involving the pressure due to unit heave at vibration minus rotation frequency eq. (56) (COMPLEX) evaluated at the center of each of MM blade segments. Used in lateral problem.
PPLPA	(10)	chordwise integral involving the pressure due to a unit pitch at vibration plus rotation frequency eq. (58) (COMPLEX) evaluated at the center of each of MM blade segments. Used in lateral problem.

PPLMA	(10)	chordwise integral involving the pressure due to a unit pitch at vibration minus rotation frequency eq. (58) (COMPLEX) evaluated at the center of each of MM blade segments. Used in lateral problem.

3. Subroutine Descriptions

The 22 subroutines of PRAMAD are described in the following sections. Those subroutines marked (*) are from the public files of the University of Michigan Computing Center. In general, these are fully documented by comments in the program listing which is included in Appendix C. Therefore, only brief functional descriptions of these subroutines will be included here. Those subroutine marked (**) are adaptations of subroutines developed previously by Professor W. S. Vorus of the Department of Naval Architecture and Marine Engineering, The University of Michigan, under American Bureau of Shipping support. Numbers in parentheses after argument names indicate the dimensions of arrays. Explicit mode declarations are noted.

3.1. SUBROUTINE AMDAXL

Purpose:	Uses the chordwise pressure integral results PHX in COMMON PRINTG to evaluate the axial added mass and damping. Called when IAXL=1. The torsional/axial inertia and velocity coupling are also evaluated if ITOR=1. Results are loaded into AMXX and DXX in COMMON OUTPUT.
Method:	The axial added mass and damping are evaluated using eq. (15), (16), and (50). The spanwise integration is performed using subroutine MODSIM. The torsional/axial inertia and velocity coupling are evaluated using eqs. (15), and (16) and F^*_{h41} in Table 3.
Calling Arguments:	None.
Common Blocks:	INPUT, OUTPUT, OPTION, PRINTG.
Comments:	Variable F11 (COMPLEX) corresponds to F^*_{h11} in eq. (50). Variable F41 (COMPLEX) corresponds to F^*_{h41} in Table 3.

3.2. SUBROUTINE AMDLAT

Purpose: Uses the chordwise pressure integral results PHLP, PHLM, PPLP, PPLM, PHLPA, PHLMA, PPLPA, and PPLMA in COMMON PRINTG to evaluate the added mass and damping for lateral vibrations. Called when ILAT=1. Results are loaded into AMYZ and DYZ in COMMON OUTPUT.

Method: The lateral added mass and damping are evaluated using eqs. (15) and (16) and the appropriate $F^*_{hj\ell}$ from Table 3. The span wise integration is performed using subroutine MODSIM.

Calling Arguments: None.

Common Blocks: INPUT, OUTPUT, PRINTG.

Comments: Variable F55 (COMPLEX) corresponds to F^*_{h55} in Table 3; variable F52 (COMPLEX) corresponds to F^*_{h52} in Table 3; variable F22 (COMPLEX) corresponds to F^*_{h22} in Table 3; variable F65 (COMPLEX) corresponds to F^*_{h65} in Table 3; variable F62 (COMPLEX) corresponds to F^*_{h62} in Table 3; variable F32 (COMPLEX) corresponds to F^*_{h32} in Table 3.

3.3. SUBROUTINE AMDTOR

Purpose: Uses the chordwise pressure integral results PHX in COMMON PRINTG to evaluate the torsional added mass moment of inertia and damping. Called when ITOR=1. Results are loaded into AMXX and DXX in COMMON OUTPUT.

Method: The torsional added mass moment of inertia and damping are evaluated using eqs. (15) and (16), and F^*_{h44} from Table 3. The spanwise integration is performed using subroutine MODSIM.

Calling Arguments: None.

Common Blocks: INPUT, OUTPUT, PRINTG.

Comments: Variable F44 (COMPLEX) corresponds to F^*_{h44} in Table 3.

3.4. SUBROUTINE BESJ*

Purpose: Computes the J Bessel Function for a given argument and order. See listing.

3.5. SUBROUTINE BESY*

Purpose: Computes the Y Bessel Function for a given argument and order. See listing.

3.6. SUBROUTINE CAXMB*

Purpose: Computes $\underline{r} = A\underline{x} - \underline{b}$, where A is an NxN matrix, in twice the precision of the data. This accuracy is crucial to the iterative refinement algorithm used in Subroutine CIR. See listing.

3.7. SUBROUTINE CBS*

Purpose: Solves the system of N linear equations $A\underline{x} = \underline{b}$ by back substitution in the LU-decomposition of matrix A. Used to obtain the starting solution vector utilized by Subroutine CIR. See listing.

3.8. SUBROUTINE CIR*

Purpose: Solves the system of N linear equations $A\underline{x} = \underline{b}$ using iterative refinement based upon LU-decomposition of matrix A. See listing.

3.9. SUBROUTINE COLECT**

Purpose: Sums the wake induced velocities for the index blade.

Method: Refer to reference 19.

Calling Arguments:

SUM	complex induced velocity at radius II due to tangential trailers from radius JJ (COMPLEX).
SUM1	complex induced velocity at radius II due to radial trailers from radius JJ (COMPLEX*16).

Common Blocks: unlabeled, B, D, INPUT

Comments: Summation variable MC introduces the influence of each of the NB blades upon the index blade.

3.10. SUBROUTINE CLUD*

Purpose: Computes the LU-decomposition of a matrix using Gaussian elimination with partial pivoting. See listing.

3.11. SUBROUTINE FETCH

Purpose: Assembles the wake induced normal velocity at radius II on the index blade due to radius JJ on blade having location A(AB) and B(BB) from COMMON D.

Method: Refer to reference 19.

Calling Arguments:

XINT — complex amplitude of induced velocity (COMPLEX).

Common Blocks: unlabeled, A, B, D, INPUT.

Comments: None.

3.12. SUBROUTINE FETCH0

Purpose: Performs improper integration over wake for non-oscillating arguments.

Method: Rectangular rule integration using Subroutine RRINT.

Calling Arguments:

XINT — value of complex integral (COMPLEX).

ALFA — argument of oscillating exponential, = 0.

Common Blocks: unlabeled, A, B, D, INPUT.

Comments: None.

3.13. SUBROUTINE FETCH1

Purpose: Performs improper integration over wake for oscillating arguments.

Method: Combination of rectangular rule over first part of cycle using Subroutine RRINT and Gauss quadrature over remainder to infinity using Subroutine FORINT.

Calling Arguments:

XINT	value of complex integral (COMPLEX).
ALFA	argument of oscillating exponential, $\neq 0$.

Common Blocks: unlabeled, A, B, D, INPUT.

Comments: None.

3.14. SUBROUTINE FOIL2D**

Purpose: To compute the two-dimensional thin foil theory results for lift coefficient, complex circulation, and chordwise pressure distribution given reduced frequency and geometry.

Method: Implements the results of Section 3.1.

Calling Arguments:

XK	reduced frequency, $k = \omega b/U$.
R	complex pressure amplitude, eq. (65) (COMPLEX)
CL	complex lift coefficient, eq. (80) divided by ω or eq. (82) divided by U (COMPLEX).
GL	complex circulation, eq. (94) (COMPLEX).
KEY	control parameter; = 1, 2, or 4 for heave problem; = 3 or 5 are for pitch problem.
ICALL	control parameter for return point; = 1 if pressure amplitude not needed; = 2 if pressure amplitude is needed.

Common Blocks: DEVICE.

Comments: When the reduced frequency XK is zero the known lift coefficient and complex circulation results are used to avoid numerical difficulties.

3.15. SUBROUTINE FORINT**

Purpose: Integrates a function of oscillating, decaying character from some point to infinity.

Method: Gauss quadrature on half-cycles with half-cycle summation. (See: "Numerical Quadrature of Fourier Transform Integrals," Mathematical Tables and Other Aids to Computation, Vol. 10, No. 55, July, 1965, pp. 140-149.)

Calling Arguments:

ALFA	argument of oscillating exponential.
XI	value of integral (COMPLEX).
KEY	index for data initialization; = 0 only lines 673-684 performed; = 1 only lines 688-690 performed; = 2 full use of subroutine, lines 693-738.

Common Blocks: A.

Comments: None.

3.16. SUBROUTINE FUNCT**

Purpose: Provide vortex kernel function for improper integration over wake.

Method: Refer to reference 19.

Calling Arguments:

D	vortex kernel function (REAL*8).
S	steamwise coordinate; argument of vortex function.

Common Blocks: unlabeled, B, D, INPUT.

Comments: None.

3.17. SUBROUTINE INSUB

Purpose: Reads all input data from I/O channel INFILE, transforms input data to a common set of program units (see COMMON INPUT description), calculates data derived from input data, and prints input verification. Some input and the calculated results are loaded into COMMON INPUT. Remaining input are returned through COMMON OPTION and argument KODE.

Method: Straight forward.

Calling Arguments:

KODE	control index; = 0 for normal output, = 1 for additional output from subroutine PRES3D.

Common Blocks: DEVICE, INPUT, OPTION.

Comments: None.

3.18. SUBROUTINE LSCORR

Purpose: Calculates the lifting-surface corrections for use with the lifting-line results for the added mass and damping for the coupled torsional/axial case and the lateral case with force parallel to the motion. The results are loaded into COMMON CORR.

Method: Corrections are evaluated using regression equations (102) through (113). The first part of the subroutine uses Simpson's Rule integration to obtain the blade area needed to evaluate the blade aspect ratio, AR. A Newton's forward difference interpolation is then used to obtain the pitch-diameter ratio at the 0.7 radius, PD07. The corrections are then evaluated as needed.

Calling Arguments: None.

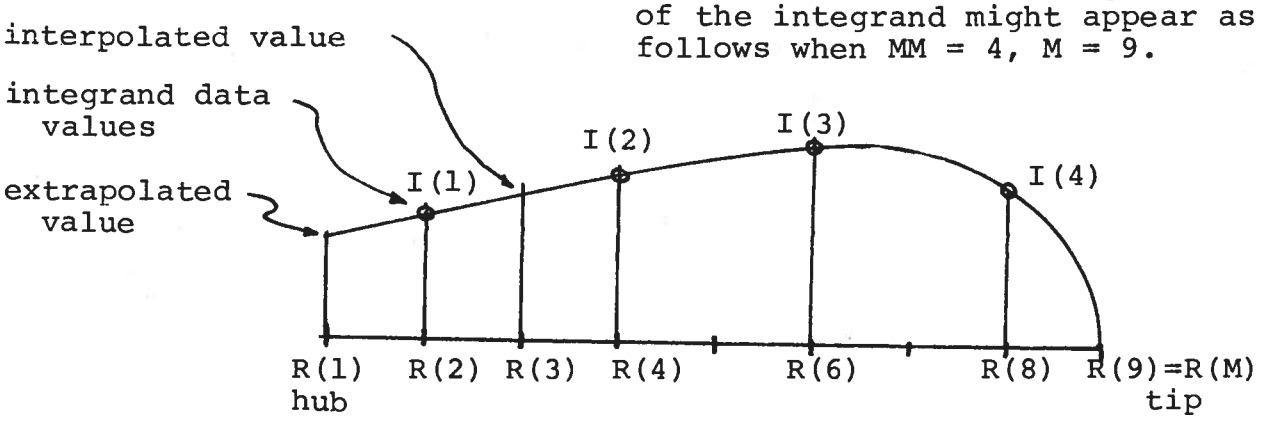
Common Blocks: CORR, INPUT, OPTION.

Comments: None.

3.19. SUBROUTINE MODSIM

Purpose: Perform spanwise integration of complex integrands which are known only at the midpoint of an even number of equal width segments (MM) of the variable of integration.

Method: Subroutine uses a modified Simpson's Rule integration. The integrand is assumed to be zero at the upper limit, the blade tip. The integrand at the lower limit, the hub, is obtained by quadratic extrapolation of the integrand values at the adjacent three points. The real or imaginary part of the integrand might appear as follows when MM = 4, M = 9.



The integral between R(2) and R(M-3=6) is obtained by the conventional Simpson's Rule. The integrands at R(1) and R(3) can be obtained algebraically in terms of the integrands at R(2), R(4), and R(6) using quadratic extrapolation, respectively; i.e.,

$$I(R(1)) = \frac{15}{8}I(1) - \frac{5}{4}I(2) + \frac{3}{8}I(3),$$

$$I(R(3)) = -\frac{3}{8}I(1) + \frac{3}{4}I(2) - \frac{1}{8}I(3)$$

The (5,8, minus 1) integration rule is then used to obtain the integral between R(1) and R(2) using the integrands at R(1), R(2), and R(3). The integral between R(M-3=6) and R(M=9) is obtained using a Simpson's Rule for unequal intervals using the integrands at R(6) and R(8), and the assumed zero integrand at R(9). (see: Van Gunsteren, L.A., "Numerical Calculation of Areas and Moments," International Shipbuilding Progress, Vol. 14, No. 157, Sept. 1967, pp. 357-364.)

The combined integration rule using only the integrand values at the even index radii then becomes as follows:

$$\text{INTEGRAL} = \frac{H}{24} \sum_{i=1}^{MM} A(i)*I(i),$$

where H=(R(M)-R(1))/MM, the segment width.

The algorithm in MODSIM generates the needed multipliers A(i) which are as follows:

MM	A(I)
4	25, 25, 19, 27
6	25, 25, 18, 32, 17, 27
8	25, 25, 18, 32, 16, 32, 17, 27
10	25, 25, 18, 32, 16, 32, 16, 32, 17, 27

Calling Arguments

INTGRD (MM)	value of integrand at the midpoint of MM spanwise segments equal width H (COMPLEX).
INTGRL	value of integral (COMPLEX).
H	width of spanwise segments.
MM	number of spanwise segments (MM≤10, even).

Common Blocks: None.

Comments: None.

3.20. SUBROUTINE PRES3D

Purpose: Develops the chordwise pressure integrals: PHX(I), PHLP(I), PHLPA(I), PPLP(I), PPLPA(I), PHLM(I), PHLMA(I), PPLM(I), and PPLMA(I) in accordance with eq. (49) and equations (55) through (58).

Method: The unsteady lifting-line theory of Brown, reference 19, is used with modification to accommodate skew and to apply to the oscillating foil problem instead of the oscillatory wake problem. See Section 3.2. Output loaded into COMMON PRINTG.

Calling Arguments:

KODE	control index; = 0 for normal output which includes nothing from PRES3D, = 1 for additional test output from PRES3D.

Common Blocks: unlabeled, D, INPUT, OPTION, PRINTG.

Comments: Variable MUM governs the basic operation. In the torsional/axial problem oscillation only at ω is needed so MUM = K = 1 and a single case is calculated. In the lateral problem oscillation at the $\omega \pm \Omega$ frequencies are needed for heave and pitch so K = 2, 3, 4, 5. Cases K = 2 and 4 are heave. Cases 3 and 5 are pitch. Cases K = 2 and 3 are for $\omega+\Omega$; cases 4 and 5 are for $\omega-\Omega$.

The subroutine also handles the two-dimensional thin foil theory problem with lines 272-325 omitted since there are no induced velocities.

3.21 SUBROUTINE PRINT

Purpose:	Calculates SI unit and non-dimensional output from the English unit output in COMMON OUTPUT. Calls subroutine LSCORR for lifting-surface corrections if lifting-line theory is used (METHOD=1).
Method:	Straight forward.
Calling Arguments:	None.
Common Blocks:	CORR, DEVICE, INPUT, OPTION, OUTPUT.
Comments:	None.

3.22. SUBROUTINE RRINT**

Purpose	Performs rectangular rule integration of a complex integrand of the type found in evaluating the downwash caused by bound and trailing vortices.
Method:	The subroutine FUNCT is used to obtain variable part of the integrand. This result, Q, is then multiplied by e^{i*XM*S}, where S is the variable of integration, in order to form the integrand. The integration is performed from S=0 to S=DELS*NC. The integrand is evaluated at the midpoint of NP intervals of the variable of integration.
Calling Arguments:	
DELS	A range of the variable of integration. Call is either 1/16 cycle, when obtaining the induced velocity on segment I due to trailers at the segment I on index blade, or 1/4 cycle.
NC	Number of DELS's in interval of integration. Call is either 1 when DELS is 1/16 cycle or 3 when DELS is 1/4 cycle.
XINT	value of integral (COMPLEX).
NP	number of intervals of variable of integration in each DELS in rectandular rule integration.
XM	integration parameter included in exponential part of integrand. Here the reduced frequency.
Common Blocks:	None.
Comments:	None.

4. PRAMAD In MTS

The PRAMAD program is written in IBM System 360/370 FORTRAN IV. It consists of 1822 source lines which are listed in Appendix C. The object version occupies 60642 bytes when compiled using the Michigan Terminal System (MTS) *FTN (level G) compiler. A size breakdown by module is as follows:

module	source lines	source statements	bytes
MAIN	17	13	480
INSUB	179	164	5556
PRES3D	185	171	9458
FOIL2D	95	91	4888
COLECT	34	32	852
FUNCT	27	25	838
FETCH	39	38	2362
FETCH0	36	34	848
FETCH1	32	31	904
RRINT	18	17	914
FORINT	78	75	6184
AMDTOR	24	23	792
AMDAXL	34	33	1222
AMDLAT	95	82	6982
MODSIM	26	21	864
PRINT	175	147	6032
LSCORR	91	79	2754
BESJ	105	53	1536
BESY	135	69	2058
CLUD	146	49	1950
CIR	115	32	1316
CBS	91	25	1038
CAXMB	45	18	814
	1822	1322	60642

APPENDIX C. LISTING OF PRAMAD

```
      C     UNIVERSITY OF MICHIGAN
      C     DEPT. OF NAVAL ARCHITECTURE AND MARINE ENGINEERING
      C     PROPELLER ADDED MASS AND DAMPING PROGRAM (PRAMAD)
      C        REVISION OF SEPTEMBER 5, 1980
            COMMON /DEVICE/ INFILE,OUTDEV
              INTEGER       OUTDEV
            COMMON /OPTION/ ITOR,IAXL,ILAT,METHOD
            OUTDEV=6
            INFILE=4
            CALL INSUB(KODE)
            CALL PRES3D(KODE)
            IF (ITOR .EQ. 1) CALL AMDTOR
            IF (IAXL .EQ. 1) CALL AMDAXL
            IF (ILAT .EQ. 1) CALL AMDLAT
            CALL PRINT
            STOP
            END


            SUBROUTINE INSUB(KODE)
      C
      C     READS DATA FROM FILE, LOADS COMMON INPUT,
      C     PRINTS INPUT VERIFICATION
      C
            COMMON /DEVICE/ INFILE,OUTDEV
              INTEGER OUTDEV
            COMMON /INPUT/ CO,SI,BR,TE,R,PITCH,AO,AS,UT,US,W,FREQ,RHO,
           1                M,MM,NB,BETAG,AR
              DIMENSION CO(21),SI(21),BR(21),TE(21),R(21),AS(21),BETAG(21)
              DIMENSION AO(21),UT(21),PITCH(21),AR(21)
            COMMON /OPTION/ ITOR,IAXL,ILAT,METHOD
            DIMENSION RADIUS(21),CHORD(21),WAKE(21),SKEW(21),PCH(21)
            DIMENSION TITLE(18),RAD(21),RAKE(21)
            DATA PI /3.14159265/
            READ(INFILE,1000) TITLE
            READ(INFILE,1010) ITOR,IAXL,ILAT,METHOD,KODE
            READ(INFILE,1020)M,NB
            MM=(M-1)/2
            MN1=M-1
            READ(INFILE,1025) RPM,USI,FREQ,RHOI,IUNIT
            IF(IUNIT.EQ.1) GO TO 10
               US=USI*12.
               RHO=RHOI
               GO TO 20
       10   CONTINUE
            US=USI*39.3701
            RHO=RHOI*.001941
       20   CONTINUE
            W=RPM*PI/30.
            READ(INFILE,1010) IRAD,ICHORD,ISKEW,IPITCH,IRAKE
```

```
      READ(INFILE,1030)(RADIUS(I),CHORD(I),WAKE(I),SKEW(I),PCH(I),
     1                  RAKE(I),I=1,M)
      GO TO (30,50,70,90),IRAD
   30 CONTINUE
      DO 40 I=1,M
         R(I)=RADIUS(I)
   40 CONTINUE
      GO TO 110
   50 CONTINUE
      DO 60 I=1,M
         R(I)=RADIUS(I)*39.3701
   60 CONTINUE
      GO TO 110
   70 CONTINUE
      R(M)=RADIUS(M)
      DO 80 I=1,MN1
         R(I)=RADIUS(I)*R(M)
   80 CONTINUE
      GO TO 110
   90 CONTINUE
      R(M)=RADIUS(M)*39.3701
      DO 100 I=1,MN1
         R(I)=RADIUS(I)*R(M)
  100 CONTINUE
  110 CONTINUE
      DO 120 I=1,M
         AO(I)=WAKE(I)
         UT(I)=SQRT((W*R(I))**2+(US*AO(I))**2)
         CO(I)=W*R(I)/UT(I)
         SI(I)=US*AO(I)/UT(I)
         BR(I)=US*AO(I)/(W*R(M))
         RAD(I)=R(I)/R(M)
         IF (ICHORD .EQ. 1) TE(I)=CHORD(I)
  120 CONTINUE
      GO TO (130,150), ISKEW
  130 CONTINUE
      DO 140 I=1,M
         AS(I)=SKEW(I)
  140 CONTINUE
      GO TO 170
  150 CONTINUE
      DO 160 I=1,M
         AS(I)=SKEW(I)*2*PI/360.
  160 CONTINUE
  170 CONTINUE
      GO TO (180,200,220),IPITCH
  180 CONTINUE
      DO 190 I=1,M
         PITCH(I)=PCH(I)
  190 CONTINUE
      GO TO 240
  200 CONTINUE
      DO 210 I=1,M
         PITCH(I)=PCH(I)*39.3701
```

```
  210  CONTINUE
       GO TO 240
  220  CONTINUE
       DO 230 I=1,M
          PITCH(I)=PCH(I)*2*R(M)
  230  CONTINUE
  240  CONTINUE
       GO TO (250,270,290),IRAKE
  250  CONTINUE
       DO 260 I=1,M
          AR(I)=RAKE(I)/12.
  260  CONTINUE
       GO TO 310
  270  CONTINUE
       DO 280 I=1,M
          AR(I)=RAKE(I)/1000.
  280  CONTINUE
       GO TO 310
  290  CONTINUE
       DO 300 I=1,M
          ARR=RAKE(I)*PI/180.
          AR(I)=TAN(ARR)
  300  CONTINUE
  310  CONTINUE
       DO 320 I=1,M
          BETAG(I)=ATAN(PITCH(I)/(2.*PI*R(I)))
          IF(ICHORD.EQ.2) TE(I)=CHORD(I)*COS(BETAG(I))/RAD(I)
  320  CONTINUE
       WRITE(OUTDEV,1050)
       WRITE(OUTDEV,1060) (TITLE(I), I=1,18)
       WRITE(OUTDEV,1065)
       WRITE(OUTDEV,1070) ITOR,IAXL,ILAT,METHOD
       WRITE(OUTDEV,1080) M,NB
       WRITE(OUTDEV,1090) RPM,USI,FREQ,RHOI
       IF(IUNIT.EQ.0)WRITE(OUTDEV,1095)IUNIT
       IF(IUNIT.EQ.1)WRITE(OUTDEV,1096)IUNIT
       IF(IRAD .EQ. 1)WRITE(OUTDEV,1100)IRAD
       IF(IRAD .EQ. 2)WRITE(OUTDEV,1101)IRAD
       IF(IRAD .EQ. 3)WRITE(OUTDEV,1102)IRAD
       IF(IRAD .EQ. 4)WRITE(OUTDEV,1103)IRAD
       IF(ICHORD .EQ. 1)WRITE(OUTDEV,1104)ICHORD
       IF(ICHORD .EQ. 2)WRITE(OUTDEV,1105)ICHORD
       IF(ISKEW .EQ. 1)WRITE(OUTDEV,1106)ISKEW
       IF(ISKEW .EQ. 2)WRITE(OUTDEV,1107)ISKEW
       IF(IPITCH .EQ. 1)WRITE(OUTDEV,1108)IPITCH
       IF(IPITCH .EQ. 2)WRITE(OUTDEV,1109)IPITCH
       IF(IPITCH .EQ. 3)WRITE(OUTDEV,1110)IPITCH
       IF(IRAKE.EQ.1)WRITE(OUTDEV,1111)IRAKE
       IF(IRAKE.EQ.2)WRITE(OUTDEV,1112)IRAKE
       IF(IRAKE.EQ.3)WRITE(OUTDEV,1113)IRAKE
       WRITE(OUTDEV,1115)
       DO 330 I=1,M
          WRITE(OUTDEV,1120) RAD(I),RADIUS(I),CHORD(I),WAKE(I),SKEW(I),
     1                       PCH(I),RAKE(I)
```

```
157       330 CONTINUE
158      1000 FORMAT(18A4)
159      1010 FORMAT(5I5)
160      1020 FORMAT(2I5)
161      1025 FORMAT(4F12.6,I5)
162      1030 FORMAT(6F12.6)
163      1050 FORMAT('-',T25,'UNIVERSITY OF MICHIGAN'/' ',T11,'DEPT. OF NAVAL AR
164          1CHITECTURE AND MARINE ENGINEERING'/' ',T16,'PROPELLER ADDED MASS A
165          2ND DAMPING PROGRAM'/'0',T11,'DEVELOPED UNDER MARAD CONTRACT NO. MA
166          3-79-SAC-B0012')
167      1060 FORMAT('-',18A4)
168      1065 FORMAT('-***INPUT VERIFICATION***')
169      1070 FORMAT('0ITOR=',I1,5X,'IAXL=',I1,5X,'ILAT=',I1,5X,'METHOD=',I1)
170      1080 FORMAT('0M=',I2,7X,'NB=',I1)
171      1090 FORMAT('0RPM=',F6.2,5X,'USI=',F6.2,5X,'FREQ=',F7.3,5X,'RHOI=',F6.4
172          1)
173      1095 FORMAT('0IUNIT=',I1,'  USI IN FEET PER SECOND, RHOI IN SLUGS PER C
174          1UBIC FOOT')
175      1096 FORMAT('0IUNIT=',I1,'  USI IN METERS PER SECOND, RHOI IN KILOGRAMS
176          1 PER CUBIC METER')
177      1100 FORMAT('0IRAD=',I1,'  RADIUS IN INCHES')
178      1101 FORMAT('0IRAD=',I1,'  RADIUS IN METERS')
179      1102 FORMAT('0IRAD=',I1,'  NONDIMENSIONAL RADII, TIP RADIUS IN INCHES')
180      1103 FORMAT('0IRAD=',I1,'  NONDIMENSIONAL RADII, TIP RADIUS IN METERS')
181      1104 FORMAT('0ICHORD=',I1,'  PROJECTED SEMICHORD IN RADIANS')
182      1105 FORMAT('0ICHORD=',I1,'  NONDIMENSIONAL CHORD INPUT AS C/D')
183      1106 FORMAT('0ISKEW=',I1,'  SKEW INPUT IN RADIANS')
184      1107 FORMAT('0ISKEW=',I1,'  SKEW INPUT IN DEGREES')
185      1108 FORMAT('0IPITCH=',I1,'  PITCH IN INCHES')
186      1109 FORMAT('0IPITCH=',I1,'  PITCH IN METERS')
187      1110 FORMAT('0IPITCH=',I1,'  NONDIMENSIONAL PITCH INPUT AS P/D')
188      1111 FORMAT('0IRAKE=',I1,'  RAKE IN IN/FT')
189      1112 FORMAT('0IRAKE=',I1,'  RAKE IN MM/M')
190      1113 FORMAT('0IRAKE=',I1,'  RAKE IN DEGREES')
191      1115 FORMAT('0 R/RT    RADIUS    CHORD     WAKE      SKEW     PITCH',
192          1        '     RAKE')
193      1120 FORMAT(' ',F6.4,2X,F7.3,3X,F6.4,3X,F6.4,2X,F7.3,2X,F7.3,
194          1       2X,F7.3)
195           RETURN
196           END


197           SUBROUTINE PRES3D(KODE)
198       C   HEAVE AND PITCH CHORDWISE PRESSURE INTEGRALS
199           COMMON /DEVICE/ INFILE,OUTDEV
200             INTEGER OUTDEV
201           COMMON /INPUT/ CO,SI,BR,TE,R,PITCH,AO,AS,UT,US,W,FREQ,RHO,
202          1                M,MM,NB ,BETAG,AR
203             DIMENSION CO(21),SI(21),BR(21),TE(21),R(21),AS(21),BETAG(21)
204             DIMENSION AO(21),UT(21),PITCH(21),AR(21)
205           COMMON /OPTION/ ITOR,IAXL,ILAT,METHOD
206           COMMON /PRINTG/ PHX,PHLP,PHLM,PPLP,PPLM,PHLPA,PHLMA,PPLPA,PPLMA
207             COMPLEX PHX,PHLP,PHLM,PPLP,PPLM,PHLPA,PHLMA,PPLPA,PPLMA
```

```
208             DIMENSION PHX(10),PHLP(10),PHLM(10),PPLP(10),PPLM(10)
209             DIMENSION PHLPA(10),PHLMA(10),PPLPA(10),PPLMA(10)
210           COMMON I,J,II,JJ,RI,RK,N
211              REAL N
212           COMMON /D/ SS1,KON,AB,BB
213           DIMENSION PRES(11),VN(10),RHS(10),AC(10),C(10,10),A(10,10)
214           DIMENSION V(10),L(10),SV(10),AC1(10),ACFAC(10),XK(10,3),X(21)
215           DIMENSION SM(11),F(10),FA(10)
216           COMPLEX PRES,VN,DUM,CL,GL,XI,C,V,AC,A,AC1,RHS,SV,F,EXPO,FA
217           COMPLEX*16 SUM1,SUM2
218           DATA PI/3.14159265/
219           DATA SM/1.,4.,2.,4.,2.,4.,2.,4.,2.,4.,1./
220         1 FORMAT('1CHORDWISE PRESSURE INTEGRALS, 2D OPTION')
221         2 FORMAT('1CHORDWISE PRESSURE INTEGRALS, LL OPTION')
222         3 FORMAT('0')
223         4 FORMAT('ONON-DIM. PRESSURES AT R = ',F8.3,
224          1       '   REDUCED FREQUENCY = ',F7.4)
225         5 FORMAT(6E15.5)
226         6 FORMAT(' ',37X,'CL = ',2E12.4/)
227         8 FORMAT ('OHARMONIC ORDER = ',F6.3)
228         9 FORMAT ('ORADIUS',10X,'VN')
229        10 FORMAT (1XF8.4,1XE14.6,1XE14.6)
230        11 FORMAT ('OREDUCED FREQ AND CL AT R = ',F8.3/)
231        12 FORMAT ('OPRESSURE INTEGRAL = ',2E14.5/)
232        13 FORMAT (1X4E15.5)
233        14 FORMAT ('ONORMAL ONSET VELOCITIES FOR K = ',I1/)
234        15 FORMAT ('ONET NORMAL VELOCITIES FOR K = ',I1/)
235        16 FORMAT ('OINFLUENCE COEFFICIENTS'/)
236     C     CALCULATE REDUCED FREQUENCIES AND COMMON TERMS
237           DO 20 I=1,M
238              X(I)=R(I)/R(M)
239       20  CONTINUE
240           DO 40 I=1,MM
241              II=2*I
242              XK(I,1)=FREQ*TE(II)*R(II)/(UT(II)*COS(BETAG(II)))
243              IF (ILAT.EQ.0) GO TO 30
244                 XK(I,2)=XK(I,1)*(FREQ+W)/FREQ
245                 XK(I,3)=XK(I,1)*(FREQ-W)/FREQ
246       30        CONTINUE
247              ACFAC(I)=TE(II)*SQRT(BR(II)**2+X(II)**2)
248       40  CONTINUE
249     C     BEGIN CALCULATIONS
250           CALL FORINT(0.0,XI,0)
251           N=FREQ/W
252           RHOU=RHO/20736.
253           IF (KODE.EQ.1.AND.METHOD.EQ.0) WRITE(OUTDEV,1)
254           IF (KODE.EQ.1.AND.METHOD.EQ.1) WRITE(OUTDEV,2)
255           IF (KODE.EQ.1) WRITE(OUTDEV,8) N
256           IF (ILAT.EQ.0) MUM=1
257           IF (ILAT.EQ.1) MUM=5
258     C     LOOP FOR DIFFERENT FREQUENCY CASES
259           DO 220 K=1,MUM
260              IF (K.EQ.1.AND.IAXL.EQ.0.AND.ITOR.EQ.0) GO TO 220
261              IF (K.LE.3) CC=1.
```

```
      IF (K.GT.3) CC=-1.
      IF (KODE.EQ.1) WRITE(OUTDEV,14) K
C     ESTABLISH NORMAL ONSET VELOCITIES
      DO 50 I=1,MM
         II=2*I
         IF (K.NE.3.AND.K.NE.5) VN(I)=(1.,0.)*FREQ
         IF (K.EQ.3.OR.K.EQ.5)  VN(I)=(1.,0.)*UT(II)
         IF (KODE.EQ.1) WRITE(OUTDEV,10) R(II),VN(I)
 50   CONTINUE
C     2D THEORY BRANCH POINT
      IF (METHOD.EQ.0) GO TO 130
      DO 60 I=1,MM
         RHS(I)=(1.,0.)
C        FIRST CALL TO FOIL2D FOR CIRCULATION
         KK=1+K/2
         CALL FOIL2D(XK(I,KK),PRES,CL,GL,K,1)
         AC(I)=ACFAC(I)*VN(I)*GL/(4.*PI)
         AC1(I)=-1./VN(I)
 60   CONTINUE
C     DOUBLE LOOP FOR INFLUENCE COEFFICIENTS
      DO 90 I=1,MM
         II=2*I
         RI=X(II)
         DO 80 J=1,MM
            KON=0
            JJ=2*J
            RK=X(JJ-1)
            CALL COLECT (XI,SUM1)
            RK=X(JJ+1)
            CALL COLECT (DUM,SUM2)
            C(I,J)=(0.,-1.)*N*(DUM-XI)+SUM2-SUM1
            KON=3
            JJ=JJ-1
            RK=X(JJ)
            CALL COLECT (XI,SUM1)
            IF (J .GT. 1) C(I,J-1)=C(I,J-1)+XI*AC(J-1)*AC1(I)
            C(I,J)=(C(I,J)-XI)*AC(J)*AC1(I)
            IF (J .NE. MM) GO TO 70
            JJ=JJ+2
            RK=X(JJ)
            CALL COLECT(XI,SUM1)
            C(I,J)=C(I,J)+XI*AC(J)*AC1(I)
 70         CONTINUE
            IF (I .EQ. J) C(I,J)=1.+C(I,J)
 80      CONTINUE
 90   CONTINUE
      IF (KODE.EQ.0) GO TO 110
         WRITE(OUTDEV,16)
         DO 100 I=1,MM
            WRITE(OUTDEV,13) (C(I,J),J=1,MM)
            WRITE(OUTDEV,3)
100      CONTINUE
110   CONTINUE
      CALL CLUD (MM,10,C,10,A,L)
```

```
316              CALL CIR (MM,10,C,10,A,L,V,RHS,SV,IER)
317              IF (IER.EQ.0) WRITE(OUTDEV,300)
318              IF (IER.LT.0) ITER=-IER
319              IF (IER.LT.0) WRITE(OUTDEV,301) ITER
320     C        COMPUTE NET NORMAL VELOCITIES
321              IF (KODE.EQ.1) WRITE(OUTDEV,15) K
322              DO 120 I=1,MM
323                 VN(I)=V(I)*VN(I)
324                 IF (KODE.EQ.1) WRITE(OUTDEV,10) R(2*I),VN(I)
325      120     CONTINUE
326     C        2D THEORY RETURN POINT
327      130     CONTINUE
328     C        CALCULATE PRESSURE INTEGRALS
329              DO 210 I=1,MM
330                 II=2*I
331                 KK=1+K/2
332                 CALL FOIL2D(XK(I,KK),PRES,CL,GL,K,2)
333                 IF (K.GT.1) GO TO 140
334                    IF (KODE.EQ.1) WRITE(OUTDEV,11) R(II)
335                    IF (KODE.EQ.1) WRITE(OUTDEV,13) XK(I,KK),CL
336                    PHX(I)=RHOU*UT(II)*VN(I)*TE(II)*CL/COS(BETAG(II))
337                    IF (KODE.EQ.1) WRITE(OUTDEV,12) PHX(I)
338                    GO TO 200
339      140        CONTINUE
340                 IF (KODE.EQ.1) WRITE(OUTDEV,4) R(II),XK(I,KK)
341                 IF (KODE.EQ.1) WRITE(OUTDEV,6) CL
342                 IF (KODE.EQ.1) WRITE(OUTDEV,5) (PRES(KL),KL=1,11)
343                 DT=TE(II)/5.
344                 ALFA=-TE(II)-DT
345                 F(I)=(0.,0.)
346                 FA(I)=(0.,0.)
347                 DO 150 KM=1,11
348                    ALFA=ALFA+DT
349                    EXPO=(0.,1.)*ALFA*CC
350                    F(I)=F(I)+SM(KM)*PRES(KM)*CEXP(EXPO)
351                    FA(I)=FA(I)+SM(KM)*PRES(KM)*ALFA*CEXP(EXPO)
352      150        CONTINUE
353                 F(I)=F(I)*0.2/3.
354                 FA(I)=FA(I)*0.2/3.
355                 F(I)=RHOU*UT(II)*VN(I)*TE(II)*F(I)/COS(BETAG(II))
356                 FA(I)=RHOU*UT(II)*VN(I)*TE(II)*FA(I)/COS(BETAG(II))
357                 IF (KODE.EQ.1) WRITE(OUTDEV,12) F(I)
358                 GO TO (200,160,170,180,190),K
359      160        CONTINUE
360                 PHLP(I)=F(I)
361                 PHLPA(I)=FA(I)
362                 GO TO 200
363      170        CONTINUE
364                 PPLP(I)=F(I)
365                 PPLPA(I)=FA(I)
366                 GO TO 200
367      180        CONTINUE
368                 PHLM(I)=F(I)
369                 PHLMA(I)=FA(I)
```

```
370                 GO TO 200
371       190       CONTINUE
372                 PPLM(I)=F(I)
373                 PPLMA(I)=FA(I)
374       200       CONTINUE
375       210     CONTINUE
376       220   CONTINUE
377             RETURN
378       300   FORMAT('0***WARNING*** SINGULAR COEFFICIENT MATRIX IN CIR'/)
379       301   FORMAT('0***WARNING*** ITERATIVE REFINEMENT FAILED TO CONVERGE IN
380           1CIR - ITERATIONS = ',I5/)
381             END


382             SUBROUTINE FOIL2D (XK,R,CL,GL,KEY,ICALL)
383       C     2D FOIL IN HEAVE/PITCH
384             COMMON /DEVICE/ INFILE,OUTDEV
385               INTEGER OUTDEV
386             DIMENSION R(11),X(11)
387             COMPLEX C,C1,P,VY,P1,R,R1,R2,CL,GL
388             DATA PI/3.14159265/
389             DATA X/-1.,-.8,-.6,-.4,-.2,0.,.2,.4,.6,.8,1./
390       C     CALCULATE LIFT COEFFICIENT
391             IF (XK.EQ.0.0) GO TO 10
392                CALL BESJ(XK,0,BJ0,.001,IER)
393                IF (IER.EQ.2.OR.IER.EQ.3) WRITE(OUTDEV,200) IER
394                CALL BESJ(XK,1,BJ1,.001,IER)
395                IF (IER.EQ.2.OR.IER.EQ.3) WRITE(OUTDEV,200) IER
396                CALL BESY(XK,0,BY0,IER)
397                IF (IER.GE.2) WRITE(OUTDEV,201) IER
398                CALL BESY(XK,1,BY1,IER)
399                IF (IER.GE.2) WRITE(OUTDEV,201) IER
400                C=BJ1-(0.,1.)*BY1
401                P=(0.,1.)*(BJ0-(0.,1.)*BY0)
402                C1=C+P
403                C=C/C1
404                GO TO 20
405        10   CONTINUE
406             C=(1.,0.)
407        20   CONTINUE
408             GO TO (30,30,40,30,40), KEY
409        30   CONTINUE
410             CL=PI*(XK-2.*(0.,1.)*C)
411             GO TO 50
412        40   CONTINUE
413             CL=-PI*(2.*C+(0.,1.)*XK*(1.+C))
414        50   CONTINUE
415       C     CALCULATE COMPLEX CIRCULATION
416             IF (XK.NE.0.0) GO TO 60
417                GL=2.*PI*(1.,0.)
418                GO TO 90
419        60   CONTINUE
420             DXI=.02
```

```
421           VY=(0.,1.)
422           N=100
423           P=(0.,0.)
424           XI=-1.+.5*DXI
425           DO 80 I=1,N
426              IF (KEY .EQ. 3 .OR. KEY .EQ. 5) VY=1.+(0.,1.)*XK*XI
427              IF (I .EQ. N) GO TO 70
428                 P1=SQRT((1.+XI)/(1.-XI))*VY
429                 P=P+P1
430        70    CONTINUE
431              XI=XI+DXI
432        80    CONTINUE
433           IF (KEY .EQ. 3 .OR. KEY .EQ. 5) VY=1.+(0.,1.)*XK
434           P=P*DXI+2.*SQRT(2.*DXI)*VY
435           GL=4.*P*CEXP((0.,-1.)*XK)/(PI*XK*C1)
436        90 CONTINUE
437           IF (KEY.EQ.1.OR.ICALL.EQ.1) RETURN
438        C  CALCULATE CHORDWISE PRESSURE
439           P=P*(1.-C)
440           VY=(0.,1.)
441           D5=SQRT(DXI)
442           R(11)=(0.,0.)
443           DO 130 J=2,10
444              D1=SQRT((1.-X(J))/(1.+X(J)))
445              D2=SQRT(1.-X(J)*X(J))
446              D4=SQRT(1.-X(J))
447              R1=(0.,0.)
448              R2=(0.,0.)
449              XI=-1.+DXI/2
450              DO 120 I=1,N
451                 IF (ABS(X(J)-XI) .LT. .001) GO TO 110
452                    D3=SQRT(1.-XI*XI)
453                    FUN=.5*ALOG((1.-X(J)*XI+D2*D3)/(1.-X(J)*XI-D2*D3))
454                    IF (KEY .EQ.3 .OR. KEY .EQ. 5) VY=1.+(0.,1.)*XK*XI
455                    IF (I .EQ. N) GO TO 100
456                       P1=SQRT((1.+XI)/(1.-XI))*VY
457                       R1=R1+P1/(X(J)-XI)
458        100            CONTINUE
459                    R2=R2+FUN*VY
460        110         CONTINUE
461                 XI=XI+DXI
462        120      CONTINUE
463              VY=(0.,1.)
464              IF (KEY .EQ. 3 .OR. KEY .EQ. 5) VY=1.+(0.,1.)*XK
465              D6=1.414*ALOG((D4-D5)/(D4+D5))/D4
466              R1=R1*DXI+D6*VY
467              R(J)=(0.,-1.)*XK*R2
468              R(J)=R(J)*DXI+D1*R1
469              R(J)=(R(J)+P*D1)*2./PI
470        130 CONTINUE
471           R(1)=+15.*CL-4.*R(2)-2.*R(3)-4.*R(4)-2.*R(5)
472           R(1)=R(1)-4.*R(6)-2.*R(7)-4.*R(8)-2.*R(9)-4.*R(10)
473           RETURN
474        200 FORMAT('0***WARNING*** ERROR RETURN FROM BESJ - IER = ',I2/)
475        201 FORMAT('0***WARNING*** ERROR RETURN FROM BESY - IER = ',I2/)
476           END
```

```
      SUBROUTINE COLECT (SUM,SUM1)
C     SUBR FOR BLADE SUMMATION
      COMMON /INPUT/ CO,SI,BR,TE,R,PITCH,AO,AS,UT,US,W,FREQ,RHO,
     1              M,MM,NB,BETAG,AR
        DIMENSION CO(21),SI(21),BR(21),TE(21),R(21),AS(21),BETAG(21)
        DIMENSION AO(21),UT(21),PITCH(21),AR(21)
      COMMON I,J,II,JJ,RI,RK,N
        REAL N
      COMMON /B/ AA,KONTRL
      COMMON /D/ S1,KON,A,B
      COMPLEX*16 SUM1
      COMPLEX XINT,SUM
      REAL*8 AM,D
      DATA PI/3.14159265/
      SUM=(0.,0.)
      SUM1=(0.,0.)
      B=AS(II)*BR(II)-AS(JJ)*BR(JJ)
      S1=AS(II)-AS(JJ)
      DO 20 MC=1,NB
         AM=2.*PI*(MC-1)/NB
         A=-S1+AM
         IF (KON .GT. 1) GO TO 10
            ALFA=SIN(A)
            IF (ALFA .EQ. 0. .AND. B .EQ. 0.) GO TO 20
            AA=0.
            CALL FUNCT (D,0.)
            SUM1=SUM1+D
            GO TO 20
   10    CONTINUE
         CALL FETCH (XINT)
         SUM=SUM+XINT
   20 CONTINUE
      RETURN
      END


      SUBROUTINE FUNCT (D,S)
C     SUBR FOR VNBT KERNEL
      COMMON /INPUT/ CO,SI,BR,TE,R,PITCH,AO,AS,UT,US,W,FREQ,RHO,
     1              M,MM,NB,BETAG,AR
        DIMENSION CO(21),SI(21),BR(21),TE(21),R(21),AS(21),BETAG(21)
        DIMENSION AO(21),UT(21),PITCH(21),AR(21)
      COMMON I,J,II,JJ,RI,RK,N
        REAL N
      COMMON /B/ AA,KONTRL
      COMMON /D/ X1,KON,A,B
      REAL*8 S2,CC,SS,D,F
      S1=S+AA
      S2=B-BR(JJ)*S1*1.D0
      CC=DCOS(A+S1*1.D0)
      SS=DSIN(A+S1*1.D0)
      D=DSQRT(S2*S2+RI*RI+RK*RK-2.*RI*RK*CC)
      IF (KON .GT. 0) GO TO 10
```

```
528               D=1./(D*((RI*SS)**2+S2*S2))
529               D=D*(RK-RI*CC)
530               D=D*(RI*CO(II)*SS-S2*SI(II)*CC)
531               RETURN
532         10    CONTINUE
533               D=1./(D**3)
534               IF (KONTRL .NE. 3) RETURN
535               D=D*S1
536               RETURN
537               END


538               SUBROUTINE FETCH (XINT)
539               COMMON /INPUT/ CO,SI,BR,TE,R,PITCH,AO,AS,UT,US,W,FREQ,RHO,
540              1               M,MM,NB,BETAG,AR
541                 DIMENSION CO(21),SI(21),BR(21),TE(21),R(21),AS(21),BETAG(21)
542                 DIMENSION AO(21),UT(21),PITCH(21),AR(21)
543               COMMON I,J,II,JJ,RI,RK,N
544                 REAL N
545               COMMON /A/ EPS,LIMIT,NORD
546               COMMON /B/ AA,KONTRL
547               COMMON /D/ S1,KON,A,B
548               COMPLEX XINT,XI,EINT,EX1,EX2
549               DIMENSION ALFA(5),EINT(5)
550               DATA PI/3.14159265/
551               ALFA(1)=N
552               ALFA(2)=N-1
553               ALFA(3)=N-1
554               ALFA(4)=N+1
555               ALFA(5)=N+1
556               DO 10 JK=1,5
557                  AB=-ALFA(JK)
558                  CALL FORINT(AB,XI,1)
559                  IF (JK .LT. 3 .OR. JK .EQ. 4) KONTRL=1
560                  IF (JK .EQ. 3 .OR. JK .EQ. 5) KONTRL =3
561                  IF (ALFA(JK).LT.0.5) CALL FETCH0 (XI,AB)
562                  IF (ALFA(JK).GE.0.5) CALL FETCH1 (XI,AB)
563                  EINT(JK)=XI
564         10    CONTINUE
565               EX1=CEXP((0.,1.)*A)
566               EX2=1./EX1
567               EC=RI*CO(II)*CO(JJ)-RK*SI(II)*SI(JJ)
568               ES=RI*SI(II)*SI(JJ)-RK*CO(II)*CO(JJ)
569               XINT=ES*EINT(1)
570               XINT=XINT+.5*(EC-(0.,1.)*SI(II)*CO(JJ)*B)*EX1*EINT(2)
571               XINT=XINT+.5*(0.,1.)*SI(II)*CO(JJ)*BR(JJ)*EX1*EINT(3)
572               XINT=XINT+.5*(EC+(0.,1.)*SI(II)*CO(JJ)*B)*EX2*EINT(4)
573               XINT=XINT-.5*(0.,1.)*SI(II)*CO(JJ)*BR(JJ)*EX2*EINT(5)
574               XINT=XINT*SQRT(RK*RK+BR(JJ)*BR(JJ))
575               RETURN
576               END
```

```
      SUBROUTINE FETCH0 (XINT,ALFA)
      COMMON /DEVICE/ INFILE,OUTDEV
        INTEGER OUTDEV
      COMMON /INPUT/ CO,SI,BR,TE,R,PITCH,AO,AS,UT,US,W,FREQ,RHO,
     1               M,MM,NB,BETAG,AR
        DIMENSION CO(21),SI(21),BR(21),TE(21),R(21),AS(21),BETAG(21)
        DIMENSION AO(21),UT(21),PITCH(21),AR(21)
      COMMON I,J,II,JJ,RI,RK,N
        REAL N
      COMMON /A/ EPS,LIMIT,NORD
      COMMON /B/ AA,KONTRL
      COMMON /D/ S1,KON,A,B
      COMPLEX XINT,XI,F1
      DATA PI/3.14159265/
      AA=0.
      DS1Q=PI/2.
      XINT=(0.,0.)
      SDEL=DS1Q
      DIFF=A+S1
      IF (I .EQ. J .AND. DIFF .EQ. 0.) SDEL=PI/8.
      MK=0
   10 CONTINUE
      CALL RRINT (SDEL,1,XI,40,ALFA)
      XINT=XINT+XI
      AA=AA+SDEL
      IF (SDEL .NE. DS1Q) SDEL=DS1Q
      ERR=CABS(XI)/CABS(XINT)
      IF (ERR.LE.0.05) GO TO 20
         MK=MK+1
         IF (MK.LE.20) GO TO 10
         WRITE (OUTDEV,1) ERR
    1    FORMAT('0***WARNING*** CONVERGENCE NOT REACHED IN FETCH0',
     1          '  ERR = ',E12.6)
   20 CONTINUE
      RETURN
      END


      SUBROUTINE FETCH1(XINT,ALFA)
      COMMON /INPUT/ CO,SI,BR,TE,R,PITCH,AO,AS,UT,US,W,FREQ,RHO,
     1               M,MM,NB,BETAG,AR
        DIMENSION CO(21),SI(21),BR(21),TE(21),R(21),AS(21),BETAG(21)
        DIMENSION AO(21),UT(21),PITCH(21),AR(21)
      COMMON I,J,II,JJ,RI,RK,N
        REAL N
      COMMON /A/ EPS,LIMIT,NORD
      COMMON /B/ AA,KONTRL
      COMMON /D/ S1,KON,A,B
      COMPLEX XINT,XI,F1,XIADD
      DATA PI/3.14159265/
      AA=0.
      DS1Q=PI/(2.*ABS(ALFA))
      DIFF=A+S1
      SDEL=DS1Q
```

```
      IF(I .EQ. J .AND. DIFF .EQ. 0.) SDEL=DS1Q/4.
      CALL RRINT (SDEL,1,XINT,40,ALFA)
      AA=AA+SDEL
      IF (SDEL .EQ. DS1Q) GO TO 10
         CALL RRINT (SDEL,3,XI,10,ALFA)
         AA=AA+3.*SDEL
         XINT=XINT+XI
   10 CONTINUE
      EPS=.1
      NORD=3
      LIMIT=30
      CALL FORINT(ALFA,XI,2)
      XIADD=XI*CEXP((0.,1.)*AA)
      XINT=XINT+XIADD
      RETURN
      END


      SUBROUTINE RRINT (DELS,NC,XINT,NP,XM)
C     RECT RULE INT OF VNBT AND VNTT KERNELS
      COMPLEX XINT
      REAL*8 Q
      DATA PI/3.14159265/
      XINT=(0.,0.)
      DS=DELS/NP
      S=.5*DS
      DO 20 I=1,NC
         DO 10 J=1,NP
            CALL FUNCT(Q,S)
            XINT=XINT+Q*CEXP((0.,1.)*XM*S)
            S=S+DS
   10    CONTINUE
   20 CONTINUE
      XINT=XINT*DS
      RETURN
      END


      SUBROUTINE FORINT(ALFA,XI,KEY)
C     FOURIER TRANSFORM TYPE INTEGRAL
      COMMON /DEVICE/ INFILE,OUTDEV
        INTEGER OUTDEV
      COMMON /A/ EPS,LIMIT,NORD
      COMPLEX XI
      REAL*8 VN,VN1
      DIMENSION BF(3,3),UJ(3,3),SN(30),BETA(30),A(30,30)
      DATA PI/3.14159265E0/
      IF (KEY.GT.0) GO TO 10
         UJ(1,1)=1.E0/6.E0
         UJ(2,1)=1.E0/10.E0
         UJ(2,2)=3.E0/10.E0
         UJ(3,1)=1.E0/14.E0
```

```
677               UJ(3,2)=3.E0/14.E0
678               UJ(3,3)=5.E0/14.E0
679               BF(1,1)=.25E0/COS(PI*UJ(1,1))
680               BF(2,1)=.180901699437S0E0/COS(PI*UJ(2,1))
681               BF(2,2)=.69098300562505E-1/COS(PI*UJ(2,2))
682               BF(3,1)=.13578349056446E0/COS(PI*UJ(3,1))
683               BF(3,2)=.87322923854022E-1/COS(PI*UJ(3,2))
684               BF(3,3)=.26893585581519E-1/COS(PI*UJ(3,3))
685               RETURN
686        10     CONTINUE
687               IF (KEY.GT.1) GO TO 20
688                  XSIGN=1.E0
689                  IF(ALFA .LT. 0.) XSIGN=-1.E0
690                  GAMA=ABS(ALFA)
691                  RETURN
692        20     CONTINUE
693               P1=PI/GAMA
694               XI=(0.E0,0.E0)
695               DO 90 KM=1,2
696                  CC=1.E0
697                  C1=-1.E0
698                  DO 70 NV=1,LIMIT
699                     NU=NV-1
700                     SN(NV)=0.E0
701                     CC=CC*C1
702                     DO 30 II=1,NORD
703                        ZETA=(UJ(NORD,II)+NU+.5E0*(KM-1))*P1
704                        CALL FUNCT (VN,ZETA)
705                        ZETA=ABS((-UJ(NORD,II)+NU+.5E0*(KM-1))*P1)
706                        CALL FUNCT (VN1,ZETA)
707                        VN=(VN+VN1)*BF(NORD,II)
708                        SN(NV)=SN(NV)+VN
709        30              CONTINUE
710                     SN(NV)=-SN(NV)*CC*P1
711                     IF (NV-4) 70,40,50
712        40           CONTINUE
713                     BETA(1)=SN(1)+SN(2)
714                     BETA(2)=BETA(1)+SN(3)
715                     ML=0
716                     A(1,1)=.5E0*(BETA(1)+BETA(2))
717        50           CONTINUE
718                     IL=NU-1
719                     ML=ML+1
720                     BETA(IL+1)=BETA(IL)+SN(IL+2)
721                     A(IL,1)=.5E0*(BETA(IL)+BETA(IL+1))
722                     JL=IL-1
723                     LJ=IL
724                     DO 60 KL=1,ML
725                        A(JL,KL+1)=.5E0*(A(JL,KL)+A(LJ,KL))
726                        JL=JL-1
727                        LJ=LJ-1
728        60           CONTINUE
729                     ERR=ABS((A(1,IL)-A(1,IL-1))/A(1,IL))
730                     IF (ERR.LE.EPS) GO TO 80
```

```
731      70      CONTINUE
732              WRITE(OUTDEV,75) ERR
733      75      FORMAT('0***WARNING*** CONVERGENCE NOT REACHED IN FORINT',
734          1           '  ERR = ',E12.6)
735      80      CONTINUE
736              XI=XI+CMPLX(1.E0*(2-KM),XSIGN*(KM-1))
737          1   *(A(1,IL)-.5E0*SN(1)*(2-KM))
738      90   CONTINUE
739           RETURN
740           END


741           SUBROUTINE AMDTOR
742           COMMON /INPUT/ CO,SI,BR,TE,R,PITCH,AO,AS,UT,US,W,FREQ,RHO,
743          1               M,MM,NB,BETAG,AR
744             DIMENSION CO(21),SI(21),BR(21),TE(21),R(21),AS(21),BETAG(21)
745             DIMENSION AO(21),UT(21),PITCH(21),AR(21)
746           COMMON /OUTPUT/ AMXX,DXX,AMYZ,DYZ
747             DIMENSION AMXX(2,2),DXX(2,2),AMYZ(2,4),DYZ(2,4)
748           COMMON /PRINTG/ PHX,PHLP,PHLM,PPLP,PPLM,PHLPA,PHLMA,PPLPA,PPLMA
749             COMPLEX PHX,PHLP,PHLM,PPLP,PPLM,PHLPA,PHLMA,PPLPA,PPLMA
750             DIMENSION PHX(10),PHLP(10),PHLM(10),PPLP(10),PPLM(10)
751             DIMENSION PHLPA(10),PHLMA(10),PPLPA(10),PPLMA(10)
752           COMPLEX F44,INTGRD,INTGRL
753           DIMENSION INTGRD(10)
754           H=(R(M)-R(1))/MM
755           DO 10 I=1,MM
756             II=2*I
757             INTGRD(I)=SIN(BETAG(II))**2*R(II)**3*PHX(I)
758        10 CONTINUE
759           CALL MODSIM(INTGRD,INTGRL,H,MM)
760           F44=NB*INTGRL
761           AMXX(1,1)=(1./(FREQ*FREQ))*REAL(F44)
762           DXX(1,1)=(-1./FREQ)*AIMAG(F44)
763           RETURN
764           END


765           SUBROUTINE AMDAXL
766           COMMON /INPUT/ CO,SI,BR,TE,R,PITCH,AO,AS,UT,US,W,FREQ,RHO,
767          1               M,MM,NB,BETAG,AR
768             DIMENSION CO(21),SI(21),BR(21),TE(21),R(21),AS(21),BETAG(21)
769             DIMENSION AO(21),UT(21),PITCH(21),AR(21)
770           COMMON /OUTPUT/ AMXX,DXX,AMYZ,DYZ
771             DIMENSION AMXX(2,2),DXX(2,2),AMYZ(2,4),DYZ(2,4)
772           COMMON /OPTION/ ITOR,IAXL,ILAT,METHOD
773           COMMON /PRINTG/ PHX,PHLP,PHLM,PPLP,PPLM,PHLPA,PHLMA,PPLPA,PPLMA
774             COMPLEX PHX,PHLP,PHLM,PPLP,PPLM,PHLPA,PHLMA,PPLPA,PPLMA
775             DIMENSION PHX(10),PHLP(10),PHLM(10),PPLP(10),PPLM(10)
776             DIMENSION PHLPA(10),PHLMA(10),PPLPA(10),PPLMA(10)
777           COMPLEX F11,F41,INTGRD,INTGRL
778           DIMENSION INTGRD(10)
779           H=(R(M)-R(1))/MM
780           DO 10 I=1,MM
781             II=2*I
```

```
782           INTGRD(I)=COS(BETAG(II))**2*R(II)*PHX(I)
783        10 CONTINUE
784           CALL MODSIM(INTGRD,INTGRL,H,MM)
785           F11=NB*INTGRL
786           AMXX(2,2)=(1./(FREQ*FREQ))*REAL(F11)
787           DXX(2,2)=(-1./FREQ)*AIMAG(F11)
788           IF(ITOR.EQ.0) RETURN
789           DO 20 I=1,MM
790             II=2*I
791             INTGRD(I)=COS(BETAG(II))*SIN(BETAG(II))*R(II)**2*PHX(I)
792        20 CONTINUE
793           CALL MODSIM(INTGRD,INTGRL,H,MM)
794           F41=-NB*INTGRL
795           AMXX(1,2)=(1./(FREQ*FREQ))*REAL(F41)
796           DXX(1,2)=(-1./FREQ)*AIMAG(F41)
797           RETURN
798           END


799           SUBROUTINE AMDLAT
800           COMMON /INPUT/ CO,SI,BR,TE,R,PITCH,AO,AS,UT,US,W,FREQ,RHO,
801          1                M,MM,NB,BETAG,AR
802             DIMENSION CO(21),SI(21),BR(21),TE(21),R(21),AS(21),BETAG(21)
803             DIMENSION AO(21),UT(21),PITCH(21),AR(21)
804           COMMON /OUTPUT/ AMXX,DXX,AMYZ,DYZ
805             DIMENSION AMXX(2,2),DXX(2,2),AMYZ(2,4),DYZ(2,4)
806           COMMON /PRINTG/ PHX,PHLP,PHLM,PPLP,PPLM,PHLPA,PHLMA,PPLPA,PPLMA
807             COMPLEX PHX,PHLP,PHLM,PPLP,PPLM,PHLPA,PHLMA,PPLPA,PPLMA
808             DIMENSION PHX(10),PHLP(10),PHLM(10),PPLP(10),PPLM(10)
809             DIMENSION PHLPA(10),PHLMA(10),PPLPA(10),PPLMA(10)
810           COMPLEX J,F55,F52,F22,F65,F62,F32,INTGRD,INTGRL
811           DIMENSION INTGRD(10),G(10),AA(10),BB(10),CC(10),DD(10)
812           DIMENSION COG(10),SIG(10),A(10),B(10),C(10),D(10),E(10),F(10)
813           H=(R(M)-R(1))/MM
814           J=CMPLX(0.0,1.0)
815           DO 5 I=1,MM
816             II=2*I
817             COG(I)=COS(BETAG(II))
818             SIG(I)=SIN(BETAG(II))
819             XO=R(II)*AS(II)*SIG(I)/COG(I)+AR(II)*R(II)
820             A(I)=R(II)**2*COG(I)
821             B(I)=XO*R(II)*SIG(I)
822             C(I)=XO*SIG(I)*COG(I)
823             D(I)=R(II)*COG(I)**2
824             E(I)=R(II)**2*SIG(I)**2/COG(I)
825             F(I)=R(II)*SIG(I)**2
826             G(I)=R(II)**2*D(I)+XO*B(I)*SIG(I)
827             AA(I)=XO*E(I)*SIG(I)
828             BB(I)=F(I)*R(II)**2
829             CC(I)=B(I)*SIG(I)**2
830             DD(I)=XO*C(I)*SIG(I)+A(I)
831         5 CONTINUE
832           DO 10 I=1,MM
833             II=2*I
834             INTGRD(I)=G(I)*PHLM(I)+(AA(I)-J*BB(I))*PHLMA(I)
835          1  +(CC(I)+J*DD(I))*PPLM(I)+(E(I)+J*CC(I))*PPLMA(I)+G(I)*PHLP(I)
```

```
     2  +(AA(I)+J*BB(I))*PHLPA(I)+(CC(I)-J*DD(I))*PPLP(I)
     3  +(E(I)-J*CC(I))*PPLPA(I)
   10 CONTINUE
      CALL MODSIM(INTGRD,INTGRL,H,MM)
      F55=NB/4.*INTGRL
      AMYZ(1,1)=(1./(FREQ*FREQ))*REAL(F55)
      DYZ(1,1)=(-1./FREQ)*AIMAG(F55)
      DO 20 I=1,MM
        II=2*I
        INTGRD(I)=SIG(I)*((A(I)-J*B(I))*PHLM(I)-J*E(I)*PHLMA(I)
     1  +(C(I)+J*D(I))*PPLM(I)+F(I)*PPLMA(I)+(A(I)+J*B(I))*PHLP(I)
     2  +J*E(I)*PHLPA(I)+(C(I)-J*D(I))*PPLP(I)+F(I)*PPLPA(I))
   20 CONTINUE
      CALL MODSIM(INTGRD,INTGRL,H,MM)
      F52=NB/4.*INTGRL
      AMYZ(1,2)=(1./(FREQ*FREQ))*REAL(F52)
      DYZ(1,2)=(-1./FREQ)*AIMAG(F52)
      DO 30 I=1,MM
        II=2*I
        INTGRD(I)=R(II)*(SIG(I)**2)*(PHLM(I)+(J*COG(I)/R(II))*
     1  PPLM(I)+PHLP(I)-(J*COG(I)/R(II))*PPLP(I))
   30 CONTINUE
      CALL MODSIM(INTGRD,INTGRL,H,MM)
      F22=NB/4.*INTGRL
      AMYZ(2,2)=(1./(FREQ*FREQ))*REAL(F22)
      DYZ(2,2)=(-1./FREQ)*AIMAG(F22)
      DO 40 I=1,MM
        II=2*I
        INTGRD(I)=J*G(I)*PHLM(I)+(BB(I)+J*AA(I))*PHLMA(I)
     1  -(DD(I)-J*CC(I))*PPLM(I)-(CC(I)-J*E(I))*PPLMA(I)-J*G(I)*PHLP(I)
     2  +(BB(I)-J*AA(I))*PHLPA(I)-(DD(I)+J*CC(I))*PPLP(I)
     3  -(CC(I)+J*E(I))*PPLPA(I)
   40 CONTINUE
      CALL MODSIM(INTGRD,INTGRL,H,MM)
      F65=NB/4.*INTGRL
      AMYZ(1,3)=(1./(FREQ*FREQ))*REAL(F65)
      DYZ(1,3)=(-1./FREQ)*AIMAG(F65)
      DO 50 I=1,MM
        II=2*I
        INTGRD(I)=SIG(I)*((B(I)+J*A(I))*PHLM(I)+E(I)*PHLMA(I)
     1  -(D(I)-J*C(I))*PPLM(I)+J*F(I)*PPLMA(I)+(B(I)-J*A(I))*PHLP(I)
     2  +E(I)*PHLPA(I)-(D(I)+J*C(I))*PPLP(I)-J*F(I)*PPLPA(I))
   50 CONTINUE
      CALL MODSIM(INTGRD,INTGRL,H,MM)
      F62=NB/4.*INTGRL
      AMYZ(1,4)=(1./(FREQ*FREQ))*REAL(F62)
      DYZ(1,4)=(-1./FREQ)*AIMAG(F62)
      DO 60 I=1,MM
        II=2*I
        INTGRD(I)=R(II)*(SIG(I)**2)*(J*PHLM(I)-(COG(I)/R(II))*
     1  PPLM(I)-J*PHLP(I)-(COG(I)/R(II))*PPLP(I))
   60 CONTINUE
      CALL MODSIM(INTGRD,INTGRL,H,MM)
      F32=NB/4.*INTGRL
      AMYZ(2,4)=(1./(FREQ*FREQ))*REAL(F32)
      DYZ(2,4)=(-1./FREQ)*AIMAG(F32)
      RETURN
      END
```

```
      SUBROUTINE MODSIM(INTGRD,INTGRL,H,MM)
C     INTEGRATION BY A MODIFIED SIMPSON'S RULE
C     COMPLEX INTEGRAND GIVEN AT THE MIDPOINTS OF MM
C     SEGMENTS OF WIDTH H.  VALUE AT LOWER LIMIT (HUB)
C     OBTAINED BY QUADRATIC EXTRAPOLATION.  VALUE AT
C     UPPER LIMIT (TIP) ASSUMED ZERO.
      COMPLEX INTGRD,INTGRL
      DIMENSION INTGRD(MM),A(10)
      DATA A/25.,25.,8*0.0/
      A(MM)=27.
      A(MM-1)=17.
      IF(MM.EQ.4) GO TO 20
         NN=(MM-4)/2
         DO 10 N=1,NN
            A(2*N+1)=16.
            A(2*N+2)=32.
   10    CONTINUE
   20 CONTINUE
      A(3)=A(3)+2.
      INTGRL=CMPLX(0.0,0.0)
      DO 30 I=1,MM
         INTGRL=INTGRL+A(I)*INTGRD(I)
   30 CONTINUE
      INTGRL=INTGRL*H/24.
      RETURN
      END


      SUBROUTINE PRINT
      COMMON /DEVICE/ INFILE,OUTDEV
        INTEGER OUTDEV
      COMMON /INPUT/ CO,SI,BR,TE,R,PITCH,AO,AS,UT,US,W,FREQ,RHO,
     1                M,MM,NB,BETAG,AR
        DIMENSION CO(21),SI(21),BR(21),TE(21),R(21),AS(21),BETAG(21)
        DIMENSION AO(21),UT(21),PITCH(21),AR(21)
      COMMON /OUTPUT/ AMXX,DXX,AMYZ,DYZ
        DIMENSION AMXX(2,2),DXX(2,2),AMYZ(2,4),DYZ(2,4)
      COMMON /OPTION/ ITOR,IAXL,ILAT,METHOD
      COMMON /CORR/ JXXLSC,CXXLSC,MXXLSC,BRXLSC,BCXLSC,BXXLSC,
     1              JYYLSC,CYYLSC,MYYLSC,BRYLSC,BCYLSC,BYYLSC
        REAL JXXLSC,MXXLSC,JYYLSC,MYYLSC
      REAL JXX,MXX,JYY,MYY,JZY,MZY
      REAL JXXND,MXXND,JYYND,MYYND,JZYND,MZYND
      DATA PI/3.14159265/
      RHOSI=RHO*515.198
      DSI=R(M)*2.*.0254
      RPS=W/(2.*PI)
      IF (METHOD.EQ.1) CALL LSCORR
      IF(ITOR.EQ.0.AND.IAXL.EQ.0) GO TO 30
         IF (ITOR .EQ. 1 .AND. IAXL .EQ. 0) GO TO 20
            IF (ITOR .EQ. 1 .AND. IAXL .EQ. 1) GO TO 10
               WRITE(OUTDEV,1010)
               MXX=AMXX(2,2)*175.133
               BXX=DXX(2,2)*175.133
               MXXND=MXX/(RHOSI*DSI**3)
```

```
1003            BRZ=DYZ(1,3)*0.11299
1004            BCZ=DYZ(1,4)*4.44840
1005            BZY=DYZ(2,4)*175.133
1006            JYYND=JYY/(RHOSI*DSI**5)
1007            CYYND=CYY/(RHOSI*DSI**4)
1008            MYYND=MYY/(RHOSI*DSI**3)
1009            JZYND=JZY/(RHOSI*DSI**5)
1010            CZYND=CZY/(RHOSI*DSI**4)
1011            MZYND=MZY/(RHOSI*DSI**3)
1012            BRYND=BRY/(RHOSI*RPS*DSI**5)
1013            BCYND=BCY/(RHOSI*RPS*DSI**4)
1014            BYYND=BYY/(RHOSI*RPS*DSI**3)
1015            BRZND=BRZ/(RHOSI*RPS*DSI**5)
1016            BCZND=BCZ/(RHOSI*RPS*DSI**4)
1017            BZYND=BZY/(RHOSI*RPS*DSI**3)
1018           WRITE(OUTDEV,1070)
1019           WRITE(OUTDEV,1071)AMYZ(1,1),JYY,JYYND
1020           IF (METHOD.EQ.1) WRITE(6,1065) JYYLSC
1021           WRITE(OUTDEV,1072)AMYZ(1,2),CYY,CYYND
1022           IF (METHOD.EQ.1) WRITE(6,1065) CYYLSC
1023           WRITE(OUTDEV,1073)AMYZ(2,2),MYY,MYYND
1024           IF (METHOD.EQ.1) WRITE(6,1065) MYYLSC
1025           WRITE(OUTDEV,1074)DYZ(1,1),BRY,BRYND
1026           IF (METHOD.EQ.1) WRITE(6,1065) BRYLSC
1027           WRITE(OUTDEV,1075)DYZ(1,2),BCY,BCYND
1028           IF (METHOD.EQ.1) WRITE(6,1065) BCYLSC
1029           WRITE(OUTDEV,1076)DYZ(2,2),BYY,BYYND
1030           IF (METHOD.EQ.1) WRITE(6,1065) BYYLSC
1031           WRITE(OUTDEV,1080)
1032           WRITE(OUTDEV,1081)AMYZ(1,3),JZY,JZYND
1033           WRITE(OUTDEV,1082)AMYZ(1,4),CZY,CZYND
1034           WRITE(OUTDEV,1083)AMYZ(2,4),MZY,MZYND
1035           WRITE(OUTDEV,1084)DYZ(1,3),BRZ,BRZND
1036           WRITE(OUTDEV,1085)DYZ(1,4),BCZ,BCZND
1037           WRITE(OUTDEV,1086)DYZ(2,4),BZY,BZYND
1038      40   CONTINUE
1039           WRITE(OUTDEV,1100)
1040      1000 FORMAT('2***RESULTS FOR THE TORSIONAL CASE***',/)
1041      1010 FORMAT('2***RESULTS FOR THE AXIAL CASE***',/)
1042      1020 FORMAT('2***RESULTS FOR THE LATERAL CASE***',//)
1043      1030 FORMAT('2***RESULTS FOR THE COUPLED TORSIONAL AND AXIAL CASE***'/)
1044      1050 FORMAT('0TORSIONAL ADDED MASS = ',E12.6,' LBF-IN-SEC**2'/' MOMENT
1045          1OF INERTIA    = ',E12.6,' N-M-S**2'/' ',5X,'M44',13X,'= ',E12.6,
1046          2' RHO*D**5')
1047      1051 FORMAT('0TORSIONAL/AXIAL',6X,'= ',E12.6,' LBF-SEC**2'/' INERTIA CO
1048          1UPLING     = ',E12.6,' N-S**2'/' ',5X,'M41',13X,'= ',E12.6,
1049          2' RHO*D**4')
1050      1052 FORMAT('0AXIAL ADDED MASS     = ',E12.6,' LBF-SEC**2/IN'/' ',21X,'
1051          1= ',E12.6,' N-S**2/M'/' ',5X,'M11',13X,'= ',E12.6,' RHO*D**3')
1052      1053 FORMAT('0TORSIONAL DAMPING    = ',E12.6,' LBF-IN-SEC'/' ',21X,'= '
1053          1,E12.6,' N-M-S'/' ',5X,'C44',13X,'= ',E12.6,' RHO*N*D**5')
1054      1054 FORMAT('0TORSIONAL/AXIAL      = ',E12.6,' LBF-SEC'/' VELOCITY COUP
1055          1LING    = ',E12.6,' N-S'/' ',5X,'C41',13X,'= ',E12.6,
1056          2' RHO*N*D**4')
1057      1055 FORMAT('0AXIAL DAMPING',8X,'= ',E12.6,' LBF-SEC/IN'/' ',21X,'= ',E
```

```
                BXXND=BXX/(RHOSI*NPS*DSI**3)
               WRITE(OUTDEV,1052)AMXX(2,2),MXX,MXXND
                IF (METHOD.EQ.1) WRITE(6,1060) MXXLSC
               WRITE(OUTDEV,1055)DXX(2,2),BXX,BXXND
                IF (METHOD.EQ.1) WRITE(6,1060) BXXLSC
                GO TO 30
   10         CONTINUE
              WRITE(OUTDEV,1030)
              JXX=AMXX(1,1)*0.11299
              CXX=AMXX(1,2)*4.44840
              MXX=AMXX(2,2)*175.133
              BRX=DXX(1,1)*0.11299
              BCX=DXX(1,2)*4.44840
              BXX=DXX(2,2)*175.133
              JXXND=JXX/(RHOSI*DSI**5)
              CXXND=CXX/(RHOSI*DSI**4)
              MXXND=MXX/(RHOSI*DSI**3)
              BRXND=BRX/(RHOSI*RPS*DSI**5)
              BCXND=BCX/(RHOSI*RPS*DSI**4)
              BXXND=BXX/(RHOSI*RPS*DSI**3)
              WRITE(OUTDEV,1050)AMXX(1,1),JXX,JXXND
              IF (METHOD.EQ.1) WRITE(6,1060) JXXLSC
              WRITE(OUTDEV,1051)AMXX(1,2),CXX,CXXND
              IF (METHOD.EQ.1) WRITE(6,1060) CXXLSC
              WRITE(OUTDEV,1052)AMXX(2,2),MXX,MXXND
              IF (METHOD.EQ.1) WRITE(6,1060) MXXLSC
              WRITE(OUTDEV,1053)DXX(1,1),BRX,BRXND
              IF (METHOD.EQ.1) WRITE(6,1060) BRXLSC
              WRITE(OUTDEV,1054)DXX(1,2),BCX,BCXND
              IF (METHOD.EQ.1) WRITE(6,1060) BCXLSC
              WRITE(OUTDEV,1055)DXX(2,2),BXX,BXXND
              IF (METHOD.EQ.1) WRITE(6,1060) BXXLSC
    1         FORMAT(6E12.6)
              GO TO 30
   20       CONTINUE
            WRITE(OUTDEV,1000)
            JXX=AMXX(1,1)*0.11299
            BRX=DXX(1,1)*0.11299
            JXXND=JXX/(RHOSI*DSI**5)
            BRXND=BRX/(RHOSI*N*DSI**5)
            WRITE(OUTDEV,1050)AMXX(1,1),JXX,JXXND
            IF (METHOD.EQ.1) WRITE(6,1060) JXXLSC
            WRITE(OUTDEV,1053)DXX(1,1),BRX,BRXND
            IF (METHOD.EQ.1) WRITE(6,1060) BRXLSC
   30    CONTINUE
         IF (ILAT.EQ.0) GO TO 40
            WRITE(OUTDEV,1020)
            JYY=AMYZ(1,1)*0.11299
            CYY=AMYZ(1,2)*4.44840
            MYY=AMYZ(2,2)*175.133
            JZY=AMYZ(1,3)*0.11299
            CZY=AMYZ(1,4)*4.44840
            MZY=AMYZ(2,4)*175.133
            BRY=DYZ(1,1)*0.11299
            BCY=DYZ(1,2)*4.44840
            BYY=DYZ(2,2)*175.133
```

```
1058          112.6,' N-S/M'/' ',5X,'C11',13X,'= ',E12.6,' RHO*N*D**3')
1059      1060 FORMAT(8X,'LS CORRECTION = ',E12.6/)
1060      1065 FORMAT(6X,'LS CORRECTION = ',E12.6/)
1061      1070 FORMAT('-*** FORCE PARALLEL TO MOTION ***',/)
1062      1071 FORMAT('OLATERAL ADDED MASS = ',E12.6,' LBF-IN-SEC**2'/' MOMENT OF
1063          1 INERTIA  = ',E12.6,' N-M-S**2'/' ',5X,'M55',11X,'= ',E12.6,
1064          2' RHO*D**5')
1065      1072 FORMAT('OLATERAL INERTIA    = ',E12.6,' LBF-SEC**2'/' COUPLING',11
1066          1X,'= ',E12.6,' N-S**2'/' ',5X,'M52',11X,'= ',E12.6,' RHO*D**4')
1067      1073 FORMAT('OLATERAL ADDED MASS = ',E12.6,' LBF-SEC**2/IN'/' ',19X,'=
1068          1',E12.6,' N-S**2/M'/' ',5X,'M22',11X,'= ',E12.6,' RHO*D**3')
1069      1074 FORMAT('OLATERAL ROTATIONAL = ',E12.6,' LBF-IN-SEC'/' DAMPING',12X
1070          1,'= ',E12.6,' N-M-S'/' ',5X,'C55',11X,'= ',E12.6,' RHO*N*D**5')
1071      1075 FORMAT('OLATERAL VELOCITY   = ',E12.6,' LBF-SEC'/' COUPLING',11X,'
1072          1= ',E12.6,' N-S'/' ',5X,'C52',11X,'= ',E12.6,' RHO*N*D**4')
1073      1076 FORMAT('OLATERAL LINEAR     = ',E12.6,' LBF-SEC/IN'/' DAMPING',12X
1074          1,'= ',E12.6,' N-S/M'/' ',5X,'C22',11X,'= ',E12.6,' RHO*N*D**3')
1075      1080 FORMAT('-*** FORCE PERPENDICULAR TO MOTION ***',/)
1076      1081 FORMAT('OLATERAL INERTIA    = ',E12.6,' LBF-IN-SEC**2'/' COUPLING'
1077          1,11X,'= ',E12.6,' N-M-S**2'/' ',5X,'M65',11X,'= ',E12.6,
1078          2' RHO*D**5'/)
1079      1082 FORMAT('OLATERAL INERTIA    = ',E12.6,' LBF-SEC**2'/' COUPLING',11
1080          1X,'= ',E12.6,' N-S**2'/' ',5X,'M62',11X,'= ',E12.6,' RHO*D**4'/)
1081      1083 FORMAT('OLATERAL INERTIA    = ',E12.6,' LBF-SEC**2/IN'/' COUPLING'
1082          1,11X,'= ',E12.6,' N-S**2/M'/' ',5X,'M32',11X,'= ',E12.6,
1083          2' RHO*D**3')
1084      1084 FORMAT('OLATERAL VELOCITY   = ',E12.6,' LBF-IN-SEC'/' COUPLING',
1085          111X,'= ',E12.6,' N-M-S'/' ',5X,'C65',11X,'= ',E12.6,
1086          2' RHO*N*D**5'/)
1087      1085 FORMAT('OLATERAL VELOCITY   = ',E12.6,' LBF-SEC'/' COUPLING',11X,'
1088          1= ',E12.6,' N-S'/' ',5X,'C62',11X,'= ',E12.6,' RHO*N*D**4'/)
1089      1086 FORMAT('OLATERAL VELOCITY   = ',E12.6,' LBF-SEC/IN'/' COUPLING',
1090          111X,'= ',E12.6,' N-S/M'/' ',5X,'C32',11X,'= ',E12.6,
1091          2' RHO*N*D**3'/)
1092      1100 FORMAT('-*****************************'////)
1093           RETURN
1094           END


1095           SUBROUTINE LSCORR
1096      C    CALCULATES LIFTING-SURFACE CORRECTIONS AS NEEDED
1097           COMMON /INPUT/ CO,SI,BR,TE,R,PITCH,AO,AS,UT,US,W,FREQ,RHO,
1098          1               M,MM,NB,BETAG,ARR
1099             DIMENSION CO(21),SI(21),BR(21),TE(21),R(21),AS(21),BETAG(21)
1100             DIMENSION AO(21),UT(21),PITCH(21),ARR(21)
1101           COMMON /OPTION/ ITOR,IAXL,ILAT,METHOD
1102           COMMON /CORR/ JXXLSC,CXXLSC,MXXLSC,BRXLSC,BCXLSC,BXXLSC,
1103          1          JYYLSC,CYYLSC,MYYLSC,BRYLSC,BCYLSC,BYYLSC
1104             REAL JXXLSC,MXXLSC,JYYLSC,MYYLSC
1105           DIMENSION SM(21)
1106           DATA PI/3.14159265/
1107      C    CALCULATE ASPECT RATIO OF BLADES
1108           MMM=M-1
```

```
1109            SM(1)=1.
1110            SM(M)=1.
1111            H=(R(M)-R(1))/MMM
1112            DO 5 K=2,MMM
1113               KK=(K/2)*2
1114               IF (KK.EQ.K) SM(K)=4.
1115               IF (KK.NE.K) SM(K)=2.
1116       5    CONTINUE
1117            AREA=0.0
1118            DO 10 I=1,M
1119               AREA=AREA+SM(I)*2.*TE(I)*R(I)/COS(BETAG(I))
1120      10    CONTINUE
1121            AREA=AREA*H/3.
1122            AR=(R(M)-R(1))**2/AREA
1123      C     INTERPOLATE FOR P/D AT X=0.7
1124            DO 20 I=1,M
1125               IF (R(I)/R(M).GT.0.7) GO TO 30
1126      20    CONTINUE
1127      30    CONTINUE
1128            U=(0.7*R(M)-R(I-1))/(R(I)-R(I-1))
1129            DEL0=PITCH(I)-PITCH(I-1)
1130            DEL1=PITCH(I+1)-PITCH(I)
1131            DEL2=(DEL1-DEL0)/2.
1132            PD07=(PITCH(I-1)+U*DEL0+U*(U-1)*DEL2)/(2.*R(M))
1133            A=AR
1134            B=PD07
1135            C=AR**2
1136            D=PD07*AR
1137            E=PD07*C
1138            F=1./AR
1139            G=F**2
1140            H=F**3
1141            O=PD07*F
1142            P=PD07*G
1143            Q=PD07*H
1144            S=AR-2.
1145            T=S**2
1146            U=PD07*S
1147            V=PD07*T
1148            X=S*PD07**2
1149            Y=G**2
1150            Z=PD07*Y
1151            IF (ILAT.EQ.0) GO TO 40
1152               JYYLSC=-.13964+.89760*A+.34086*B-0.15307*C-0.36619*D+0.070192*E
1153               CYYLSC=0.0010398+.66020*A+.39850*B-0.10261*C-0.34101*D
1154           1          +0.060368*E
1155               MYYLSC=.78170+.36153*S-.19256*U+.17908*V-.16110*T-.061038*X
1156               BRYLSC=.78255+.061046*A-2.5056*H+1.6426*Y+1.8440*Z
1157               BCYLSC=1.0121+.73647*G-3.8691*H-1.5129*Q+3.0614*Y+3.0984*Z
1158               BYYLSC=0.84266+6.7849*G+.12809*O-21.030*H-3.3471*Q+15.842*Y
1159           1          +5.1905*Z
1160      40    CONTINUE
1161            IF (IAXL.EQ.0.AND.ITOR.EQ.0) GO TO 70
1162               IF (IAXL.EQ.0.OR.ITOR.EQ.0) GO TO 50
```

```
1163                JXXLSC=.61046+.34674*B+.60294*F-.56159*G-.80696*O+.45806*P
1164                CXXLSC=.65348+.28788*B+.39805*F-.42582*G-.61189*O+.33373*P
1165                MXXLSC=.61791+.23741*B+.42253*F-.43911*G-.46697*O+.25124*P
1166                BRXLSC=.82761-.41165*G+1.2196*O+6.3993*H-13.803*Q-6.9091*Y
1167          1               +15.594*Z
1168                BCXLSC=.80998-.63077*G+1.3909*O+7.5424*H-15.689*Q-8.0097*Y
1169          1               +17.665*Z
1170                BXXLSC=.82004-.67190*G+1.3913*O+7.7476*H-16.807*Q-8.2798*Y
1171          1               +19.121*Z
1172                GO TO 70
1173       50       CONTINUE
1174                IF (ITOR.EQ.1) GO TO 60
1175                   MXXLSC=.61791+.23741*B+.42253*F-.43911*G-.46697*O+.25124*P
1176                   BXXLSC=.82004-.67190*G+1.3913*O+7.7476*H-16.807*Q-8.2798*Y
1177          1               +19.121*Z
1178                   GO TO 70
1179       60       CONTINUE
1180                JXXLSC=.61046+.34674*B+.60294*F-.56159*G-.80696*O+.45806*P
1181                BRXLSC=.82761-.41165*G+1.2196*O+6.3993*H-13.803*Q-6.9091*Y
1182          1          +15.594*Z
1183       70    CONTINUE
1184             RETURN
1185             END


1186             SUBROUTINE BESJ(X,N,BJ,D,IER)
1187       C
1188       C NAASA 10.1.003 BESJ     FTN   06-24-75     THE UNIV OF MICH COMP CTR
1189       C
1190       C     ...............................................................
1191       C
1192       C        SUBROUTINE BESJ
1193       C
1194       C        PURPOSE
1195       C           COMPUTE THE J BESSEL FUNCTION FOR A GIVEN ARGUMENT AND ORDER
1196       C
1197       C        USAGE
1198       C           CALL BESJ(X,N,BJ,D,IER)
1199       C
1200       C        DESCRIPTION OF PARAMETERS
1201       C           X  -THE ARGUMENT OF THE J BESSEL FUNCTION DESIRED
1202       C           N  -THE ORDER OF THE J BESSEL FUNCTION DESIRED
1203       C           BJ -THE RESULTANT J BESSEL FUNCTION
1204       C           D  -REQUIRED ACCURACY
1205       C           IER-RESULTANT ERROR CODE WHERE
1206       C              IER=0  NO ERROR
1207       C              IER=1  N IS NEGATIVE
1208       C              IER=2  X IS NEGATIVE OR ZERO
1209       C              IER=3  REQUIRED ACCURACY NOT OBTAINED
1210       C              IER=4  RANGE OF N COMPARED TO X NOT CORRECT (SEE REMARKS)
1211       C
1212       C        REMARKS
1213       C           N MUST BE GREATER THAN OR EQUAL TO ZERO, BUT IT MUST BE
1214       C           LESS THAN
```

```
C              20+10*X-X** 2/3   FOR X LESS THAN OR EQUAL TO 15
C              90+X/2            FOR X GREATER THAN 15
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        METHOD
C           RECURRENCE RELATION TECHNIQUE DESCRIBED BY H. GOLDSTEIN AND
C           R.M. THALER,'RECURRENCE TECHNIQUES FOR THE CALCULATION OF
C           BESSEL FUNCTIONS',M.T.A.C.,V.13,PP.102-108 AND I.A. STEGUN
C           AND M. ABRAMOWITZ,'GENERATION OF BESSEL FUNCTIONS ON HIGH
C           SPEED COMPUTERS',M.T.A.C.,V.11,1957,PP.255-257
C
C     .....................................................................
C
      BJ=.0
      IF(N)10,20,20
   10 IER=1
      RETURN
   20 IF(X)30,30,31
   30 IER=2
      RETURN
   31 IF(X-15.)32,32,34
   32 NTEST=20.+10.*X-X** 2/3
      GO TO 36
   34 NTEST=90.+X/2.
   36 IF(N-NTEST)40,38,38
   38 IER=4
      RETURN
   40 IER=0
      BPREV=.0
C
C     COMPUTE STARTING VALUE OF M
C
      IF(X-5.)50,60,60
   50 MA=X+6.
      GO TO 70
   60 MA=1.4*X+60./X
   70 MB=N+IFIX(X)/4+2
      MZERO=MAXO(MA,MB)
C
C     SET UPPER LIMIT OF M
C
      MMAX=NTEST
   DO 190 M=MZERO,MMAX,3
C
C     SET F(M),F(M-1)
C
      FM1=1.0E-28
      FM=.0
      ALPHA=.0
      IF(M-(M/2)*2)120,110,120
  110 JT=-1
      GO TO 130
  120 JT=1
  130 M2=M-2
```

```
1271            DO 160 K=1,M2
1272            MK=M-K
1273            BMK=2.*FLOAT(MK)*FM1/X-FM
1274            FM=FM1
1275            FM1=BMK
1276            IF(MK-N-1)150,140,150
1277        140 BJ=BMK
1278        150 JT=-JT
1279            S=1+JT
1280        160 ALPHA=ALPHA+BMK*S
1281            BMK=2.*FM1/X-FM
1282            IF(N)180,170,180
1283        170 BJ=BMK
1284        180 ALPHA=ALPHA+BMK
1285            BJ=BJ/ALPHA
1286            IF(ABS(BJ-BPREV)-ABS(D*BJ))200,200,190
1287        190 BPREV=BJ
1288            IER=3
1289        200 RETURN
1290            END


1292      C
1293      C NAASA 10.1.004 BESY     FTN   06-24-75     THE UNIV OF MICH COMP CTR
1294      C
1295      C     ..................................................................
1296      C
1297      C        SUBROUTINE BESY
1298      C
1299      C        PURPOSE
1300      C           COMPUTE THE Y BESSEL FUNCTION FOR A GIVEN ARGUMENT AND ORDER
1301      C
1302      C        USAGE
1303      C           CALL BESY(X,N,BY,IER)
1304      C
1305      C        DESCRIPTION OF PARAMETERS
1306      C           X  -THE ARGUMENT OF THE Y BESSEL FUNCTION DESIRED
1307      C           N  -THE ORDER OF THE Y BESSEL FUNCTION DESIRED
1308      C           BY -THE RESULTANT Y BESSEL FUNCTION
1309      C           IER-RESULTANT ERROR CODE WHERE
1310      C              IER=0  NO ERROR
1311      C              IER=1  N IS NEGATIVE
1312      C              IER=2  X IS NEGATIVE OR ZERO
1313      C              IER=3  BY HAS EXCEEDED MAGNITUDE OF 10**70
1314      C
1315      C        REMARKS
1316      C           VERY SMALL VALUES OF X MAY CAUSE THE RANGE OF THE LIBRARY
1317      C           FUNCTION ALOG TO BE EXCEEDED
1318      C           X MUST BE GREATER THAN ZERO
1319      C           N MUST BE GREATER THAN OR EQUAL TO ZERO
1320      C
```

```
C         SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C            NONE
C
C         METHOD
C            RECURRENCE RELATION AND POLYNOMIAL APPROXIMATION TECHNIQUE
C            AS DESCRIBED BY A.J.M.HITCHCOCK,'POLYNOMIAL APPROXIMATIONS
C            TO BESSEL FUNCTIONS OF ORDER ZERO AND ONE AND TO RELATED
C            FUNCTIONS', M.T.A.C., V.11,1957,PP.86-88, AND G.N. WATSON,
C            'A TREATISE ON THE THEORY OF BESSEL FUNCTIONS', CAMBRIDGE
C            UNIVERSITY PRESS, 1958, P. 62
C
C     ..................................................................
C
C     CHECK FOR ERRORS IN N AND X
C
      IF(N)180,10,10
   10 IER=0
      IF(X)190,190,20
C
C     BRANCH IF X LESS THAN OR EQUAL 4
C
   20 IF(X-4.0)40,40,30
C
C       COMPUTE YO AND Y1 FOR X GREATER THAN 4
C
   30 T1=4.0/X
      T2=T1*T1
      P0=((((-.0000037043*T2+.0000173565)*T2-.0000487613)*T2
     1  +.00017343)*T2-.001753062)*T2+.3989423
      Q0=((((.0000032312*T2-.0000142078)*T2+.0000342468)*T2
     1  -.0000869791)*T2+.0004564324)*T2-.01246694
      P1=((((.0000042414*T2-.0000200920)*T2+.0000580759)*T2
     1  -.000223203)*T2+.002921826)*T2+.3989423
      Q1=((((-.0000036594*T2+.00001622)*T2-.0000398708)*T2
     1  +.0001064741)*T2-.0006390400)*T2+.03740084
      A=2.0/SQRT(X)
      B=A*T1
      C=X-.7853982
      Y0=A*P0*SIN(C)+B*Q0*COS(C)
      Y1=-A*P1*COS(C)+B*Q1*SIN(C)
      GO TO 90
C
C       COMPUTE YO AND Y1 FOR X LESS THAN OR EQUAL TO 4
C
   40 XX=X/2.
      X2=XX*XX
      T=ALOG(XX)+.5772157
      SUM=0.
      TERM=T
      Y0=T
      DO 70 L=1,15
      IF(L-1)50,60,50
   50 SUM=SUM+1./FLOAT(L-1)
```

```
1374         60 FL=L
1375            TS=T-SUM
1376            TERM=(TERM*(-X2)/FL**2)*(1.-1./(FL*TS))
1377         70 Y0=Y0+TERM
1378            TERM = XX*(T-.5)
1379            SUM=0.
1380            Y1=TERM
1381            DO 80 L=2,16
1382            SUM=SUM+1./FLOAT(L-1)
1383            FL=L
1384            FL1=FL-1.
1385            TS=T-SUM
1386            TERM=(TERM*(-X2)/(FL1*FL))*((TS-.5/FL)/(TS+.5/FL1))
1387         80 Y1=Y1+TERM
1388            PI2=.6366198
1389            Y0=PI2*Y0
1390            Y1=-PI2/X+PI2*Y1
1391      C
1392      C     CHECK IF ONLY Y0 OR Y1 IS DESIRED
1393      C
1394         90 IF(N-1)100,100,130
1395      C
1396      C     RETURN EITHER Y0 OR Y1 AS REQUIRED
1397      C
1398        100 IF(N)110,120,110
1399        110 BY=Y1
1400            GO TO 170
1401        120 BY=Y0
1402            GO TO 170
1403      C
1404      C    PERFORM RECURRENCE OPERATIONS TO FIND YN(X)
1405      C
1406        130 YA=Y0
1407            YB=Y1
1408            K=1
1409        140 T=FLOAT(2*K)/X
1410            YC=T*YB-YA
1411            IF(ABS(YC)-1.0E70)145,145,141
1412        141 IER=3
1413            RETURN
1414        145 K=K+1
1415            IF(K-N)150,160,150
1416        150 YA=YB
1417            YB=YC
1418            GO TO 140
1419        160 BY=YC
1420        170 RETURN
1421        180 IER=1
1422            RETURN
1423        190 IER=2
1424            RETURN
1425            END
```

```
      SUBROUTINE CLUD(N,ADIM,A,TDIM,T,IV)
C
C NAASA  2.1.006 CLUD      FTN-A 10-29-75      THE UNIV OF MICH COMP CTR
C
C COMPUTES THE LU-DECOMPOSITION OF THE N X N MATRIX A USING
C GAUSSIAN ELIMINATION WITH PARTIAL PIVOTING. THIS FACTOR-
C IZATION MAY BE EXPRESSED IN THE FORM
C           L(N-1)*P(N-1)*...*L(1)*P(1)*A = U,
C WHERE EACH L(J) IS THE IDENTITY MATRIX EXCEPT FOR THE SUB-
C DIAGONAL ELEMENTS IN COLUMN J, EACH P(J) IS A PERMUTATION
C MATRIX, AND U IS AN UPPER TRIANGULAR MATRIX.  THIS IS THE
C PREPARATORY STEP IN SOLVING A SYSTEM OF LINEAR EQUATIONS,
C INVERTING A MATRIX, OR CALCULATING A DETERMINANT. A DISCUSSION
C OF GAUSSIAN ELIMINATION AND THE LU-DECOMPOSITION AND THEIR
C RELATIONSHIP TO THE NUMERICAL SOLUTION OF SYSTEMS OF LINEAR
C EQUATIONS MAY BE FOUND IN EITHER WILKINSON (1965,CHAPTER 4)
C OR FORSYTHE AND MOLER (1967).
C
      INTEGER N,ADIM,TDIM,IV(1)
      COMPLEX A(ADIM,N),T(TDIM,N)
C
C   N  -> ORDER OF THE MATRIX A.
C ADIM -> ROW DIMENSION OF THE ARRAY A. BECAUSE A IS AN N X N
C         MATRIX, ADIM SHOULD NOT BE LESS THAN N.  IF ADIM IS LESS
C         THAN N, THE CONTENTS OF A ARE IGNORED, AND THE MATRIX
C         TO BE FACTORED IS ASSUMED TO BE STORED IN THE ARRAY T.
C         SINCE ADIM MUST BE A POSITIVE INTEGER, IT IS RECOMMENDED
C         THAT THE ACTUAL ARGUMENTS A AND T COINCIDE WHEN ADIM IS
C         LESS THAN N TO AVOID THE INCONSISTENCY WHICH ARISES WHEN
C         N EQUALS 1.
C   A  -> TWO-DIMENSIONAL ARRAY CONTAINING THE N X N MATRIX TO
C         BE FACTORED, I.E., THE COEFFICIENT MATRIX OF THE SYSTEM
C         OF LINEAR EQUATIONS OR THE MATRIX TO BE INVERTED. THE
C         CONTENTS OF A ARE NOT ALTERED.
C TDIM -> ROW DIMENSION OF THE ARRAY T.
C   T  <- TWO-DIMENSIONAL ARRAY FOR RETURNING THE LU-DECOMPOSITION
C         OF A. THE SUBDIAGONAL ELEMENTS OF THE J-TH COLUMN OF THE
C         L(J) AND THE UPPER TRIANGULAR MATRIX U ARE RETURNED IN
C         THE CORRESPONDING ELEMENTS OF T.  IF ADIM IS LESS THAN N,
C         T MUST CONTAIN THE MATRIX TO BE FACTORED WHEN THIS SUB-
C         ROUTINE IS CALLED.
C  IV  <- VECTOR OF LENGTH N DEFINING THE PERMUTATION MATRICES
C         P(J): MULTIPLICATION ON THE LEFT BY P(J) INTERCHANGES
C         ROWS J AND IV(J).  IF IV(J) IS NOT EQUAL TO J, THEN
C         DET(A) = - DET(P(J)*A).  AND TO AID IN THE COMPUTATION
C         OF DET(A), IV(N) WILL CONTAIN +1 IF AN EVEN NUMBER OF
C         INTERCHANGES ARE PERFORMED AND -1 IF AN ODD NUMBER. THUS
C              DET(A) = IV(N)*T(1,1)* ... *T(N,N).
C         IV(N) WILL CONTAIN 0 IF A IS COMPUTATIONALLY SINGULAR.
C
      INTEGER I,J,K,KP1,L
      COMPLEX TMP
      REAL PIV
      REAL CABS,REAL,AIMAG
      CABS(TMP) = ABS( REAL(TMP) ) + ABS ( AIMAG(TMP) )
C
```

```
1482     C IF ADIM IS GREATER THAN OR EQUAL TO N, THE CONTENTS OF A ARE MOVED
1483     C TO T; WHILE IF ADIM IS LESS THAN N, THIS INITIAL DATA MOVEMENT IS
1484     C SKIPPED.  SINCE THE DATA MOVEMENT TIME IS PROPORTIONAL TO N**2 AND
1485     C THE COMPUTATION TIME PROPORTIONAL TO N**3/3, SIGNIFICANT SAVINGS
1486     C SHOULD NOT BE EXPECTED.
1487     C
1488           IF ( ADIM .LT. N ) GO TO 8110
1489             DO 8100 J = 1, N
1490                 DO 8100 I = 1, N
1491      8100           T(I,J) = A(I,J)
1492      8110 CONTINUE
1493     C
1494     C GAUSSIAN ELIMINATION CONSISTS OF N-1 STAGES. DURING THE K-TH
1495     C STAGE, THE PERMUTATION MATRIX P(K), THE LOWER TRIANGULAR MATRIX
1496     C L(K), AND THE K-TH ROW OF U ARE COMPUTED BASED ON THE CURRENT
1497     C ELEMENTS OF THE K-TH RESIDUAL MATRIX, I.E., THE ELEMENTS T(I,J),
1498     C I,J=K...N. ONLY THE ELEMENTS OF THE K-TH RESIDUAL MATRIX ARE
1499     C REFERENCED DURING THE K-TH STAGE OF GAUSSIAN ELIMINATION.
1500     C
1501           IV(N) = 1
1502           DO 8260 K = 1, N
1503               PIV = CABS(T(K,K))
1504               IF ( K .GE. N ) GO TO 8260
1505     C
1506     C SELECT THE PIVOT ROW FOR THE K-TH STAGE BY PARTIAL PIVOTING,
1507     C I.E., THE MAXIMUM ELEMENT IN THE 1-ST COLUMN OF THE K-TH RESIDUAL
1508     C MATRIX, AND SET IV(K) ACCORDINGLY. THE VARIABLE L HOLDS THE
1509     C SUBSCRIPT OF THE PIVOT ROW, AND PIV THE ABSOLUTE VALUE OF THE
1510     C PIVOT ELEMENT.
1511     C
1512               L = K
1513               KP1 = K + 1
1514               DO 8200 I = KP1, N
1515                   IF ( PIV .GE. CABS(T(I,K)) ) GO TO 8200
1516                     PIV = CABS(T(I,K))
1517                     L = I
1518      8200     CONTINUE
1519               IV(K) = L
1520     C
1521     C SAVE THE PIVOT ELEMENT IN TMP.  IF P(K) IS NONTRIVIAL, I.E., IV(K)
1522     C IS NOT EQUAL TO K, THE PIVOT ELEMENT IS ALWAYS NONZERO; OTHERWISE,
1523     C THE PIVOT ELEMENT MUST BE CHECKED.  IF THE PIVOT IS ZERO, I.E.,
1524     C THE MATRIX IS COMPUTATIONALLY SINGULAR, THEN T(I,K) IS ZERO FOR
1525     C I=K...N, AND THE COMPUTATION MAY PROCEED TO THE NEXT STAGE.
1526     C
1527               TMP = T(L,K)
1528               IF ( K .NE. L ) GO TO 8210
1529                 IF ( PIV .GT. 0.0 ) GO TO 8220
1530                   IV(N) = 0
1531                   GO TO 8260
1532      8210     CONTINUE
1533                 IV(N) = -IV(N)
1534                 T(L,K) = T(K,K)
1535                 T(K,K) = TMP
1536      8220     CONTINUE
1537     C
```

```
1538     C COMPUTE THE NONTRIVIAL ELEMENTS OF L(K).  BECAUSE OF THE PARTIAL
1539     C PIVOTAL STRATEGY, THE ABSOLUTE VALUE OF L(I,K)=-T(I,K)/T(K,K) IS
1540     C LESS THAN OR EQUAL TO SQRT(2). IF T(K) DENOTES THE K-TH RESIDUAL
1541     C MATRIX, THEN THE SUBDIAGONAL ELEMENTS OF THE K-TH COLUMN OF THE
1542     C MATRIX L(K) P(K) T(K) ARE ZERO. THESE ELEMENTS ARE NOT ACTUALLY
1543     C CALCULATED, THEY ARE REPLACED BY THE ELEMENTS OF L(K).
1544     C
1545               TMP = -TMP
1546               DO 8230 I = KP1, N
1547      8230         T(I,K) = T(I,K) / TMP
1548     C
1549     C APPLY P(K) AND L(K) TO THE K-TH RESIDUAL MATRIX COLUMNWISE, I.E.,
1550     C FOR J=K+1...N, INTERCHANGE T(K,J) AND T(L,J), THE (K,J)-ELEMENT
1551     C OF U, AND THEN FOR I=K+1...N, REPLACE T(I,J) BY
1552     C               T(I,J) + L(K)(I,K) * T(K,J).
1553     C
1554               DO 8250 J = KP1, N
1555                  TMP = T(L,J)
1556                  T(L,J) = T(K,J)
1557                  T(K,J) = TMP
1558                  DO 8240 I = KP1, N
1559      8240           T(I,J) = T(I,J) + T(I,K) * TMP
1560      8250     CONTINUE
1561      8260 CONTINUE
1562           IF (PIV .EQ. 0.0) IV(N) = 0
1563           RETURN
1564     C
1565     C FORSYTHE,G.E. AND MOLER,C.B. 1967. COMPUTER SOLUTION OF LINEAR
1566     C   ALGEBRAIC SYSTEMS. ENGLEWOOD CLIFFS,N.J.: PRENTICE-HALL.
1567     C WILKINSON,J.H. 1965. THE ALGEBRAIC EIGENVALUE PROBLEM. OXFORD:
1568     C   CLARENDON PRESS.
1569     C                          THE UNIVERSITY OF MICHIGAN COMPUTING CENTER
1570     C                          NUMERICAL ANALYSIS LIBRARY - JULY 1975
1571           END


1572           SUBROUTINE CIR(N,ADIM1,A,TDIM1,T,IV,X,B,R,IER)
1573     C
1574     C NAASA  2.1.007 CIR       FTN-A 10-29-75      THE UNIV OF MICH COMP CTR
1575     C
1576     C SOLVES THE SYSTEM OF LINEAR EQUATIONS AX=B, WHERE A DENOTES
1577     C THE N X N COEFFICIENT MATRIX AND X AND B ARE N-VECTORS,
1578     C USING ITERATIVE REFINEMENT BASED ON THE LU-DECOMPOSITION OF
1579     C A. THIS DECOMPOSITION MUST BE PROVIDED IN THE ARRAY T AND
1580     C VECTOR IV VIS-A-VIS THE SUBROUTINE CLUD. BEGINNING WITH THE
1581     C VECTOR X(0), WHICH IS OBTAINED BY BACK-SUBSTITUTION IN THE
1582     C LU-DECOMPOSITION (THE SUBROUTINE CBS), A SEQUENCE OF APPROX-
1583     C IMATE SOLUTIONS X(I) IS GENERATED BY THE ALGORITHM
1584     C                    R(I) = A X(I) - B      (COMPUTED BY AX1B)
1585     C                    A C(I) = R(I)          (SOLVED BY BS)
1586     C                    X(I+1) = X(I) - C(I).
1587     C THIS ITERATION IS CONTINUED UNTIL THE QUOTIENT OF //C(I)// AND
1588     C //X(I)// IS LESS THAN MACHEPS OR UNTIL THIS QUOTIENT EXCEEDS HALF
1589     C ITS VALUE FROM THE PREVIOUS ITERATION.  THE 1-NORM IS USED BECAUSE
1590     C IT IS THE EASIEST NORM TO COMPUTE. A DISCUSSION OF ITERATIVE
```

```
C REFINEMENT (IMPROVEMENT) MAY BE FOUND IN EITHER WILKINSON
C (1965,CHAPTER 4) OR FORSYTHE AND MOLER (1967). THE CONVERGENCE
C CRITERION IS PATTERNED AFTER THE ALGOL PROCEDURE 'UNSYM ACC
C SOLVE' (WILKINSON AND REINSCH,1971,CONTRIBUTION I/7).
C
      INTEGER N,ADIM,TDIM,IV(1),IER
      COMPLEX A(ADIM,N),T(TDIM,N),X(1),B(1),R(1)
C
C  N   -> ORDER OF THE SYSTEM OF LINEAR EQUATIONS.
C ADIM -> ROW DIMENSION OF THE ARRAY A.
C  A   -> TWO-DIMENSIONAL ARRAY CONTAINING THE COEFFICIENT MATRIX
C         OF THE SYSTEM OF LINEAR EQUATIONS.
C TDIM -> ROW DIMENSION OF THE ARRAY T.
C  T   -> TWO-DIMENSIONAL ARRAY CONTAINING THE LU-DECOMPOSITION OF
C         OF THE COEFFICIENT MATRIX VIS-A-VIS THE SUBROUTINE CLUD.
C  IV  -> VECTOR CONTAINING THE INTERCHANGE INFORMATION GENERATED
C         BY THE SUBROUTINE CLUD DURING THE COMPUTATION OF THE
C         LU-DECOMPOSITION OF THE COEFFICIENT MATRIX.
C  X   <- VECTOR FOR RETURNING THE COMPUTED SOLUTION OF THE
C         SYSTEM OF LINEAR EQUATIONS, I.E., X(IER).
C  B   -> VECTOR CONTAINING THE RIGHT-HAND SIDE OF THE SYSTEM OF
C         LINEAR EQUATIONS.
C  R   <- SCRATCH VECTOR USED TO HOLD THE RESIDUAL VECTOR R(I) AND
C         CORRECTION VECTOR C(I).
C IER  <- VARIABLE WHICH WILL CONTAIN 0 IF THE COEFFICIENT MATRIX
C         IS COMPUTATIONALLY SINGULAR, I.E., IV(N)=0; A POSITIVE
C         INTEGER IF ITERATIVE REFINEMENT CONVERGED; AND A NEGATIVE
C         INTEGER IF ITERATIVE REFINEMENT FAILED TO MEET THE
C         CONVERGENCE CRITERION.  IN ALL CASES, THE ABSOLUTE VALUE
C         OF IER INDICATES THE NUMBER OF ITERATIONS PERFORMED.
C
C     SUBROUTINES REQUIRED: CBS,CAXMB
C
      INTEGER I
      REAL QUOT,XN1,RN1
      COMPLEX Z
      REAL CABS,REAL,AIMAG
      CABS(Z) = ABS( REAL(Z) ) + ABS( AIMAG(Z) )
C
      IER = 0
      DO 8100 I = 1, N
 8100      X(I) = B(I)
      IF ( IV(N) .EQ. 0 ) RETURN
      QUOT = 1.0
C
C COMPUTE THE FIRST APPROXIMATE SOLUTION, X(0), BY BACK-SUBSTITU-
C TION IN THE LU-DECOMPOSITION OF THE COEFFICIENT MATRIX.
C
      CALL CBS(N,TDIM,T,IV,X)
C
C COMPUTE THE RESIDUAL VECTOR R(I) = A X(I) - B AND THEN THE
C CORRECTION VECTOR C(I) BY BACK-SUBSTITUTION.
C
 8200      CALL CAXMB(N,ADIM,A,X,B,R)
           CALL CBS(N,TDIM,T,IV,R)
```

```
1646                IER = IER + 1
1647          C
1648          C COMPUTE THE 1-NORMS OF X(I) AND C(I) AND REPLACE THE CONTENTS
1649          C OF X, CURRENTLY X(I), BY X(I+1).
1650          C
1651                XN1 = 0.0
1652                RN1 = 0.0
1653                DO 8210 I = 1, N
1654                    XN1 = XN1 + CABS(X(I))
1655                    RN1 = RN1 + CABS(R(I))
1656           8210     X(I) = X(I) - R(I)
1657          C
1658          C SUCCESSIVE ITERATES X(I) ARE REGARDED AS CONVERGENT IF TWICE
1659          C THE QUOTIENT OF THE 1-NORMS OF C(I) AND X(I) IS LESS THAN ITS
1660          C PREVIOUS VALUE. THUS THIS SEQUENCE OF QUOTIENTS MUST DECREASE
1661          C MORE RAPIDLY THAN THE SEQUENCE 2**-I, OR NONCONVERGENCE WILL
1662          C BE INDICATED BY SETTING THE ITERATION COUNT, IER, NEGATIVE. THE
1663          C ITERATE X(I) IS ACCEPTED AS THE SOLUTION IF THE QUOTIENT OF
1664          C THE 1-NORMS OF C(I) AND X(I) IS LESS THAN MACHEPS.  NOTE THAT
1665          C EVEN THOUGH MACHEPS IS MACHINE DEPENDENT, EXPLICIT USE OF THIS
1666          C CONSTANT HAS BEEN AVOIDED BY USING THE MACHINE INDEPENDENT
1667          C DEFINITION OF THIS QUANTITY.
1668          C
1669                IF ( XN1 .GT. 0.0 ) RN1 = RN1 / XN1
1670                IF ( (RN1 + RN1) .LE. QUOT ) GO TO 8220
1671                  IER = -IER
1672                  RETURN
1673           8220 CONTINUE
1674                QUOT = RN1
1675                IF ( (1.0 + RN1) .NE. 1.0 ) GO TO 8200
1676              RETURN
1677          C
1678          C FORSYTHE,G.E. AND MOLER,C.B. 1967. COMPUTER SOLUTION OF LINEAR
1679          C   ALGEBRAIC SYSTEMS. ENGLEWOOD CLIFFS,N.J.: PRENTICE-HALL.
1680          C WILKINSON,J.H. 1965. THE ALGEBRAIC EIGENVALUE PROBLEM. OXFORD:
1681          C   CLARENDON PRESS.
1682          C WILKINSON,J.H. AND REINSCH,C. 1971. LINEAR ALGEBRA. HANDBOOK FOR
1683          C   AUTOMATIC COMPUTATION, VOL. II. BERLIN: SPRINGER-VERLAG.
1684          C                       THE UNIVERSITY OF MICHIGAN COMPUTING CENTER
1685          C                       NUMERICAL ANALYSIS LIBRARY - JULY 1975
1686              END


1687              SUBROUTINE CBS(N,TDIM,T,IV,B)
1688          C
1689          C NAASA  2.1.008 CBS       FTN-A 10-29-75     THE UNIV OF MICH COMP CTR
1690          C
1691          C SOLVES THE SYSTEM OF LINEAR EQUATIONS AX=B, WHERE A DENOTES
1692          C THE N X N COEFFICIENT MATRIX AND X AND B ARE N-VECTORS, BY
1693          C BACK-SUBSTITUTION IN THE LU-DECOMPOSITION OF A. THIS
1694          C DECOMPOSITION MUST BE PROVIDED IN THE ARRAY T AND VECTOR IV
1695          C VIS-A-VIS THE SUBROUTINE CLUD. THE LU-DECOMPOSITION MAY BE
1696          C EXPRESSED IN THE FORM
1697          C                L(N-1)*P(N-1)*...*L(1)*P(1)*A = U,
1698          C WHERE EACH L(J) IS THE IDENTITY MATRIX EXCEPT FOR THE SUB-
```

```
C DIAGONAL ELEMENTS IN COLUMN J, EACH P(J) IS A PERMUTATION
C MATRIX, AND U IS AN UPPER TRIANGULAR MATRIX. USING THIS
C NOTATION, THE BACK-SUBSTITUTION CONSISTS OF FORMING
C              Y = L(N-1)*P(N-1)*...*L(1)*P(1)*B
C AND SOLVING THE UPPER TRIANGULAR SYSTEM OF LINEAR EQUATIONS
C UX = Y, I.E., FOR I=N...1
C   X(I)=(Y(I)-U(I,N)*X(N)-...-U(I,I+1)*X(I+1))/U(I,I).
C THIS BACK-SUBSTITUTION YIELDS A VECTOR X WHICH IS THE EXACT
C SOLUTION OF A SYSTEM OF LINEAR EQUATIONS (A+E)X=B, WHERE
C //E// IS GENERALLY ON THE ORDER OF N*//A//*MACHEPS. THIS METHOD
C OF SOLVING SYSTEMS OF LINEAR EQUATIONS IS DESCRIBED IN BOTH
C WILKINSON (1965,CHAPTER 4) AND FORSYTHE AND MOLER (1967).
C
      INTEGER N,TDIM,IV(1)
      COMPLEX T(TDIM,N),B(1)
C
C   N  -> ORDER OF THE SYSTEM OF LINEAR EQUATIONS.
C TDIM -> ROW DIMENSION OF THE ARRAY T.
C   T  -> TWO-DIMENSIONAL ARRAY CONTAINING THE LU-DECOMPOSITION
C         OF THE COEFFICIENT MATRIX VIS-A-VIS THE SUBROUTINE CLUD.
C  IV  -> VECTOR CONTAINING THE INTERCHANGE INFORMATION GENERATED
C         BY THE SUBROUTINE CLUD DURING THE COMPUTATION OF THE
C         LU-DECOMPOSITION OF THE COEFFICIENT MATRIX.
C   B  -- VECTOR CONTAINING THE RIGHT-HAND SIDE OF THE SYSTEM OF
C         LINEAR EQUATIONS. THE CONTENTS OF B ARE REPLACED BY THE
C         ELEMENTS OF THE SOLUTION.
C
      INTEGER I,K,KP1,L
      COMPLEX TMP
C
C REPLACE THE CONTENTS OF THE VECTOR B BY THE VECTOR
C       ( L(N-1) (P(N-1) ... ( L(1) (P(1) B)...).
C THIS COMPUTATION IS PERFORMED IN N-1 STAGES, WHERE DURING THE
C K-TH STAGE, THE CONTENTS OF B ARE REPLACED BY L(K)( P(K) B ).
C THE PERMUTATION MATRICES P(K) ARE DEFINED BY THE VECTOR IV, I.E.,
C MULTIPLICATION BY P(K) SIMPLY INTERCHANGES THE K-TH AND IV(K)-TH
C ELEMENTS OF THE VECTOR.  THE LOWER TRIANGULAR MATRIX L(K) IS THE
C IDENTITY MATRIX EXCEPT FOR THE SUBDIAGONAL ELEMENTS OF THE K-TH
C COLUMN, WHICH ARE STORED IN THE CORRESPONDING ELEMENTS OF THE
C ARRAY T.
C
      DO 8110 K = 1, N
          IF ( K .GE. N ) GO TO 8110
          L = IV(K)
          TMP = B(L)
          B(L) = B(K)
          B(K) = TMP
          KP1 = K + 1
          DO 8100 I = KP1, N
 8100         B(I) = B(I) + T(I,K) * TMP
 8110 CONTINUE
C
C REPLACE THE CONTENTS OF THE VECTOR B BY THE SOLUTION TO THE SYSTEM
C OF LINEAR EQUATIONS WITH UPPER TRIANGULAR COEFFICIENT MATRIX U
C AND RIGHT-HAND SIDE VECTOR B.  THE USUAL FORMULAS FOR THE BACK-
```

```
1754     C SUBSTITUTION, WHICH ARE BASED ON THE SUCCESSIVE ROWS OF THE MATRIX
1755     C AND ARE SUITABLE WHEN INNER-PRODUCTS ARE ACCUMULATED, ARE NOT
1756     C EMPLOYED.  THE COMPUTATION HAS INSTEAD BEEN ARRANGED TO REFERENCE
1757     C THE SUCCESSIVE COLUMNS OF U.  THUS AFTER B(I) HAS BEEN COMPUTED,
1758     C IT IS REMOVED FROM THE SYSTEM BY SUBTRACTING B(I) TIMES THE I-TH
1759     C COLUMN OF U FROM THE RESIDUAL VECTOR B(1)...B(I-1).
1760     C
1761           K = N
1762      8200     B(K) = B(K) / T(K,K)
1763               IF ( K .LE. 1 ) RETURN
1764               TMP = -B(K)
1765               KP1 = K
1766               K = K - 1
1767               DO 8210 I = 1, K
1768      8210         B(I) = B(I) + T(I,KP1) * TMP
1769               GO TO 8200
1770     C
1771     C FORSYTHE,G.E. AND MOLER,C.B. 1967. COMPUTER SOLUTION OF LINEAR
1772     C   ALGEBRAIC SYSTEMS. ENGLEWOOD CLIFFS,N.J.: PRENTICE-HALL.
1773     C WILKINSON,J.H. 1965. THE ALGEBRAIC EIGENVALUE PROBLEM. OXFORD:
1774     C   CLARENDON PRESS.
1775     C                          THE UNIVERSITY OF MICHIGAN COMPUTING CENTER
1776     C                          NUMERICAL ANALYSIS LIBRARY - JULY 1975
1777           END


1778           SUBROUTINE CAXMB(N,ADIM,A,X,B,R)
1779     C
1780     C NAASA  2.1.009 CAXMB    FTN-A 10-29-75      THE UNIV OF MICH COMP CTR
1781     C
1782     C COMPUTES R = AX-B, WHERE A IS AN N X N MATRIX AND R, X,
1783     C AND B ARE N-VECTORS, IN TWICE THE PRECISION OF THE DATA.
1784     C THE ACCURATE COMPUTATION OF THIS VECTOR IS CRUCIAL TO THE
1785     C SUCCESS OF THE ITERATIVE REFINEMENT ALGORITHM AS IMPLEMENTED BY
1786     C THE SUBROUTINE CIR.  ITERATIVE REFINEMENT IS GENERALLY NOT
1787     C CONVERGENT IF THESE RESIDUALS ARE COMPUTED IN THE PRECISION OF
1788     C THE SYSTEM OF LINEAR EQUATIONS.
1789     C
1790           INTEGER N,ADIM
1791           COMPLEX A(ADIM,N),X(1),B(1),R(1)
1792     C
1793     C   N  -> ORDER OF THE MATRIX A.
1794     C ADIM -> ROW DIMENSION OF THE ARRAY A.
1795     C   A  -> TWO-DIMENSIONAL ARRAY CONTAINING THE N X N MATRIX A,
1796     C             USUALLY THE COEFFICIENT MATRIX OF THE SYSTEM OF LINEAR
1797     C             EQUATIONS.
1798     C   X  -> VECTOR, USUALLY AN APPROXIMATE SOLUTION TO THE SYSTEM
1799     C             OF LINEAR EQUATIONS AX=B.
1800     C   B  -> VECTOR, USUALLY THE RIGHT-HAND SIDE VECTOR OF THE
1801     C             SYSTEM OF LINEAR EQUATIONS.
1802     C   R  <- RESIDUAL VECTOR, I.E., AX - B.
1803     C
```

```
1804           INTEGER I,J
1805           DOUBLE PRECISION SR,SI,AR,AI,XR,XI
1806       C
1807           DO 8110 I = 1, N
1808               SR = -DBLE( REAL(B(I)) )
1809               SI = -DBLE( AIMAG(B(I)) )
1810               DO 8100 J = 1, N
1811                   AR = DBLE( REAL(A(I,J)) )
1812                   AI = DBLE( AIMAG(A(I,J)) )
1813                   XR = DBLE( REAL(X(J)) )
1814                   XI = DBLE( AIMAG(X(J)) )
1815                   SR = SR + ( AR * XR - AI * XI )
1816        8100       SI = SI + ( AR * XI + AI * XR )
1817        8110       R(I) = CMPLX( SNGL(SR), SNGL(SI) )
1818           RETURN
1819       C
1820       C                          THE UNIVERSITY OF MICHIGAN COMPUTING CENTER
1821       C                          NUMERICAL ANALYSIS LIBRARY - JULY 1975
1822           END
```