NASA CR-132444

ANALYTIC MODELS OF DUCTED TURBOMACHINERY TONE NOISE SOURCES

Volume II: Subprogram Documentation

by T. L. Clark, U. W. Ganz, G. A. Graf, and J. S. Westall

D6-43296-2

May 1974

Prepared under contract NAS1-12257 by

Boeing Commercial Airplane Company

P.O. Box 3707

Seattle, Washington 98124

for

Langley Research Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-132444		

4. Title and Subtitle	5. Report Date
ANALYTIC MODELS OF DUCTED TURBOMACHINERY TONE NOISE SOURCES--Volume II: Subprogram Documentation	May 1974
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
T.L. Clark, U.W. Ganz, G.A. Graf, and J.S. Westall	D6-43296-2
9. Performing Organization Name and Address	10. Work Unit No.
Boeing Commercial Airplane Company P.O. Box 3707 Seattle, Washington 98124	11. Contract or Grant No. NAS1-12257
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Contractor Report
National Aeronautics and Space Administration Washington, D.C. 20546	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

Volume I of this report describes the analytic models developed for computing the periodic sound pressures of subsonic fans and compressors in an infinite, hardwall annular duct with uniform flow. The basic sound-generating mechanism is the scattering into sound waves of velocity disturbances appearing to the rotor or stator blades as a series of harmonic gusts. The models include component interactions and rotor alone. Volume II of this report describes the computer subprograms developed for numerical computations of sound pressure mode amplitudes from the analysis. Volume III presents some test case results from the computer programs.

17. Key Words (Suggested by Author(s))		18. Distribution Statement
Analytic models	Tone noise	Unclassified--Unlimited
Turbomachinery noise	Thin airfoils	
Fan duct		
Acoustic modes		

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	287	

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

CONTENTS

CONTENTS-Concluded

SYMBOLS

a	a constant
a_z, a_ϕ	eddy transverse length scale
A	radial location of the maximum distortion in the cone model
A_n, $A_{n,\kappa 2}$	n^{th} order Glauert coefficients
$A^{\pm}_{mn}$	a mode amplitude, or coefficient of eigenfunction expansion of the pressure spectral density
b_1, b_2	blade spacings of a two-dimensional cascade of blades
C, C_1, C_2, C_{K1}, C_{K2}	airfoil chord lengths
$C_3 \cdots C_{14}$	switches used in the generalized equations for the potential flow field interactions
C_{D1}, C_{D2}	airfoil section drag coefficient
d	axial spacing between midchords of two blade rows
$\frac{dC_L}{d\alpha}$, $\frac{dC_{L1}}{d\alpha}$, $\frac{dC_{L2}}{d\alpha}$, $\frac{dC_{L3}}{d\alpha}$, $\left(\frac{dC_L}{d\alpha}\right)_1$, $\left(\frac{dC_L}{d\alpha}\right)_{K1}$	slope of steady lift versus angle of attack
$\hat{d}_{mn}$	a function defined by equation (B41)
$\hat{D}_{mn}$	a function defined by equation (2.1.22)
D_{mn}	a function defined by equation (2.1.15)
$\vec{e}$	unit vector perpendicular to airfoil chordline
e_ϕ	z-component of $\vec{e}$
e_z	ϕ-component of $\vec{e}$
E_z, E_ϕ, $E_{z\phi}$	bandpass filter factor; see equation (3.3.23)

f	ratio of maximum camber to the half-chord of a thin airfoil
F	strength of single dipole or a surface distribution of dipoles; also, function defined by equation (3.1.15)
$\hat{F}$, $\hat{F}_j$	spectral density of the strength of a single dipole, or a surface distribution of dipoles
F_α	function defined by equation (3.1.20)
F_f	function defined by equation (3.1.21)
$G_{m\nu\ell}$, $G_{mn\sigma}$	complex term in the equation for the induced velocities resulting from a potential flow field interaction
$g_{\sigma,1}$, $G_{\sigma,1}$, $G_{\kappa,K1}$	airfoil acoustic response function, defined by equation (3.1.39)
h	oblique gust wave number
$\hat{H}$	the nondimensional dipole surface density response function for convected, harmonic gusts
$H_m^{(1)}$	Hankel function of the first kind of order m
$H_o^{(2)}$	Hankel function of the second kind of zeroth order
$H_1^{(2)}$	Hankel function of the second kind of order one
$H_{\sigma,2}$, $H_{\kappa,K2}$	complex term in the equation for the induced velocities resulting from a potential flow field interaction, equation (3.2.31)
I_ℓ	modified Bessel function of the first kind and order ℓ
j	dummy index or blade number index
J	$J_0 + iJ_1$
J_m	Bessel function of order m
J_0	Bessel function of order zero
J_1	Bessel function of order one

k	wave number; also, dummy index
K	one-dimensional Fourier space wave number
K_L	Kemp-Sears lift response function
$K_{mn}^{\pm}$	axial propagation wave number of mn^{th} mode
$K_{\sigma,1}$, $K_{\kappa,K1}$	complex conjugate of Kemp-Sears lift response function
ℓ	dummy index, or harmonic index
L	airfoil section lift force
$\hat{L}_j$	spectral density of airfoil section lift force
L_z, L_ϕ	eddy axial length scale
m	polar angle harmonic index, or spinning mode index
M	duct uniform, axial flow velocity for acoustic calculations
M_E, M_{1E}, M_{2E}, M_{3E}, $M_{E,K}$	mean exit velocity from a blade row, relative to the blades of the row
M_I, M_{1I}, M_{2I}, M_{3I}, $M_{M,K1}$	mean inlet velocity to a blade row, relative to the blades of the row
M_M, M_{1M}, M_{2M}, M_{3M}, $M_{M,1}$, $M_{M,2}$, $M_{M,K}$, $M_{M,K1}$, $M_{M,K2}$	mean velocity through a blade row, relative to the blades of the row
M_T	tip velocity of rotating blades
M_z, M_{1z}, M_{2z}, M_{3z}, $M_{z,K}$	mean axial flow velocity
$\bar{M}_z$	mean axial flow velocity when there is steady distortion
n	dummy index, harmonic index, or radial mode index
N, N_1, N_2, N_3, N_{K2}	number of blades in a blade row

p	linear pressure perturbation
$\hat{p}$	spectral density of linear pressure perturbation
q	exponent in the "power law," equation (3.3.8)
$Q_{\sigma,2}$	complex term in the equation for the induced velocities resulting from a potential flow field interaction, equation (3.2.3)
$\vec{r}$	position vector, field location
$\vec{r}_o$, $\vec{r}_s$	position vector, source location
R	radial location of eddy center
R_m	unnormalized radial eigenfunction
R^*	distance from origin in Prandtl-Glauert scaled coordinates
s_{mn}	axial propagation wave number of mn^{th} mode in Prandtl-Glauert scaled coordinates and frequency
S	Sears function
SPL	sound pressure level, measured in decibels
t	time at field location
t_o, t_s	time at source location
T	function defined by equation (3.1.17)
$T^\dagger$	Filotas lift response function
T_z, T_ϕ, $T_{z,\phi}$	eddy temporal length scale
U	relative velocity of two-dimensional cascade of blades
$\hat{u}$, $\hat{u}_j$	spectral density of perturbation velocity
V	gust convection velocity
V_A	value of the maximum velocity distortion in the cone model

V_1	value of the velocity at the outer radius in the cone model
W, W_j	perturbation velocity, viscous wake defect, unsteady induced velocity
$\bar{W}$	single spatial Fourier transform of W
$\bar{\bar{W}}$	double spatial Fourier transform of W
W_z, W_ϕ	perturbation velocity components at the eddy center
$W_{z\ell}$, $W_{\phi\ell}$	ℓ^{th} Fourier series coefficient of the eddy velocity components
Y, Y', $\bar{Y}'$, Y_1', Y_j'	rectangular coordinate
$Y'_{\ell j}$	rectangular coordinate
Y_m	Neumann function of order m
Y_o	width of viscous wake; also, Neumann function of order o
Z, Z', Z', Z_1'	rectangular coordinate
$Z'_{\ell j}$	rectangular coordinate
$z_{M.C.}$	axial position of midchord plane
*	multiplier
α	mean blade angle of attack
β	relative stagger angle
β_{mn}	square root term defined by equation (2.1.16)
γ, γ_1, γ_2, γ_{K1}, γ_{K2}	stagger angle
$\hat{\gamma}$	Helmholtz equation Green's function
Γ	acoustic propagator
Γ_2 , Γ_{K2}	steady-state circulation of cascade airfoil

δ	Dirac delta function
$\delta_{m-\ell,\sigma N}$	Kronecker delta symbol
Δ_o	factor in viscous wakes formula, equation (3.1.35)
$\bar{\Delta}_z$, $\bar{\Delta}_\phi$	axial eddy strength modulation function
ε	small, positive constant; also, unit step function
ζ	rectangular coordinate normal to the airfoil
η	hub-to-tip ratio, annular duct inner radius
Θ	inverse cosine of nondimensionalized chordwise coordinate
κ	temporal or spacial harmonic index
$\kappa^{\pm}_{mn}$	chordwise compactness parameter; see equation (2.2.18)
λ, λ_1, $\lambda_{\sigma,1}$, $\lambda_{\kappa,K1}$	wave number; also, reduced complex frequency of the chordwise velocity distribution
Λ_ℓ, Λ_σ	Fourier coefficients of typical wake profile
$\mu_{\sigma,1}$	complex frequency of the chordwise velocity distribution
μ_{mn}	annular duct eigenvalue
ν, ν_ℓ, ν_σ, ν_1, $\nu_{\sigma,1}$, $\nu_{\kappa,K1}$	reduced frequency
ξ	chordwise rectangular coordinate
ξ'	chordwise rectangular coordinate nondimensionalized to the half-chord
ρ, ρ_o, ρ_s	polar radial coordinate of cylindrical coordinate system
σ	temporal Fourier series coefficient, dummy index used in summations
τ	time delay resulting from the axial distance between the midchord plane and the eddy center position at the temporal origin; also, $\tau - \tau o$

ϕ , ϕ_o , ϕ_s , $\phi_{M.C.}$ ϕ_j , ϕ_s' , Φ	polar angle coordinate of cylindrical coordinate system
Φ	polar angle coordinate of eddy center
ψ	relative exit flow angle; also, oblique gust angle
ω , ω^* , $\omega_{\sigma,1}$	angular frequency
Ω	angular velocity of rotor

Subscripts:

E	blade row exit flow
I	blade row inlet flow
j	j^{th} blade
K	either K1 or K2
K1	sound-producing blade row
K2	velocity-inducing blade row
ℓ	spatial harmonic index
m	spinning mode index
M	blade row mean flow
M. C.	midchord point location
n	radial mode index
S	source
z	axial direction, axial eddy velocity component
κ	either σ or ℓ
σ	temporal Fourier series index
ϕ	angular direction, angular eddy velocity component
o	source

1	inlet stator parameter, upstream component in viscous wakes interaction, or unsteady lift-producing component in potential flow field interaction
2	rotor parameter, downstream component in viscous wakes interaction, or velocity-inducing component in potential flow field interaction
3	outlet stator parameter

Superscripts:

$\pm$	downstream (+) and upstream (-) propagation
$\dagger$	complex conjugate
$*$	complex conjugate; also, generalized Prandtl-Glauert transform, variable
$'$	blade-attached rectangular coordinate non-dimensionalized to the half-chord
$\rightarrow$	vector
$\hat{}$	temporal Fourier integral transform
$\bar{}$	spacial Fourier series transform
$\_$	blade-attached rectangular coordinate system in viscous wakes interaction, polar angle in the rotating system, or averaged value of a variable
$=$	double spatial Fourier series transform

1.0 INTRODUCTION

The subprograms described herein are designed to calculate the acoustic pressure annular duct mode amplitudes for a given harmonic of blade passing frequency, with upstream or downstream propagation, for the acoustic sources described in volume I. Subroutines AAAAA, AABAA, BCDAA, and BBCAA are the primary subprograms provided for this purpose. These subroutines, along with the secondary subprograms, are described in section 3. The primary and secondary subprograms receive standardized treatment, if they are considered as special-purpose routines dependent on the details of the primary subroutine; otherwise, as in the case of the general-purpose math routines, the secondary subprograms receive nonstandardized, or general-purpose, treatment.

A subprogram is treated in a standardized way by having all of its FORTRAN variable names drawn from a dictionary of such names. Thus, any name used in any of the standardized subprograms is defined in the dictionary and nowhere else, and has the definition and use given it in the dictionary and no other, regardless of the subprogram in which it is used.

In the description of a subprogram, the question of output variable accuracy is generally answered by placing the operation performed in producing the output in one of a number of categories. Thus, the output variable may be limited in accuracy by the particular computer, or machine, or by the nature of the algorithm. If the algorithm is of the converging iteration type, then the convergence criterion sets the accuracy. If the algorithm results from an approximation formula, then the remainder term associated with the approximation sets the accuracy. These are not always specified in detail for each subprogram, but a note is made when necessary to indicate whether the accuracy is limited by the algorithm or not. Comparison with other sources is made when comparable numbers are available.

2.0 DICTIONARY

This dictionary replaces the list of definitions usually included in a subprogram description for all the subprograms written specifically for this work. General mathematical routines are documented in the usual way. The purpose of the dictionary is to standardize the use and definition of all FORTRAN variable names within the several primary and secondary subprograms. This is desirable for purposes of modifying or updating the routines as well as aiding in understanding the coded algorithms and the relationships between the different subprograms.

2.1 Guide to Dictionary

For each FORTRAN name, the dictionary indicates:

1) The subprograms in which the variable appears; see location code

2) The function performed by the variable in each subprogram in which it appears; see function code

3) The variable definition, by a phrase or sentence

Items 1 and 2 are contained in the location-function code (LOC-FNC code) occupying the middle column of the dictionary. In many cases, the item 3 definitions contain equations and figure numbers. All equation numbers refer to the equations in appendix I of volume I; all figure numbers refer to the figures of volume I.

LOCATION CODE

Subprogram	Code
AAAAA	1
ZEROS	3
EQATION	4
UNEGNFN	5
EGNNORM	6
FACTINT	7
AABAA	9
FACTIN2	10
LIFTFN2	11
BCDAA	12
EGNVAL2	13
FACTIN3	14
LIFTFN3	15
DISINT	16
BBCAA	17
NONCPT	18
FACTIN4	19
FUNIN4	20
LIFTFN4	21

Subprogram	Code
APROX1	50
APROX2	51
JARRATT	52
GAUSS	53
BSSLS	54
BF4F	55
MLTUP	56
GAUSS2	57
ROCABES	58
ALGAMF	59
BESIE	60
BESJLA	61
BESNX	62
BESIK	63
SICI	64
GRTHFCN	65

FUNCTION CODE

Function	Code
Calling sequence input	1
Calling sequence input/output	2
Special calling sequence: the few variables and arrays that are reused in subsequent calls to the primary subroutine—they must not be changed by the user	2S
Calling sequence output	3
Internal name	4
Name in common	5
Name of subroutine	6
Name of function subprogram	7

2.2 Dictionary of FORTRAN Names

FORTRAN name	LOC-FNC code	Definition
AAAAA	1-6	Primary subroutine which calculates the acoustic pressure mode amplitudes resulting from the interaction of a turbomachinery blade row with the viscous wakes of another upstream-located blade row
AABAA	9-1	Primary subroutine which computes the acoustic pressure mode amplitudes resulting from the interaction of a turbomachinery blade row with the potential flow field of an adjacent blade row

FORTRAN name	LOC-FNC code	Definition
ABN	14-4	Array of dimension 2 which contains the cosine and sine distortion coefficients $a_{\|\ell\|}(\rho)$, $b_{\|\ell\|}(\rho)$ of index $\|\ell\|$ and radial position ρ
ABSLAM	11-4	Absolute value of the complex variable LAMDA, $\|\lambda\|$
ABSKAPA	18-4	Absolute value of RKAPA, $\|\kappa^{\pm}_{mn\sigma}\|$
ABSNU	15-4,18-4, 21-4	Absolute value of the reduced frequency RNU, $\|\nu\|$
ALFA	19-4	The blade angle of attack, α; this quantity is input in AR(I,11,K) as a function of radial position
ALGAMF	59-6	The standard LRC subroutine which computes the log of the gamma function for complex arguments
ALPHA	21-4,21-5, 65-5	Tan θ, where θ is the gust yaw angle used in the Filotas lift response function; also, name of common block containing ALPHA
ALPHAMN	1-3,9-3, 12-3,17-3	Complex array of dimension NDIM x MDIM which contains the matrix of mode amplitudes, α_{mn}, where α_{mn} = ALPHAMN(N,I) with M = MUSE(I)
APROX1	3-4,50-6	Subroutine which calculates approximate zeros of equation (6), where $.2 \leq \eta < 1.0$ (η is the hub-to-tip ratio)

FORTRAN name	LOC-FNC code	Definition
APROX2	3-4,51-6	Subroutine which calculates approximate zeros of equation (6), where $0 \leq \eta < .2$
AR	1-1,7-1,9-1, 10-1,12-1,14-1, 17-1,19-1	Array of dimension MAXDIM x MAXJ x 3, where AR(I,J,K) contains data described as follows: K = 1: Inlet stator data K = 2: Rotor data K = 3: Outlet stator data J = 1: Nondimensional duct radial position, ρ J = 2: Nondimensional chord, $C(\rho)$ J = 3: Not used J = 4: Drag coefficient, $C_D(\rho)$ J = 5: Steady-state lift coefficient, $C_L(\rho)$, which is not required as an input for any of the existing primary subroutines J = 6: Derivative of C_L with respect to incident angle, α $\frac{dC_L}{d\alpha}(\rho)$ J = 7: Relative inflow Mach number of a blade row, $M_I(\rho)$; see figure 3 J = 8: Relative exit flow Mach number of a blade row; $M_E(\rho)$; see figure 3 J = 9: Axial flow Mach number, $M_Z(\rho)$; see figure 3

FORTRAN name	LOC-FNC code	Description
	Note: ARMISC(18+K) can be zero, in which case no Glauert coefficients are input.	J = 10: Glauert coefficients of order 0 J = 11: Glauert coefficients or order 1 . . . J = ARMISC(18+K) + 9: Glauert coefficients of order ARMISC(18+K) -1
	If ARMISC(25) = 3,	J = ARMISC(18+K) + 10: The ratio of maximum blade camber to the half-chord, $f(\rho)$, used in the lift response function J = ARMISC(18+K) + 11: The blade angle of attack, $\alpha(\rho)$, used in the lift response function
	If ARMISC(22) = 2,	J = ARMISC(18+K) + P: Parameter a_1, which is used in the power model, where: P = 10 if ARMISC(25) $\neq$ 3 P = 12 if ARMISC(25) = 3
	If ARMISC(22) = 3	J = ARMISC(18+K) + P: Cosine distortion coefficient of index r, a_r J = ARMISC(18+K) + P+1: Sine distortion coefficient of index r, b_r J = ARMISC(18+K) + P+2: Cosine distortion coefficient of index 2·r, $a_{2 \cdot r}$ . . .

FORTRAN name	LOC-FNC code	Description
If ARMISC(22) = 3 (concluded)		. . . J = ARMISC(18+K) + P+2·MAXCOEF+1: Sine distortion coefficient of index MAXCOEF·r, $b_{MAXCOEF \cdot r}$, where: P = 10 if ARMISC(25) ≠ 3; P = 12 if ARMISC(25) = 3; r = MULTFCT
		I = 1: The number of radial positions, where the parameters contained in the AR array are defined or 0 if only a value averaged in the radial direction is given for the $(K,J)^{th}$ parameter in the AR array
		I = 2: Radially averaged value of the $(K,J)^{th}$ parameter of the AR array
		I = 3: I = 4: . . . Set of values corresponding to nondimensional duct radial positions. The nondimensional duct radial positions *must be in increasing order*.
ARG	16-4	Argument $-\ell\phi$ of the exponential in equation (48)
ARGETA	5-4	Product of eigenvalue and hub-to-tip ratio, $\mu_{mn} \cdot \eta$
ARGEXP	1-4,9-4	Argument of the exponential in the constant factor of the integrand; see equations (33), (45), (53), (57)

FORTRAN name	LOC-FNC code	Description
ARGEXP2	7-4	Argument of the exponential in the oscillatory factor of the integrand; see equation (36)
ARGS	5-4	Argument ($\mu_{mn} \cdot S$) of the unnormalized duct radial eigenfunction R_m in equation (6). It is actually the product of an eigenvalue μ_{mn} and the dummy argument S of subroutine UNEGNFN.
ARHO	10-4	Variable $a_{\kappa,K1}(\rho)$ as defined in equation (38)
ARMISC	1-1,7-1,9-1, 10-1,12-1,14-1, 17-1,19-1	Array of dimension 40, where ARMISC(I) contains data described as follows: I = 1: Nondimensionalized average distance between the midchord planes of the inlet guide vanes and the rotor; see figure 4 I = 2: Nondimensional average distance between the midchord planes of the rotor and the outlet guide vanes; see figure 5 I = 3: Hub-to-tip ratio, η I = 4: Option IFLOW, where: -1 indicates upstream sound propagation 1 indicates downstream sound propagation

FORTRAN name	LOC-FNC code	Description
ARMISC (continued)		I = 5: Option ISOROS, where:
		1 indicates inlet stator-rotor interaction
		2 indicates rotor-outlet stator interaction
		I = 6: Option ITRACE, where:
		0 indicates no printout
		1 indicates printout from primary subroutine
		2 indicates printout from primary subroutines and subroutine ZEROS
		I = 7: Rotor blade tip Mach number, M_T
		I = 8: Number of inlet stator vanes, N_{ISV}
		I = 9: Number of outlet stator vanes, N_{OSV}
		I = 10: Number of rotor blades, N_{RB}
		I = 11: Not used
		I = 12: Phase angle for adjustment of skewness of the incident wake at the outlet stator, ϕ_{OS}, in radians; see figure 14

FORTRAN name	LOC-FNC code	Description
ARMISC (continued)		I = 13: Phase angle for adjustment of skewness of the incident wake at the rotor, ϕ_R, in radians; see figure 14
		I = 14: Harmonic index, σ
		I = 15: Axial position of the inlet stator, Z_{IS}
		I = 16: Axial position of the outlet stator, Z_{OS}
		I = 17: Axial position of the rotor, Z_R
		I = 18: Option IAERO, where in a potential flow field interaction: -1 indicates the upstream blade row is the sound generator 1 indicates the downstream blade row is the sound generator
		I = 19: Number of inlet stator vane Glauert coefficients
		I = 20: Number of rotor blade Glauert coefficients
		I = 21: Number of outlet stator vane Glauert coefficients

FORTRAN name	LOC-FNC code	Description
ARMISC (continued)		I = 22: Distortion model selector, where: 0 indicates no distortion 1 indicates distortion is represented by the cone model; see equation (48) 2 indicates distortion is represented by the power model; see equation (49) 3 indicates that the distortion coefficients are input; see equation (50)
		I = 23: Distortion input, where: If ARMISC(22) = 1, ARMISC(23) = VADBV1. If ARMISC(22) = 2, ARMISC(23) = Q. If ARMISC(22) = 3, ARMISC(23) = MAXCOEF.
		I = 24: Distortion input, where: If ARMISC(22) = 1, ARMISC(24) = CAPADIS. If ARMISC(22) = 3, ARMISC(24) = MULTFCT.
		I = 25: Lift response function selector, where: 2 indicates the generalized Sears lift response function (LIFTFN2) used with the primary subroutine AABAA; see equation (24)

FORTRAN name	LOC-FNC code	Description
ARMISC (continued)		I = 25: 3 indicates the combination of lift response functions as developed in reference 6 (LIFTFN3), or the lift response function for noncompact source theory NONCPT (see ARMISC [38]). It can be used with the primary subroutines AAAAA, BBCAA, and BCDAA; see equation (22). 4 indicates the Filotas lift response function (LIFTFN4) used with the primary subroutine BBCAA; see equation (25)
		I = 26: Not used
		I = 27: Not used
		I = 28: Nondimensional radial position of the eddy center, R
		I = 29: Angular position of the eddy center, Φ, in radians
		I = 30: Axial eddy velocity component, W_Z, at the eddy center, nondimensionalized with the average axial flow velocity; see figure 10.
		I = 31: Angular eddy velocity component, W_ϕ, at the eddy center, nondimensionalized with the average axial flow velocity; see figure 10
		I = 32: Nondimensional eddy length scale in the direction normal to the average flow velocity for the axial eddy velocity component, a_Z; see figure 10

FORTRAN name	LOC-FNC code	Description
ARMISC (concluded)		I = 33: Nondimensional eddy length scale in the direction normal to the average flow velocity for the angular eddy velocity component, a_ϕ; see figure 10
		I = 34: Nondimensional eddy length scale in the direction of the average flow velocity for the axial eddy velocity component; L_Z; see figure 10
		I = 35: Nondimensional eddy length scale in the direction of the average flow velocity for the angular eddy velocity component, L_ϕ; see figure 10
		I = 36: Upper bound of the frequency band considered in the generation of tone duct mode amplitudes by nonsteady distortion, B; see figure 11
		I = 37: Time when eddy center is located in rotor plane, τ
		I = 38: Compactness selector 0 indicates compact source option (LIFTFN3 is used) ≠0 indicates noncompact source option (NONCPT is used); can be used only if ARMISC(25) = 3
		I = 39: Not used
		I = 40: Not used

FORTRAN name	LOC-FNC code	Description
ARMUMN	1-2S,3-3,9-2S, 12-2S,13-3, 17-2S	Array of dimension NDIM x MDIM which contains the matrix of eigenvalues where ARMUMN(N,M) = μ_{MN}
AV	1-4,12-4,17-4	Array of dimension 11 which contains radially average values. An average value is calculated if a set of values is used. The input average value is used if this is indicated by a 0 in the corresponding element AR(1,J,K) or array AR. The contents of AV(I) are described as follows:
		I = 1: Midpoint of the subinterval locally used in the integration of the integral of equation (9)
		I = 2: { Average inlet stator vane chord if ISOROS = 1; Average rotor blade chord if ISOROS = 2 } Used with primary subroutine AAAAA only.
		I = 3: Average blade chord of the sound-generating blade row
		I = 4: { Average inlet stator vane drag coefficient if ISOROS = 1; Average rotor blade drag coefficient if ISOROS = 2 } Used with primary subroutine AAAAA only.
		I = 5: Not used

FORTRAN name	LOC-FNC code	Description
AV (concluded)		I = 6: Average derivative of the steady-state lift coefficient with respect to the angle of incidence for the sound-generating blade row
		I = 7: Average relative inflow Mach number of the sound-generating blade row
		I = 8: Average relative exit flow Mach number of the sound-generating blade row
		I = 9: Average axial flow Mach number of the sound-generating blade row
		I = 10: Average value of f, the ratio of the maximum camber to the half-chord for the sound-generating blade row
		I = 11: Average value of α, the blade angle of attack for the sound-generating blade row
AVSPAN	1-4,12-4,17-4	Midpoint of the subinterval locally used in the computation of the integral of equation (9). It is equivalenced to AV(1).
AXIALM	1-4,9-4, 12-4,17-4	Average axial Mach number of the sound-generating blade row
A1	4-4	A value of $J_M(x)$, Bessel function of first kind and order M
A2	4-4	A value of $J_{M+1}(x)$, Bessel function of first kind and order (M+1)

FORTRAN name	LOC-FNC code	Description
A3	4-4	A value of $Y_M(x)$, Bessel function of second kind (the Neumann function) and order M
A4	4-4	A value of $Y_{M+1}(x)$, Bessel function of second kind (the Neumann function) and order (M+1)
B	9-5,10-5	In the case of an inlet guide vane-rotor interaction (K1+K2 = 3), B = ARMISC(1). In the case of a rotor-outlet guide vane interaction (K1+K2 = 5), B = ARMISC(2).
BBCAA	17-6	Primary subroutine which computes the acoustic pressure mode amplitudes resulting from the interaction of a rotor with the nonsteady distortion resulting from a convected eddy
BCDAA	12-6	Primary subroutine which computes the acoustic pressure mode amplitudes resulting from the interaction of a rotor with a distorted inflow
BES	4-5,5-5,10-5 11-5,15-5,18-5, 19-5,21-5	Array of dimension 1000 in common block SCRATCH used as a scratch array by Bessel function subroutines
BESIE	19-4	The subroutine which computes $I_\ell(x)e^{-x}$ when I is the modified Bessel function, ℓ is an integer, and x is a real argument

FORTRAN name	LOC-FNC code	Description
BESIEJ	19-4	Array of dimension 2 which contains values of $I_\ell\left(\frac{\rho R}{a_j^2}\right) \cdot e^{-\rho R/a_j^2}$, $j = 1,2$ with I_ℓ a modified Bessel function of order ℓ; see equation (68).
BESIK	21-4,63-6	Subroutine which computes modified Bessel functions with real argument x, $I_o(x)$, $I_1(x)$, $K_o(x)$, $K_1(x)$
BESIO	21-4	A value of $I_o(x)$, a modified Bessel function of order 0
BESI1	21-4	A value of $I_1(x)$, a modified Bessel function of order 1
BESJLA	61-6	Subroutine that computes $J_\nu(x)$, a Bessel function of the first kind and order ν, where $x >> \nu$
BESJLAM	11-4	A variable which contains J_0(LAMDA) - i J_1(LAMDA)
BESKO	21-4	A value of $K_o(x)$, a modified Bessel function of order 0
BESK1	21-4	A value of $K_1(x)$, a modified Bessel function of order 1
BESNX	18-4,62-6	Subroutine which computes the Bessel function of the first kind $J_n(x)$ with no restrictions on the magnitude of the integer order n and the real argument x

FORTRAN name	LOC-FNC code	Description
BETAMN	1-4,9-4, 12-4,17-4	β_{mn}, which is defined in equation (1)
BF4F	4-4,5-4,11-4, 15-4,18-4,55-6	Standard LRC library subroutine which calculates the Bessel function of the second kind, or Neumann function, Y
BJHI	10-4	Array of dimension 250 containing the imaginary parts of the Bessel function of the first kind with complex arguments, $J_0(h_{K2}[\rho])$, $J_1(h_{K2}[\rho])$, ...
BJHR	10-4	Array of dimension 250 containing the real parts of the Bessel function of the first kind with complex arguments, $J_0(h_{K2}[\rho])$, $J_1(h_{K2}[\rho])$, ...
BJLAMI	11-4	Array of dimension 250 containing the imaginary parts of the Bessel function of the first kind with complex arguments, J_0(LAMDA), J_1(LAMDA)
BJLAMR	11-4	Array of dimension 250 containing real parts of the Bessel function of the first kind with complex arguments, J_0(LAMDA), J_1(LAMDA)
BJ1	15-4,18-4	$J_0(x)$, a value of the Bessel function of the first kind, with order zero and real argument x

FORTRAN name	LOC-FNC code	Description
BJ1LAM	11-4	J_0(LAMDA), a value of the Bessel function of the first kind, with order zero and complex argument LAMDA
BJ1RNU	11-4	J_0(RNU), a value of the Bessel function of the first kind, with order zero and real argument RNU
BJ2	15-4,18-4	$J_1(x)$, a value of the Bessel function of the first kind, with order one and real argument x
BJ2LAM	11-4	J_1(LAMDA), a value of the Bessel function of the first kind, with order one and complex argument LAMDA
BJ2RNU	11-4	J_1(RNU), a value of the Bessel function of the first kind, with order one and real argument RNU
BSSLS	4-4,5-4,11-4, 15-4,18-4,54-6	Subroutine which calculates the Bessel function of the first kind, J. This subroutine is a modification of the standard LRC library subroutine of the same name. The order used by BSSLS is less than or equal to 100; see MBES.
BTAU	19-4	Variable containing $B \cdot \tau$; see equation (61).
BTJ	19-4	Array of dimension 2 which contains $B \cdot T_j$ for $j = 1,2$; see equation (61).

FORTRAN name	LOC-FNC code	Description
BYLAMI	11-4	Array of dimension 50 which is required in calling subroutine ROCABES
BYLAMR	11-4	Array of dimension 50 which is required in calling subroutine ROCABES
BY1	15-4,18-4	$Y_o(x)$, a value of the Bessel function of the second kind, with order zero and argument x
BY1RNU	11-4	$Y_o(RNU)$, a value of the Bessel function of the second kind, with order zero and real argument RNU
BY2	15-4,18-4	$Y_1(x)$, a value of the Bessel function of the second kind, with order one and argument x
BY2RNU	11-4	$Y_1(RNU)$, a value of the Bessel function of the second kind, with order one and real argument RNU
B1, B2, B3	1-4,12-4,15-1, 18-1,19-4	Coefficients used with subroutines LIFTFN3 and NONCPT; see equations (21) and (22)
CAPA	10-4	Array of dimension 15 which contains average Glauert coefficients for component K2
CAPADIS	14-5,16-5	Contains A used in the cone model of distortion; see equation (33)

FORTRAN name	LOC-FNC code	Description
CAPF1	19-4	Variable containing $F_1(\rho)$; see equations (64) and (65)
CAPF2	19-4	Variable containing $F_2(\rho)$; see equations (66) and (67)
CAPHRHO	10-4	A variable which contains $H_{\kappa,K2}(\rho)$; see equation (41)
CAPKL	11-4	A variable containing $K_L(\nu, \lambda)$; see equation (24) where $\nu \geq 0$
CAPKMN	1-5,7-5,9-5, 10-5,12-5,14-5, 16-5,17-5,19-5	K_{mn}, variable defined in equation (2)
CAPKRHO	10-4	A variable containing $K_{\kappa,K1}(\rho)$
CAPLT	1-4,12-4, 15-3,18-3	Contains a value of the combined lift response function. If ARMISC(38) = 0, CAPLT = $L(\nu)$; see equation (21). If ARMISC(38) ≠ 0, CAPLT = $L'(\nu)$; see equation (22).
CAPNMN	1-5,7-5,9-5, 10-5,12-5,14-5, 16-5,17-5,19-5	The normalization factor for the duct radial eigenfunction, N_{mn}, which is defined by equation (7)
CAPRETA	6-4	$R_m(\mu_{mn}\eta)$, value of unnormalized eigenfunctions with argument the product of an eigenvalue times the hub-to-tip ratio

FORTRAN name	LOC-FNC code	Description
CAPRONE	6-4	$R_m(\mu_{mn})$, value of unnormalized eigenfunctions with argument an eigenvalue; see equation (6). If ISOROS = 1, CAPRONE contains average inlet stator vane drag coefficient. If ISOROS = 2, CAPRONE contains average rotor blade drag coefficient.
CD		CD is made equivalent to AV(4).
CDISINT	14-5,16-5	Common block containing CAPADIS and RHOINC
CEQUAT	3-5,4-5	Common block containing CETA, the hub-to-tip ratio, and M, the spinning mode index
CETA	3-5	Hub-to-tip ratio, η
CFACT	1-5,12-5,14-5, 7-5,16-5,17-5, 19-5	Common block containing CAPNMN, ETA, L, M, N, RMUMN, SIGN, CAPKMN
CFACTIR	1-5,7-5	Common block containing NSBIR, SIGOL, PHISBIR
CFACT2	9-5,10-5	Common block containing B, CAPKMN, CAPNMN, C3, C6, C7, C8, C9, C11, C12, C13, C14, K1, K2, L, M, N, NK2, RMUMN, SIGOL
CFUNIN4	19-5,20-5	Common block containing CTJ and TAU

FORTRAN name	LOC-FNC code	Description
CHORD	10-4,19-4	Array of dimension 3 which contains values of nondimensional chords for the three components
CI	21-4	Value of the cosine integral CI(x), where $x \geq 0$
CLIFT4	19-4	Variable containing $T^*(x,y)$, the complex conjugate of a value of the Filotas lift response function (LIFTFN4)
CMACH	1-4,9-4, 12-4,17-4	$1. - (\text{AXIALM})^2$
COEFA1	14-4	Contains a_1 used in the power model of distortion; see equation (49)
CONLIFT	10-4,11-3	If $\nu \geq 0$, CONLIFT = $K_L(\nu, \lambda)$. If $\nu < 0$, CONLIFT = $[K_L(-\nu, -\lambda^*)]^*$. See equation (28).
COSPSI	1-4,7-4	Cosine of the angle ψ, the relative exit flow angle of the blade row upstream of the sound-producing blade row; see figure 4
COSTHS	1-4,10-4,12-4, 17-4,18-1,19-4	Cosine of mean flow angle γ; see equation (11)
COTBETA	1-4,12-4	Cotangent of β, the relative stagger angle; see figure 3

FORTRAN name	LOC-FNC code	Description
COTTHS	19-4	Cotangent of the mean flow angle; see equation (11)
CTEMP1	10-4,15-4, 18-4,19-4	Variables used for temporary storage of complex numbers
CTEMP2	7-4,10-4, 19-4,21-4	
CTEMP3	21-4	
CTJ	19-5,20-5	Variable containing a value of T_j; see equation (60)
C1	1-4	If ISOROS = 1, C1 contains average inlet stator vane chord. If ISOROS = 2, C1 contains average rotor blade chord. C1 is made equivalent to AV(2).
C2	1-4,12-4, 17-4,18-1	Contains the average blade chord of the sound-generating blade row. C2 is made equivalent to AV(3).
C3,C6,C7,C8, C9,C11,C12, C13,C14	9-5,10-5	Variables in common block CFACT2 which are defined by the table on the following page.

	SIGOL ≤ -1				SIGOL ≥ 1			
ISOROS	1	2	2	1	1	2	2	1
IAERO	1	1	-1	-1	1	1	-1	-1
C3	-1	1	1	-1	1	-1	-1	1
C6	-1	1	-1	1	-1	1	-1	1
C7	-1	-1	-1	-1	1	1	1	1
C8	1	-1	-1	1	-1	1	1	-1
C9	1	-1	-1	1	1	-1	-1	1
C11	1	1	-1	-1	-1	-1	1	1
C12	-1	-1	-1	-1	1	1	1	1
C13	-1	-1	-1	-1	1	1	1	1
C14	1	-1	-1	1	1	-1	-1	1

FORTRAN name	LOC-FNC code	Description
DCL	1-4,12-4, 17-4	Average derivative of the steady-state lift coefficient with respect to the incident angle for the noise-generating blade row. DCL is made equivalent to AV(6).
DCSBL	10-4	An array of dimension 3 which contains slopes of the steady-state lift coefficients for each of the three components
DELK	19-4	Variable containing $\Delta = a_j/R$; see equation (70)
DELTALO	14-4	Contains $\delta_{L,0}$ where $\delta_{j,k}$ is the Kronecker delta
DELTAL1	14-4	Contains $(\delta_{L,1} + \delta_{L,-1})$ where $\delta_{j,k}$ is the Kronecker delta

FORTRAN name	LOC-FNC code	Description
DISINT	14-4,16-7	The complex function subprogram which evaluates the integrand $f(\phi)e^{-iL\phi}$ used in the cone model of distortion; see equation (48)
DRHO	10-4	A variable which contains $d_{\kappa,K1}(\rho)$; see equation (42)
DSPAC	1-4,7-4	If ISOROS = 1, DSPAC is the average nondimensional distance between the midchord planes of the inlet stator and the rotor, d_{ISR}. If ISOROS = 2, DSPAC is the average nondimensional distance between the midchord planes of the rotor and the outlet stator, d_{ROS}.
DVALUE	14-4	Contains the quantity $$\frac{1 - \frac{V_A}{V_1}}{A^2 - 1}$$ used in the cone model of distortion; see equation (48)
D1	4-4	$J_{M+1}(x)$, a value of the Bessel function of the first kind, with order M+1 and the real argument x
D2	4-4	$J_{M+2}(x)$, a value of the Bessel function of the first kind, with order M+2 and the real argument x

FORTRAN name	LOC-FNC code	Description
D3	4-4	$Y_{M+1}(x)$, a value of the Bessel function of the second kind, with order M+1 and the real argument x
D4	4-4	$Y_{M+2}(x)$, a value of the Bessel function of the second kind, with order M+2 and the real argument x
EGNBND	1-4,9-4,12-4, 13-1,17-4	The upper bound on eigenvalues, EGNBND = $RK/(1 - AXIALM)^2$. EGNBND = E_B of equation (3).
EGNBNDO	1-4,9-4, 12-4,17-4	Previous value (of last call to primary subroutine) of EGNBND. Initially set to -1.
EGNNORM	1-4,9-4,12-4, 17-4,6-7	Function subprogram which calculates normalization factor N_{mn} for unnormalized duct radial eigenfunction; see equation (7)
EGNVAL2	1-4,9-4,12-4, 13-6,17-4	Subroutine which computes the matrix of eigenvalues
EJ	19-4	Array of dimension 2 containing E_1 and E_2 of equation (61)
EPS	21-4	A numerical tolerance, EPS = 10^{-10}. In LIFTFN4, it is used in three different ways; see methods description of that subroutine.
EP1	3-4	Contains the value 0; used in subroutine JARRATT

FORTRAN name	LOC-FNC code	Description
EP2	3-4	Contains the value 10^{-10}; used in subroutine JARRATT
EQATION	3-4,4-7	The function subprogram used to evaluate the left side of equation (5)
ETA	1-5,3-1,4-5,5-1, 6-1,7-5,9-4,12-5, 13-1,14-5,16-5, 17-5,19-5	Hub-to-tip ratio, η
ETAO	1-4,9-4, 12-4,17-4	Previous value (of last call to primary subroutine) of ETA. It is initially set to -1.
EXPDRHO	10-4	A variable which contains $e^{-id_{\kappa,K1}(\rho)}$; see equation (42)
EXPFACT	7-4	Exponential factor in the oscillating factor of the integrand in primary subroutine AAAAA; see equation (36)
FACTAV	1-4,12-4,17-4	The factor in the integrand of equation (9) which varies slowest in the radial direction. It is averaged in each subinterval of the integration and is defined by equations (34), (35), (46), (54), (55), and (58) for the different primary subroutines.
FACTCON	1-4,9-4, 12-4,17-4	Constant factor in the integrand of equation (9). It is defined by equations (33), (45), (53), and (57).

FORTRAN name	LOC-FNC code	Description
FACTINT	1-4,7-7	Complex function subprogram which evaluates the oscillatory factor in the integrand of equation (9) for the primary subroutine AAAAA; see equation (36)
FACTIN2	9-4,10-7	Complex function subprogram which computes the oscillatory factor in the integrand of equation (9) for the primary subroutine AABAA; see equation (47)
FACTIN3	12-4,14-7	Complex function subprogram which computes the oscillatory factor in the integrand of equation (9) for the primary subroutine BCDAA; see equation (56)
FACTIN4	17-4,19-7	Complex function subprogram which computes the oscillatory factor in the integrand of equation (9) for the primary subroutine BBCAA; see equations (62), and (69)
FALPHNU	15-4	Contains the quantity $F_\alpha(\nu)$, where the function F_α is defined by equation (16) and ν is the reduced frequency
FFNU	15-4	Contains the quantity $F_f(\nu)$, where the function F_f is defined by equation (20) and ν is the reduced frequency
FJFP	4-4	A value of $J_m'(\eta \cdot x)$, the derivative of the Bessel function of the first kind of order m and argument $\eta \cdot x$, where η is the hub-to-tip ratio and x is the dummy argument of EQATION

FORTRAN name	LOC-FNC code	Description
FJP	4-4	A value of $J'_m(x)$, the derivative of the Bessel function of the first kind of order m and argument x, where x is the dummy argument of EQATION
FKMAX	19-4	Upper limit used in the truncated integral of equation (70). It is set equal to 20.
FME	1-4,12-4, 17-4	Relative exit flow Mach number of the sound-generating blade row, $M_E(\rho)$. It is made equivalent to AV(8).
FMI	1-4,12-4, 17-4	Relative inlet flow Mach number of the sound-generating blade row, $M_I(\rho)$. It is made equivalent to AV(7).
FMM	1-4,12-4	Mean flow Mach number at a radial position of a blade row, $M_M(\rho)$; see equation (10)
FMZ	1-4,12-4, 17-4,7-4	Average axial flow Mach number at the location of the sound-generating blade row, $M_Z(\rho)$
FMIE	1-4,7-4	Relative exit flow Mach number of the next blade row upstream of the sound-generating blade row, $M_{1E}(\rho)$
FNFP	4-4	A value of $Y'_m(\eta \cdot x)$, the derivative of the Bessel function of the second kind of order m and argument $\eta \cdot x$, where η is the hub-to-tip ratio and x is the dummy argument of EQATION

FORTRAN name	LOC-FNC code	Description
FNP	4-4	A value of $Y_m'(x)$, the derivative of the Bessel function of the second kind of order m and argument x, where x is the dummy argument of EQATION
FNU	18-4	$F(\nu)$, a term in the equation for the noncompact acoustic response function; see equation (19)
FRHO	19-4	Ratio of maximum blade camber to the half-chord, $f(\rho)$. It is input in AR(I,10,K).
FRTH	21-4	Variable defined by equation (A9) in reference 36
FT	3-4	The actual value of the left side of equation (5) which corresponds to a zero that was calculated by JARRATT
FUNIN4	19-4,20-7	Complex function subprogram which evaluates a function used in FACTIN4
FUNPHI	16-4	Contains $W(\rho, \phi)$, the distortion function for the cone model; see equation (48)
GAMMA	10-4	Variable which contains $\Gamma_{K2}^{o}(\rho)$; see equation (37)
GAUSS	17-4,19-4, 21-4,53-6	Subroutine which performs 4-, 8-, or 12-point Gaussian integration

FORTRAN name	LOC-FNC code	Description
GAUSS2	1-4,9-4,12-4, 17-4,57-6	Modified version of subroutine GAUSS in which the input arrays of the primary subroutine are passed to the complex function subprogram which evaluates the oscillating factor
GJ	19-4	Array of dimension 2 containing, $g_{1\ell}(\rho)$ and $g_{2\ell}(\rho)$ of equation (62)
GNRHO	10-4	Array of dimension 15 which contains $g_{1,K2}(\rho)$, $g_{2,K2}(\rho)$, ...; see equation (39)
GRTHFCN	21-4,65-7	Complex function subprogram, where: $$\mathrm{GRTHFCN} = e^{-i\alpha Z} K_o(Z)$$ with K_o a modified Bessel function, argument $Z > 0$, and α passed through common block ALPHA
GUESS	3-4	Array of dimension 3 which contains three starting values for zeros of equation (5) to be used by the iteration procedure in JARRATT
HALFPI	21-4	$\pi/2$

FORTRAN name	LOC-FNC code	Description
HANKEL	11-4	Variable which contains the ratio $$\frac{H_1^{(2)}\ (RNU)}{H_1^{(2)}\ (RNU) + i\ H_0^{(2)}\ (RNU)}$$ where $H_n^{(2)}(z) = J_n(z) - i\ Y_n(z)$ is a Hankel function of the second kind and order n
HRHO	10-4	Variable containing $h_{K2}(\rho)$; see equation (40)
HRHOI	10-4	Variable containing the imaginary part of HRHO
HRHOK	19-4	Variable containing $h(\rho,k)$; see equation (71)
HRHOR	10-4	Variable containing the real part of HRHO
H1RNU	11-4,15-4, 18-4	Variable containing $H_0^{(2)}$ (RNU) = J_0 (RNU) - i Y_0 (RNU), a Hankel function of the second kind and order zero
H2RNU	11-4,15-4, 18-4	Variable containing $H_1^{(2)}$ (RNU) = J_1 (RNU) - i Y_1 (RNU), a Hankel function of the second kind and order one
IABSL	14-4	Absolute value of L, the incidence velocity Fourier series index

FORTRAN name	LOC-FNC code	Description
IABSM	5-4,13-4	The absolute value of a spinning mode index, $\|M\|$
IAERO	9-4	Option IAERO = ARMISC(18) -1 indicates the upstream component is sound generator 1 indicates the downstream component is sound generator
IERBES	21-4	Error return from BESIK, where: 0 indicates no error 1 indicates input argument is nonpositive; no calculation is possible
IEREGNV	1-4,9-4,12-4, 13-3,17-4	Error return from EGNVAL2, where: 0 indicates successful execution 2 indicates that there are more eigenvalues required than there is space for (i.e., NDIM and/or MDIM are not large enough); as many as possible are returned 4 indicates that there are no eigenvalues IEREGNV is equivalent to IERROR.
IERJAR	3-4	Error return from JARRATT
IERLFT4	19-4,21-3	Error return from LIFTFN4, where: 0 indicates no error 1 indicates integral in FRTH did not converge according to EPS on the interval [X, 1000]

FORTRAN name	LOC-FNC code	Description
IERR	4-4,11-4,5-4, 15-4,18-4	Error return from BSSLS and BF4F. It is not used.
IERROR	1-3,9-3, 12-3,17-3	Error return from primary subroutines, where: 0 indicates successful execution 2 indicates that there are more eigenvalues required than there is space for (i.e., NDIM and/or MDIM are too small); as many eigenvalues as possible are returned 4 indicates that there are no eigenvalues IERROR is equivalent to IEREGNV.
IFLOW	1-4,9-4, 12-4,17-4	Option IFLOW = ARMISC(4) -1 indicates upstream propagation 1 indicates downstream propagation
IFORM	19-4,21-1	Option IFORM, where: 1 indicates that the exact form of the Filotas lift response function is used 2 indicates that the approximate form of the Filotas lift response function is used At the present, IFORM is set equal to 2.
IGO	11-4	If RNU $\geq$ 0, IGO = 1. If RNU < 0, IGO = 2.
ILOGIC	9-4	Internal switch which is defined by

ISOROS	1	2	2	1
IAERO	1	1	-1	-1
ILOGIC	1	2	1	2

FORTRAN name	LOC-FNC code	Description
ILOGICO	9-4	Previous value (of last call to primary subroutine) of ILOGIC. It is initially set to 0.
INDX	9-4	Variable containing the sum of ILOGIC and IAERO
INDX2	10-4	Variable containing the sum of ISOROS and IAERO
INTEGJ	19-4	Array of dimension 2 containing the integrals I_1 and I_2; see equation (70)
INTEGRL	1-4,9-4, 12-4,17-4	The value of the constant factor times the integral over the interval $(\eta,1)$. It is equal to a mode amplitude, α_{mn}; see equation (9).
IORDGS	1-4,9-4,12-4, 14-4,17-4,19-4	Option for GAUSS and GAUSS2 which, at present, is set equal to 2, where: 1 indicates 4-point Gaussian integration 2 indicates 8-point Gaussian integration 3 indicates 12-point Gaussian integration
IP	10-4,19-4	Variable set equal to -1 which is used in MTLUP
IPA	1-4,7-4,12-4, 14-4,17-4	Variable set equal to -1 which is used in MTLUP
ISIGN	4-4,11-4,5-4, 15-4,18-4	Variable set equal to -1 which is used in BF4F

FORTRAN name	LOC-FNC code	Description
ISOROS	1-4,7-4,9-4	Option ISOROS = ARMISC(5), where: 1 indicates inlet stator-rotor interaction 2 indicates rotor-outlet stator interaction
ISOROSO	1-4	Previous value (of last call to primary subroutine) of ISOROS. It is initially set equal to 0.
ITLIM	3-4	Variable set equal to 30 which is used in JARRATT
ITRACE	1-4,3-1,9-4, 12-4,13-1,17-4	Option ITRACE = ARMISC(6), where: 0 indicates no printout 1 indicates printout from primary subroutine 2 indicates printout from primary subroutines and subroutine ZEROS
JARRATT	3-4,52-6	Subroutine which calculates the zeros of equation (5)
JMAX	18-4	Upper limit of the summation in the equation for the noncompact acoustic response function; see equation (23). JMAX = MAX(RKAPA,RNU) + 1.
JMAX1	18-4	JMAX + 1
JMAX2	18-4	JMAX + 2

FORTRAN name	LOC-FNC code	Description
KI	10-4	Variable used as temporary storage for K1 and K2
KMAX	19-4	Maximum value of K: KMAX = (FKMAX/DELK) + 1. See equation (70).
K1,K2	9-5,10-5	Variables which are defined by: (see table below)
L	1-5,7-5,9-5, 10-5,12-5,13-4, 14-5,16-5, 17-5,19-5	Fourier series index of the incident velocity, ℓ
LAMDA	10-4,11-1	A variable which contains $\lambda_{\kappa,K1}$; see equation (44)
LAMDAI	11-4	A variable containing the imaginary part of LAMDA
LAMDAR	11-4	A variable containing the real part of LAMDA
LIFT	10-4,11-3	If $\nu \geq 0$, LIFT = $[K_L(\nu,\lambda)]^*$ If $\nu < 0$, LIFT = $K_L(-\nu, -\lambda^*)$ See equation (24).
LIFTFN2	10-4,11-6	The subroutine which computes the lift response function used with the primary subroutine AABAA; see reference 4

ISOROS IAERO	1 1	2 1	2 -1	1 -1
K1	2	3	2	1
K2	1	2	3	2

FORTRAN name	LOC-FNC code	Description
LIFTFN3	1-4,12-4, 15-6,19-4	The subroutine which computes the combined lift response function used with the primary subroutines AAAAA, BCDAA, and BBCAA; see reference 35
LIFTFN4	19-4,21-6	The subroutine which computes the Filotas lift response function; see reference 36
LIFT4	19-4,21-3	Variable containing T(X,Y), a value of the Filotas lift response function LIFTFN4
LUSE	13-4	A variable which is used as a counter in computing NOFM
LZERO	13-1	An option where: 0 indicates that L = 0 is acceptable 1 indicates that L = 0 is not acceptable
M	1-5,3-5,4-5, 5-1,6-1,7-5, 9-5,10-5,12-5, 13-5,14-5,16-5, 17-5,18-1,19-5, 21-5	Spinning mode index m; see RM
MAXCOEF	14-4	The number of indexes of the distortion coefficients; see MULTFCT also
MAXDIM	1-1,9-1,12-1, 17-1	A variable dimension for array AR. It must be greater than or equal to the maximum number of radial input positions + 2 for any input set.

FORTRAN name	LOC-FNC code	Description
MAXIND	10-4	A variable which contains the number of Glauert coefficients for component K2 that were input. This is equal to N+1 of equations (39) and (41).
MAXJ	1-1,9-1, 12-1,17-1	A variable dimension of array AR
MAXN	1-2S,9-2S, 12-2S,13-3, 17-2S	An array of dimension MDIM which contains for each spinning mode index a maximum radial mode index
MBES	13-4	Variable set equal to 100. It indicates the maximum order of the Bessel function which can be safely calculated by the subroutine BSSLS.
MBESSEGN	13-4	Bound or magnitude of spinning mode index M due to MBES and EGNBND
MDIM	1-1,9-1,12-1, 13-1,17-1	A variable column dimension of ALPHAMN and ARMUMN
MDIMO	1-4,9-4, 12-4,17-4	Previous value (of last call to primary subroutine) of ETA. It is initially set to 0.
MEGN	13-4	Maximum spinning mode index due to the eigenvalue bound, EGNBND
MMAX	3-4	Contains the value max $\{\lvert\min\Omega\rvert, \lvert\max\Omega\rvert\}$, where Ω = set of spinning mode indexes contained in array MUSE

FORTRAN name	LOC-FNC code	Description
MP1	3-4,4-4	Contains M + 1, where M is the spinning mode index
	13-4,5-4	Contains |M| + 1, where M is the spinning mode index
MP1MAX	3-4	Contains MMAX + 1
MP2	4-4,5-4	Contains MP1 + 1
MSAVE	3-4	Contains |MUSE(I)|, the absolute value of the I^{th} element of array MUSE
MSBE	10-4,19-4	An array of dimension 3 which contains relative exit flow Mach numbers for the three components
MSBI	10-4,19-4	An array of dimension 3 which contains relative inlet flow Mach numbers for the three components
MSBM	10-4,19-4	An array of dimension 3 which contains relative mean flow Mach numbers for the three components; see equation (10)
MSBT	1-4,9-4,12-4, 17-4,19-4	Rotor blade tip Mach number, M_T = ARMISC(7)
MSBZ	10-4,19-4	An array of dimension 3 which contains axial Mach numbers for the three components
MTLUP	1-4,7-4,56-6, 10-4,12-4,14-4, 17-4,19-4	Standard LRC library subroutine which performs multiple table lookup; see reference 42

FORTRAN name	LOC-FNC code	Description
MULTFCT	14-4	The multiplicative factor in the indexes of distortion coefficients. That is, if r = MULTFCT, then the distortion coefficients are a_r, b_r, $a_{2 \cdot r}$, $b_{2 \cdot r}$, ..., $a_{r \cdot MAXCOEF}$, $b_{r \cdot MAXCOEF}$.
MUSE	1-2S,9-2S,12-2S, 13-3,17-2S,3-1	An array of dimension MMIM which contains the set of spinning mode indexes
N	1-5,7-5,9-5, 10-5,12-5,13-4, 14-5,16-5,17-5, 19-5	N = n+1, where n is the radial mode index
NB	11-4	Variable set equal to 1. which is used in BSSLS, BF4F, and ROCABES
NBESEGN	13-4	Maximum radial mode index due to both the restriction on BSSLS (the order used in BSSLS restricted to be less than or equal to 101) and the eigenvalue bound, EGNBND
NDIM	1-1,13-1,3-1, 9-1,12-1,17-1	Variable row dimension of ALPHAMN and ARMUMN
NDIMO	1-4,9-4, 12-4,17-4	Previous value (of last call to primary subroutine) of NDIM. It is initially set to 0.

FORTRAN name	LOC-FNC code	Description
NK1	9-4	If KI = 1, NKI is the number of inlet stator vanes. If KI = 2, NKI is the number of rotor blades. If KI = 3, NKI is the number of outlet stator vanes.
NK2	9-5,10-5	(same as NK1)
NMAX	3-1,13-4	Contains min {NBESEGN, NDIM-1}
NM1	13-4	Contains N-1
NOFM	1-2S,9-2S,3-1, 12-2S,13-3,17-2S	Number of spinning mode indexes
NOFN	1-4,9-4,12-4, 13-4,17-4	Contains the maximum radial mode index corresponding to a spinning mode index
NONCPT	1-4,12-4, 18-6,19-4	Subroutine which computes the noncompact acoustic response function for the noncompact source theory; see equation (22)
NOSCE	1-4,9-4,12-4, 14-4,17-4,19-4	Number of equal subintervals used in evaluating an integral
NPTS	10-4,19-4	A variable containing the number of points in an array
NSBIR	1-5,7-5	If ISOROS = 1, NSBIR contains the number of inlet stator vanes, N_{ISV}. If ISOROS = 2, NSBIR contains the number of rotor blades, N_{RB}.

FORTRAN name	LOC-FNC code	Description
NSBNKI	9-4	If ILOGIC = 1, NSBNKI is NK1. If ILOGIC = 2, NSBNKI is NK2.
NSBRB	1-4,12-4,17-4	Number of rotor blades, N_{RB}
NSPN	1-4,7-4,12-4, 14-4,17-4	Contains the value of AR(I,J,K), which is: The number of radial positions, where the parameters of the AR array are input or 0 if only a value averaged in the radial direction is given for the $(K,J)^{th}$ parameter of the AR array
NTHZERO	3-4	A variable that indicates to APROX1 or APROX2 which zero to approximate
N1	21-4	Variable that is used to determine the first subinterval used in the integration of equation (A1) in Reference 36. Either: N1*WIDTHI < x ≤ (N1+1)*WIDTHI (N1-1)*WIDTHI < x ≤ N1*WIDTHI
PHISBIR	1-5, 7-5	If ISOROS = 1, PHISBIR contains the span-dependent phase angle at the rotor, ϕ_R. If ISOROS = 2, PHISBIR contains the span-dependent phase angle at the outlet stator, ϕ_{OS}.

FORTRAN name	LOC-FNC code	Description
PI	1-4,9-4,12-4, 14-4,15-4,17-4, 18-4,19-4,21-4	Contains the value π
PJ	19-4	Array of dimension 2 which contains P_j, $j = 1,2$; see equation (61)
PSI	19-4	Variable containing $\psi(\rho, k)$, the gust yaw angle used in Filotas lift response function; see equation (72)
Q	14-4	Contains the exponent q used in the power model of distortion; see equation (49). q = ARMISC(23) if ARMISC(22) = 2.
RATIO	6-4	Contains the ratio, m^2/μ_{mn}^2, of the square of a spinning mode m over the square of an eigenvalue μ_{mn}
RHO	7-1,10-1,14-1, 18-1,19-1	Nondimensional duct radial coordinate, ρ
RHOINC	14-5,16-5	Same as RHO
RK	1-4,9-4,12-4, 17-4	Nondimensional frequency, ω
RKAPA	18-4	$\kappa_{mn\sigma}^{\pm} = \frac{c}{2}\left[K_{mn}^{\pm} e_{\phi} - \frac{m}{\rho} e_z\right]$
RKSQD	1-4,9-4,12-4, 17-4	ω^2, the square of the nondimensional frequency

FORTRAN name	LOC-FNC code	Description
RLOW	1-4,9-4,12-4, 14-4,17-4,19-4, 21-4	Lower bound of a subinterval used in an integration
RM	3-4	Spinning mode index in floating point; see M
RMUMN	1-5,5-1,6-1,7-5, 9-5,10-5,12-5, 14-5,16-5,17-5, 19-5	An eigenvalue, μ_{mn}
RNOFSV	1-4,12-4, 13-1,17-4	Number of stator vanes. It is used in calling subroutine EGNVAL2. With primary subroutine AAAAA: If ISOROS = 1, RNOFSV = number of inlet stator vanes. If ISOROS = 2, RNOFSV = number of outlet stator vanes. If primary subroutines BBCAA or BCDAA are used, RNOFSV = 1.
RNOFSVO	1-4,12-4, 17-4	Previous value (of last call to primary subroutine) of RNOFSV. It is initially set to 0.
RNU	1-4,11-1,12-4, 15-1,18-1,21-1	Reduced frequency, ν; see equation (32)
RNUKAPA	10-4	Reduced frequency, $\nu_{\kappa,Kl}(\rho)$; see equation (43)

FORTRAN name	LOC-FNC code	Description
ROCABES	10-4,11-4,21-4, 58-6	Subroutine which computes $J_n(z)$ and $Y_n(z)$, the Bessel functions of the first and the second kind for integer order n and complex argument z
RSBNKI	9-4	If ILOGIC = 1, RSBNKI is NK2. If ILOGIC = 2, RSBNKI is NK1.
RSBNKIO	9-4	Previous value (of last call to primary subroutine) of RSBNKI. It is initially set equal to 0.
RUP	1-4,9-4,12-4, 14-4,17-4,19-4, 21-4	Upper bound of a subinterval used in evaluating an integral
S	5-1	Dummy argument of UNEGNFN, where 0 < S. UNEGNFN calculates R_m (μ_{mn}*S), the unnormalized duct radial eignfunction with argument the product of an eigenvalue times S.
SAVELAM	11-4	A variable used to temporarily save LAMDA
SAVERNU	11-4	A variable used to temporarily save RNU
SC	3-1,13-4	Array of dimension 40 that is used as a scratch array in ZEROS
SCRATCH	4-5,5-5,10-5, 11-5,15-5,18-5, 19-5,21-5	Common block name which contains the array BES, an array of dimension 1000 that is used as a scratch array in Bessel function subroutines

FORTRAN name	LOC-FNC code	Description
SCPTRMN	7-4,10-4,14-4, 19-4	A value of the normalized duct radial eigenfunction, $R_m(\mu_{mn}\rho)$
SI	21-4	Value of sine integral $Si(x) = \int_0^x \frac{\sin \tau}{\tau} d\tau$ where x is real
SICI	21-4,64-6	Subroutine which computes the sine and cosine integrals
SIGMA	1-4,9-4,12-4, 17-4	Harmonic index, σ
SIGN	1-5,7-5,12-5, 14-5,16-5,19-5	If ISOROS = 1, SIGN contains -1. If ISOROS = 2, SIGN contains 1.
	18-4	$(-1)^j$ of equation (23)
SIGNKI	9-4	A variable which contains the product of SIGMA and NSBNKI
SIGNKIO	9-4	Previous value (of last call to primary subroutine) of SIGNKI. It is initially set to 0.
SIGNRB	1-4,12-4, 13-1,17-4	$\sigma * N_{RB}$, the product of the harmonic index times the number of rotor blades
SIGNRBO	1-4,12-4, 17-4	Previous value (of last call to primary subroutine) of SIGNRB; initially set to 0.

FORTRAN name	LOC-FNC code	Description
SIGOL	1-5,7-5, 9-5,10-5	With primary subroutine AAAAA: If ISOROS = 1, SIGOL = L. If ISOROS = 2, SIGOL = SIGMA. With primary subroutine AABAA: If ILOGIC = 2, SIGOL = SIGMA. If ILOGIC $\neq$ 2, SIGOL = L.
SINBETA	1-4,12-4	Sine of β , the relative stagger angle; see figure 3
SINPSI	7-4	Sine of ψ, the relative exit flow angle of the blade row upstream of the sound-generating blade row; see figure 4
SINTHS	1-4,10-4,12-4, 17-4,18-4,19-4	Sine of the mean relative flow angle, γ (ρ): see equation (11)
SNU	15-4,18-4	Contains the quantity S (ν), where S is the Sears lift response function and ν is the reduced frequency. See equation (15).
SQRT2PI	19-4	$\sqrt{2\pi}$
SUM	18-4	The sum that appears in the equation for the noncompact acoustic response function: $\mathrm{Sum} = \frac{2}{\nu} \sum_{J=1}^{J_{max}} (-1)^j J_j (\nu_\ell)\left[J_{j+1} (\kappa^{\pm}_{mn\sigma}) + J_{j-1} (\kappa^{\pm}_{mn\sigma})\right]$ See equation (23).

FORTRAN name	LOC-FNC code	Description
TAU	19-5,20-5	Time delay resulting from the distance between the axial position of the eddy center at the temporal origin and the rotor plane
TEMP1	1-4,7-4,10-4, 12-4,15-4,17-4, 18-4,19-4,20-4, 21-4	Variables which are used for temporary storage in calculations
TEMP2	1-4,10-4,12-4, 17-4,18-4,19-4, 20-4,21-4	
TEMP3	1-4,10-4,12-4, 18-4,19-4,21-4	
TEMP4	1-4,10-4,18-4, 21-4	
TERM	10-4	Array of dimension 15 which contains the terms of a summation
THETA	10-4	Array of dimension 3 which contains the stagger angles (or mean flow angles) in radians for each of the three blade rows; see equation (11)
		or
	21-4	Gust yaw angle, ψ, used in the Filotas lift response function; see equation (72)

FORTRAN name	LOC-FNC code	Description
TJ	19-4	Array of dimension 2 which contains T_j, $j = 1,2$, the temporal length of an eddy; see equation (60)
TWOPI	1-4,9-4,12-4, 17-4,19-4	2π
T1	5-4	Contains a value of $Y_m'(\mu_{mn} * \eta)$, the derivative of the Bessel function of the second kind of order m and argument the product of an eigenvalue times the hub-to-tip ratio
T2	5-4	Contains a value of $J_m(\mu_{mn} * S)$, the Bessel function of the first kind of order m and argument the product of an eigenvalue, μ_{mn}, times the dummy argument, S, of subroutine UNEGNFN
T3	5-4	Contains a value of $J_m'(\mu_{mn} * \eta)$, the derivative of the Bessel function of the first kind of order m and argument the product of an eigenvalue times the hub-to-tip ratio
T4	5-4	Contains a value of $Y_m(\mu_{mn} * S)$, the Bessel function of the second kind of order m and argument the product of an eigenvalue, μ_{mn}, times the dummy argument, S, of subroutine UNEGNFN

FORTRAN name	LOC-FNC code	Description
UNEGNFN	5-7,6-4,7-4, 10-4,14-4 19-4	Function subprogram which computes the unnormalized duct radial eigenfunction, $R_m\ (\mu_{mn}\rho)$
VADBVI	14-4	Contains V_A/V_1 which is used in the cone model of the distortion; see equation (48)
VALUINT	1-4,9-4,12-4; 17-4	The value of the integral of the oscillatory factor over a subinterval. It is calculated by subroutine GAUSS2.
	14-4,19-4	The value of an integral in the oscillatory factor of BCDAA or BBCAA. It is calculated by subroutine GAUSS. See equation (48) for BCDAA and equation (61) for BBCAA.
WIDTHI	1-4,9-4,12-4, 14-4,17-4, 19-4,21-4	Width of a subinterval used in evaluating an integral. Every subinterval has the same width, WIDTHI.
WSBLRHO	14-4	The L^{th} complex distortion coefficient at a duct radial coordinate. For the cone model, see equation (48). For the power model, see equation (49). For the case where the coefficients are input, see equation (50).
X	1-4	Distance along mean streamline traveled by wake
		or
	4-1	Dummy argument of function subprogram, EQATION

FORTRAN name	LOC-FNC code	Description
Y	4-4	Contains $x \cdot \eta$, the product of the dummy argument, x, of EQATION times the hub-to-tip ratio, η
YIM	10-4	Array of dimension 20 which is required in calling subroutine ROCABES
YRE	10-4	Array of dimension 20 which is required in calling subroutine ROCABES
ZERO	3-4	Contains a zero of equation (5)
ZEROS	3-6,13-4	Subroutine which calculates zeros of equation (4)
ZSBIR	1-4,9-4, 12-4,17-4	In primary subroutine AAAAA: If ISOROS = 1, ZSBIR = axial position of the rotor, ARMISC(17). If ISOROS = 2, ZSBIR = axial position of the outlet stator, ARMISC(16). In primary subroutine AABAA: If INDX = 1, ZSBIR = axial position of the inlet stator, ARMISC(15). If INDX = 3, ZSBIR = axial position of the outlet stator, ARMISC(16). Otherwise, ZSBIR = axial position of the rotor, ARMISC(17). In primary subroutines BBCAA and BCDAA: ZSBIR = axial position of the rotor, ARMISC(17).

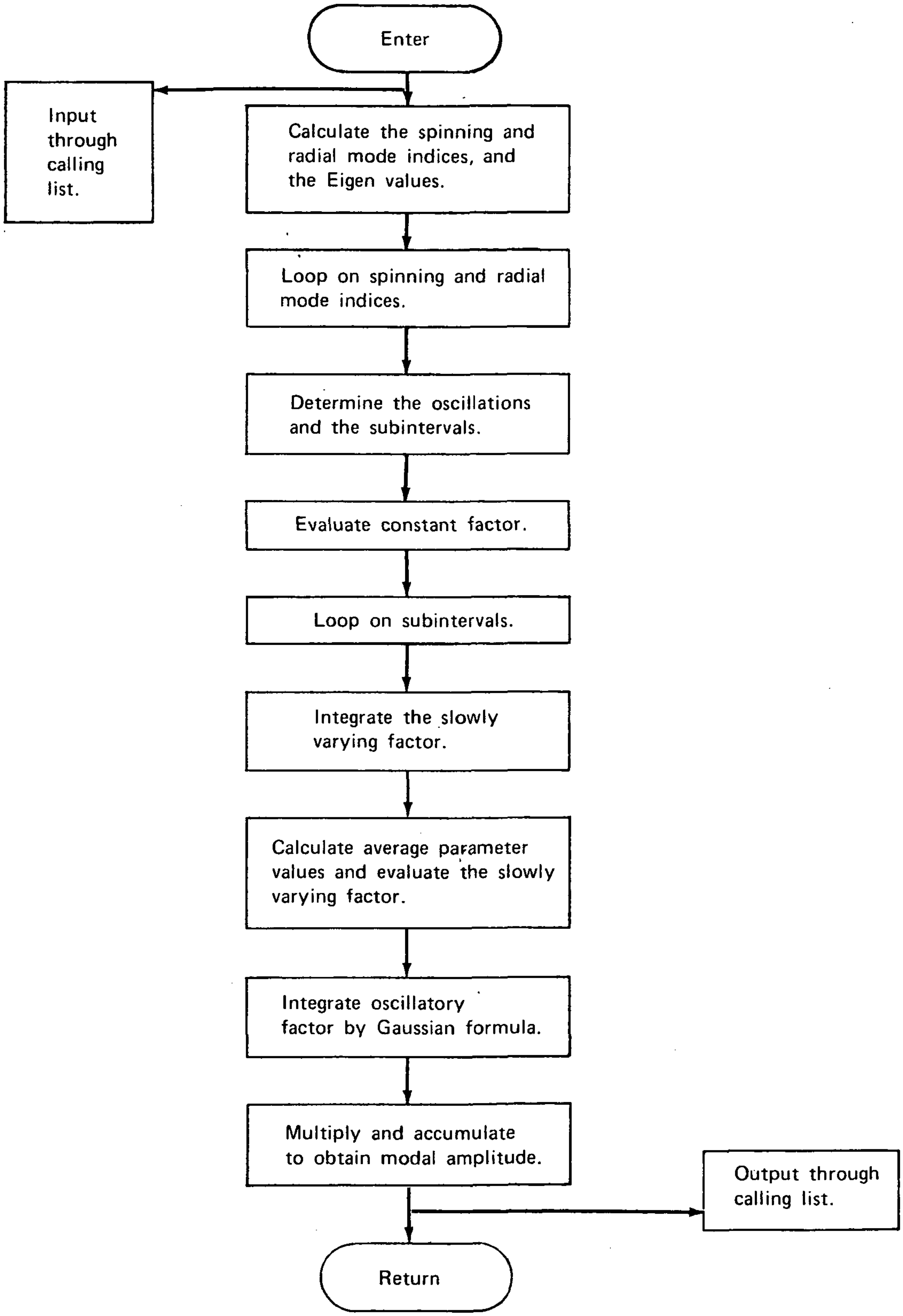
3.0 SUBPROGRAM DOCUMENTATION

As previously discussed, each subroutine package consists of a primary subroutine and a set of secondary subprograms. Each primary subroutine computes mode amplitudes according to the expression:

$$A^{\pm}_{mn\sigma} = \left\{\begin{matrix}\text{CONSTANT}\\ \text{FACTOR}\end{matrix}\right\} \sum_{j=1}^{N_{SUB}} \left\{\begin{matrix}\text{AVERAGE OF}\\ \text{NON-OSCILLATORY}\\ \text{FACTOR}\end{matrix}\right\}_j \int_{a_j}^{b_j} \left\{\begin{matrix}\text{OSCILLATORY}\\ \text{FACTOR}\end{matrix}\right\} d\rho$$

The logical flow of the primary subroutines is shown on the next page.

The remainder of this section consists of descriptions of the primary subroutines, secondary special-purpose and secondary general-purpose subprograms. Each subprogram is documented according to the format: a title and statement of purpose, a step-by-step statement of the algorithm, a flow chart, and a computer listing.



Enter
Input through calling list.
Calculate the spinning and radial mode indices, and the Eigen values.
Loop on spinning and radial mode indices.
Determine the oscillations and the subintervals.
Evaluate constant factor.
Loop on subintervals.
Integrate the slowly varying factor.
Calculate average parameter values and evaluate the slowly varying factor.
Integrate oscillatory factor by Gaussian formula.
Multiply and accumulate to obtain modal amplitude.
Output through calling list.
Return

3.1 Primary Subroutine Descriptions

3.1.1 Subroutine AAAAA

Purpose: This subroutine computes the mode amplitudes for a given harmonic from two acoustic sources--rotor blades cutting through viscous wakes from the inlet stator vanes, and the rotor blade viscous wakes washing over the outlet stator vanes. The computation essentially consists of determining which modes contribute significantly to the sum, computing the required modal parameters, and evaluating a definite integral per mode. This integral is equation (9) from appendix I of volume I and is expressed as follows for numerical evaluation:

$$A^{\pm}_{mn\sigma} = \begin{Bmatrix}\text{CONSTANT}\\\text{FACTOR}\end{Bmatrix} \sum_{j=1}^{N_{SUB}} \begin{Bmatrix}\text{AVERAGE OF}\\\text{NON-OSCILLATORY}\\\text{FACTOR}\end{Bmatrix}_j \int_{a_j}^{b_j} \begin{Bmatrix}\text{OSCILLATORY}\\\text{FACTOR}\end{Bmatrix} d\rho$$

$$\begin{Bmatrix}\text{CONSTANT}\\\text{FACTOR}\end{Bmatrix} = \frac{-1}{\beta_{mn\sigma}} \frac{N_1 N_2}{8} e^{-iK^{\pm}_{mn\sigma} Z_2}$$

$$\begin{Bmatrix}\text{NON-OSCILLATORY}\\\text{FACTOR}\end{Bmatrix}_j = C_1 C_2 \left(\frac{dC_L}{d\alpha}\right)_2 M_M M_E \left(\frac{\text{SIN}\beta}{\rho\,\text{COS}\psi}\right) \text{CAPLT}$$

$$* \Lambda_\circ\Lambda \left\{\frac{me_\phi}{\rho} + K^{\pm}_{mn\sigma} e_z\right\}$$

$$\begin{Bmatrix}\text{OSCILLATORY}\\\text{FACTOR}\end{Bmatrix} = e^{iqN_1\Theta_2} \mathcal{R}_m\left(\mu_{mn}\rho\right) e^{iq\frac{d\,\text{SIN}\,\psi}{\rho\,\text{COS}\,\psi}}$$

See the FORTRAN dictionary (sec. 2.2) for CAPLT.

Method: The procedure is as follows:

1) Set the phase angle, ϕ_{IR}, occurring in the oscillatory factor.

2) Obtain the eigenvalue generation parameters (the input to EGNVAL2).

3) Compare these parameters to stored values to determine if the required eigenvalues are already available. If values are equal, proceed to step 6.

4) Compute the mode indexes and the corresponding eigenvalues.

5) Error return if correct eigenvalues have not been computed.

6) Loop on the spinning mode index.

7) Set values of required integers.

8) Loop on the radial mode index.

9) Compute the propagation constants and the normalization of the duct radial eigenfunction.

10) Compute the constant factor in the mode amplitude expression.

11) Initialize the value of the integral to zero.

12) Compute the number of equal subintervals required, which is determined by the total number of zeros of the oscillatory factor on the full integration interval.

13) Loop on subintervals.

14) Compute the lower and upper bound and the midpoint of the subinterval.

15) Set up for accessing the input geometric and aerodynamic data.

16) If the average value over the full interval of a geometric or aerodynamic variable is input, use it and proceed to step 18.

17) Compute an average value on the subinterval for the geometric or aerodynamic variable.

18) Initialize the nonoscillatory factor to the product of the average value of the first three variables appearing in that factor.

19) Compute flow angles and multiply the average value of the next three variables in the nonoscillatory factor into that factor.

20) Compute the reduced frequency and the lift function coefficients (used for noncompact factor also).

21) When the compact option is specified, compute the value for the frequency response function of the lift and multiply this into the nonoscillatory factor.

22) When the noncompact option is specified, compute the noncompact factor and multiply this into the nonoscillatory factor.

23) Compute the inner product, or projection, factor and multiply into the nonoscillatory factor.

24) Compute the relative streamwise distance traveled by the wake, which is used to compute a wake Fourier coefficient, and multiply this into the nonoscillatory factor.

25) Integrate the oscillatory factor over the subinterval.

26) Multiply the nonoscillatory and the integrated oscillatory factors together and accumulate in the integral value, completing the loop on the subintervals.

27) Multiply the constant factor into the integral value giving the mode amplitude for the current spinning mode index and radial mode index.

28) Save the current eigenvalue generation parameters from step 2. The eigenvalues will not have to be recomputed in the next execution if these parameters remain unchanged.

Usage: CALLING SEQUENCE

```
      DIMENSION MUSE(MDIM),MAXN(MDIM),ARMUMN(NDIM,MDIM),
     *    ARMISC(40),AR(MAXDIM,MAXJ,3)
      COMPLEX ALPHAMN(NDIM,MDIM)
          .
          .
          .
      CALL AAAAA(ARMISC,MAXDIM,MAXJ,AR,MDIM,NDIM,ARMUMN,
     *    NOFM,MUSE,MAXN,ALPHAMN,IERROR)
```

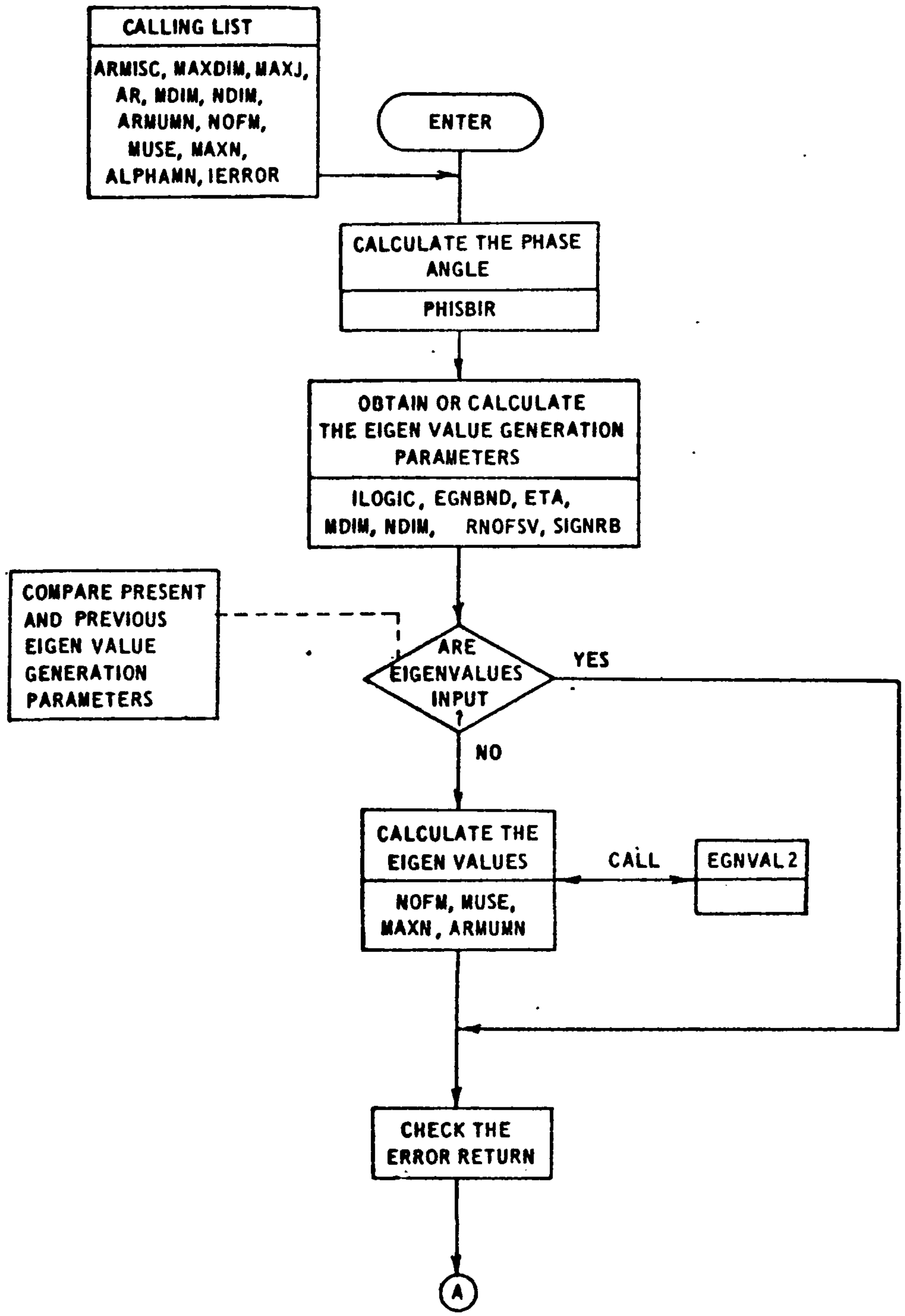
Restrictions: The use and restrictions on the input arrays ARMISC and AR and the input/output NOFM,MUSE,MAXN,ARMUMN are given in section 2.2.

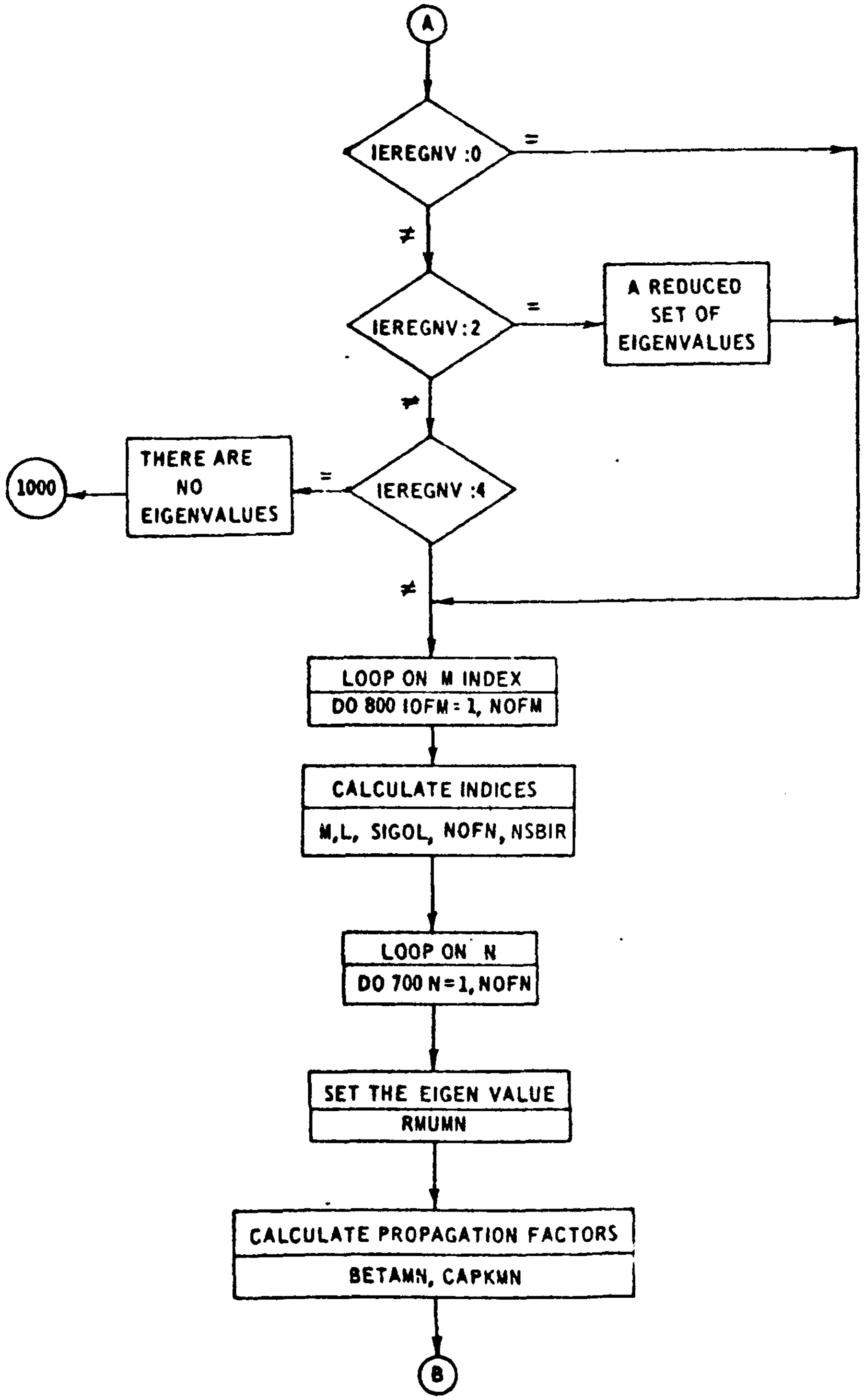
The maximum spinning mode is limited (see subroutine EGNVAL2) in absolute value to 100, and the maximum radial mode index as a result is at most 40.

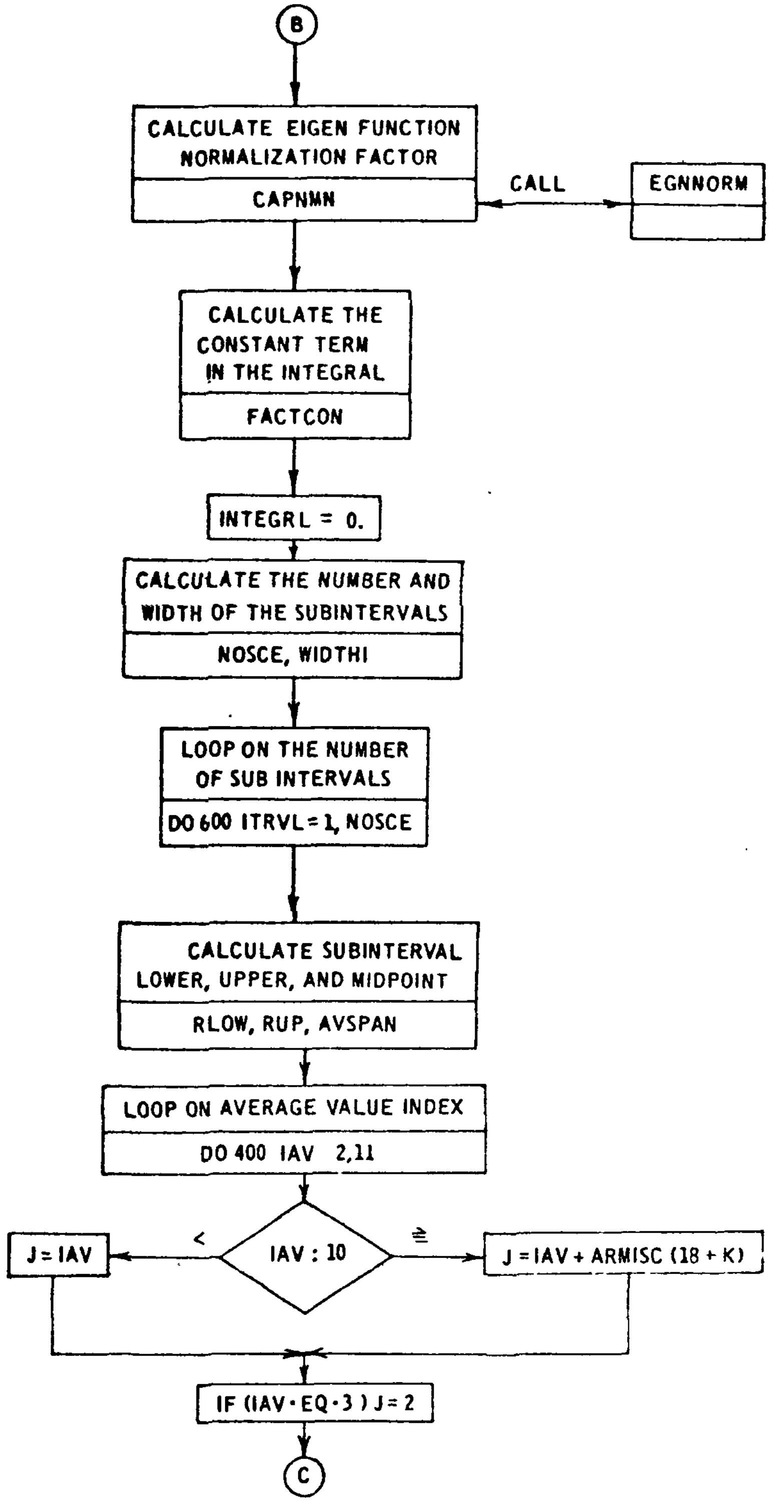
Error Return: IERROR (see the FORTRAN dictionary, sec. 2.2)

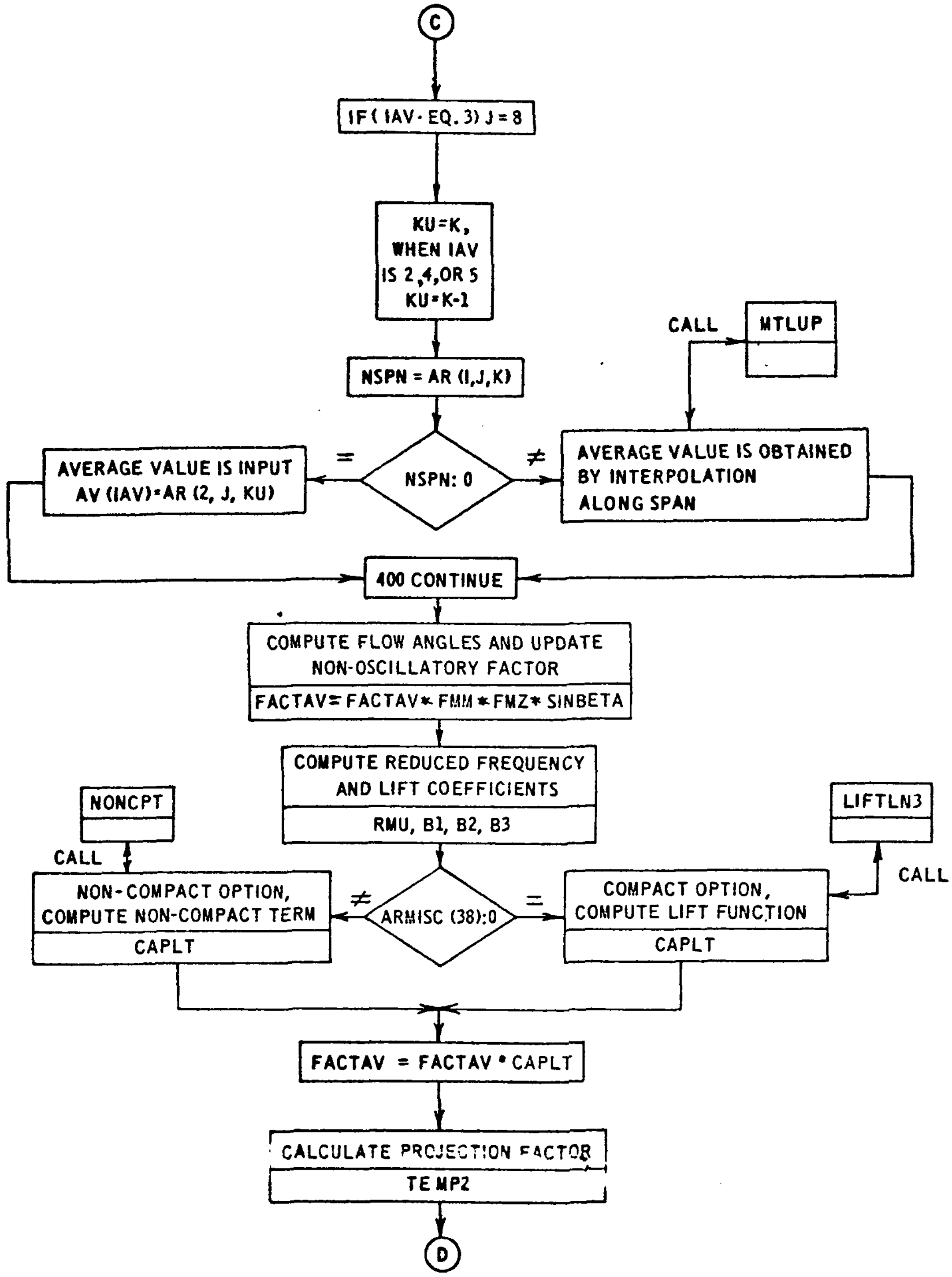
Printout and Diagnostics: See the definition of ARMISC(6),ITRACE, in the dictionary.

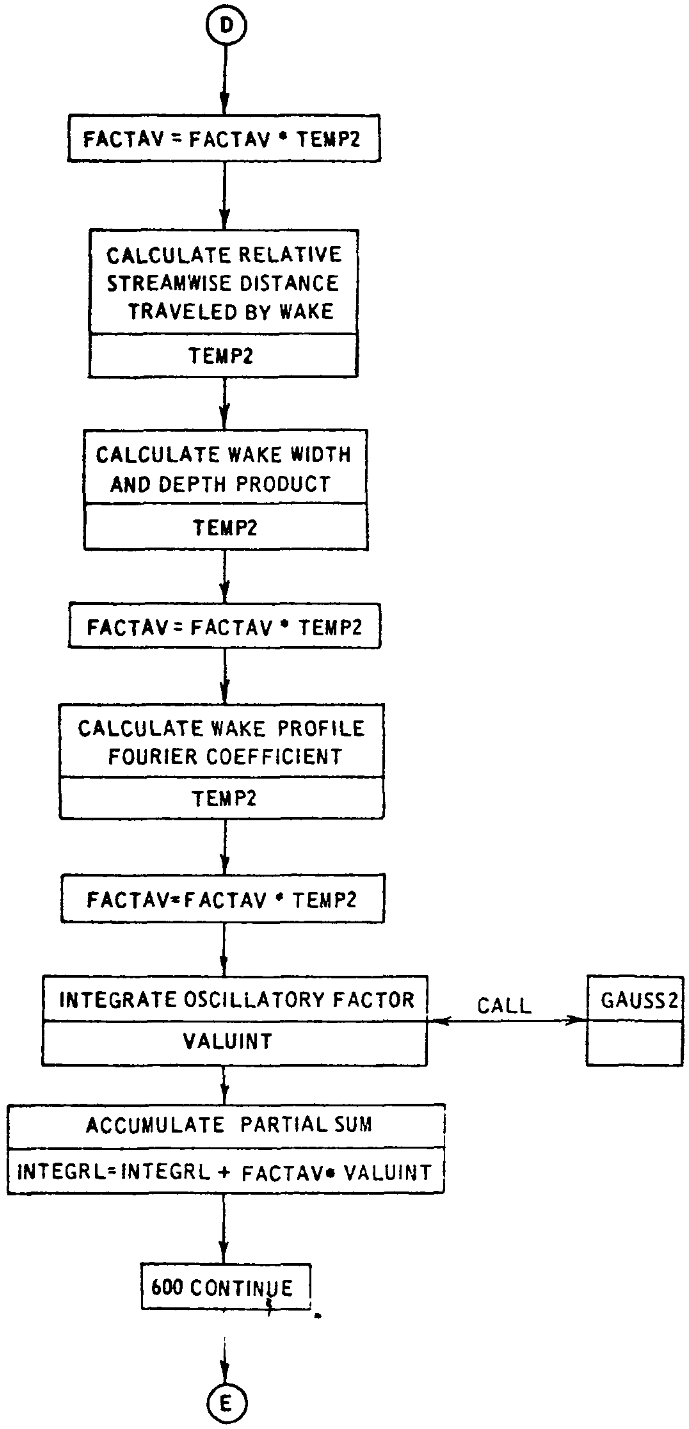
Timing: Of the cases run, the average time was 57 seconds per case.

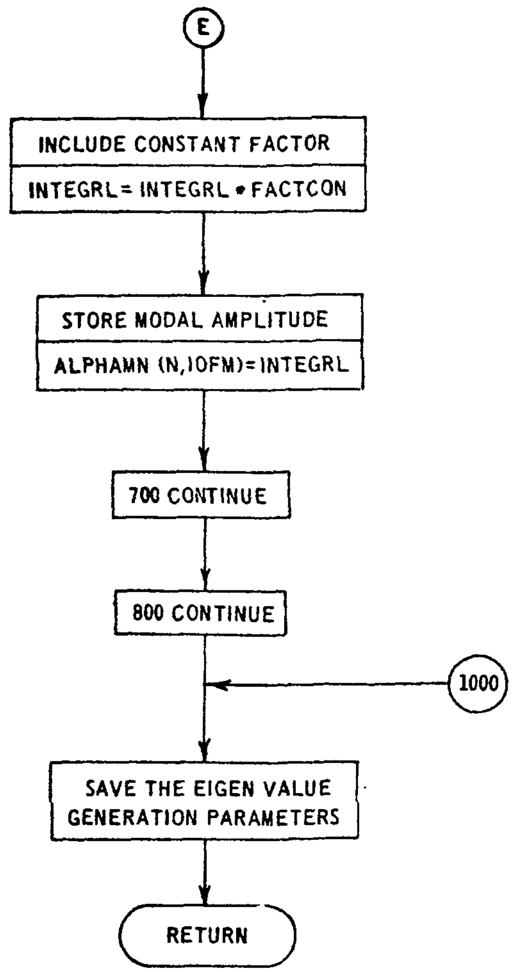

CALLING LIST
ARMISC, MAXDIM, MAXJ,
AR, MDIM, NDIM,
ARMUMN, NOFM,
MUSE, MAXN,
ALPHAMN, IERROR
ENTER
CALCULATE THE PHASE
ANGLE
PHISBIR
OBTAIN OR CALCULATE
THE EIGEN VALUE GENERATION
PARAMETERS
ILOGIC, EGNBND, ETA,
MDIM, NDIM, RNOFSV, SIGNRB
COMPARE PRESENT
AND PREVIOUS
EIGEN VALUE
GENERATION
PARAMETERS
ARE
EIGENVALUES
INPUT
?
YES
NO
CALCULATE THE
EIGEN VALUES
NOFM, MUSE,
MAXN, ARMUMN
CALL
EGNVAL 2
CHECK THE
ERROR RETURN
A


A
IEREGNV :0
=
≠
IEREGNV:2
=
A REDUCED SET OF EIGENVALUES
≠
IEREGNV :4
=
THERE ARE NO EIGENVALUES
1000
≠
LOOP ON M INDEX
DO 800 IOFM = 1, NOFM
CALCULATE INDICES
M,L, SIGOL, NOFN, NSBIR
LOOP ON N
DO 700 N=1,NOFN
SET THE EIGEN VALUE
RMUMN
CALCULATE PROPAGATION FACTORS
BETAMN, CAPKMN
B


B
CALCULATE EIGEN FUNCTION NORMALIZATION FACTOR
CAPNMN
CALL
EGNNORM
CALCULATE THE CONSTANT TERM IN THE INTEGRAL
FACTCON
INTEGRL = 0.
CALCULATE THE NUMBER AND WIDTH OF THE SUBINTERVALS
NOSCE, WIDTHI
LOOP ON THE NUMBER OF SUB INTERVALS
DO 600 ITRVL = 1, NOSCE
CALCULATE SUBINTERVAL LOWER, UPPER, AND MIDPOINT
RLOW, RUP, AVSPAN
LOOP ON AVERAGE VALUE INDEX
DO 400 IAV 2,11
IAV : 10
J = IAV
J = IAV + ARMISC (18 + K)
IF (IAV·EQ·3) J = 2
C


C
IF (IAV · EQ. 3) J = 8
KU=K,
WHEN IAV
IS 2,4,OR 5
KU=K-1
NSPN = AR (I,J,K)
CALL
MTLUP
AVERAGE VALUE IS INPUT
AV (IAV)=AR (2, J, KU)
=
NSPN: 0
≠
AVERAGE VALUE IS OBTAINED
BY INTERPOLATION
ALONG SPAN
400 CONTINUE
COMPUTE FLOW ANGLES AND UPDATE
NON-OSCILLATORY FACTOR
FACTAV= FACTAV * FMM * FMZ * SINBETA
COMPUTE REDUCED FREQUENCY
AND LIFT COEFFICIENTS
RMU, B1, B2, B3
NONCPT
CALL
LIFTLN3
CALL
NON-COMPACT OPTION,
COMPUTE NON-COMPACT TERM
CAPLT
≠
ARMISC (38):0
=
COMPACT OPTION,
COMPUTE LIFT FUNCTION
CAPLT
FACTAV = FACTAV * CAPLT
CALCULATE PROJECTION FACTOR
TEMP2
D


D
FACTAV = FACTAV * TEMP2
CALCULATE RELATIVE STREAMWISE DISTANCE TRAVELED BY WAKE
TEMP2
CALCULATE WAKE WIDTH AND DEPTH PRODUCT
TEMP2
FACTAV = FACTAV * TEMP2
CALCULATE WAKE PROFILE FOURIER COEFFICIENT
TEMP2
FACTAV=FACTAV * TEMP2
INTEGRATE OSCILLATORY FACTOR
VALUINT
CALL
GAUSS2
ACCUMULATE PARTIAL SUM
INTEGRL=INTEGRL + FACTAV* VALUINT
600 CONTINUE
E


E
INCLUDE CONSTANT FACTOR
INTEGRL = INTEGRL * FACTCON
STORE MODAL AMPLITUDE
ALPHAMN (N,IOFM) = INTEGRL
700 CONTINUE
800 CONTINUE
1000
SAVE THE EIGEN VALUE
GENERATION PARAMETERS
RETURN

```
      SUBROUTINE AAAAA(ARMISC,MAXDIM,MAXJ,AR,MDIM,NDIM,ARMUMN,
     1NOFM,MUSE,MAXN,ALPHAMN,IERROR)
C
      REAL MSBT
      COMPLEX ALPHAMN(NDIM,MDIM)
      COMPLEX FACTAV,FACTCON,FACTINT,INTEGRL,CAPLT ,VALUINT
C
      DIMENSION ARMISC(1),AR(MAXDIM,MAXJ,3),ARMUMN(NDIM,MDIM),
     1MUSE(MDIM),MAXN(MDIM)
      DIMENSION AV(11)
C
      DATA ISOROSO,EGNBNDO,ETAO,MDIMO,NDIMO,RNOFSVO,SIGNRBO/
     1      0.,-1.,-1.,0,0,0.,0./
      DATA PI,TWOPI        /3.14159265358979,6.28318530717959/
C
      COMMON/CFACT/  M,N,RMUMN,CAPNMN,ETA,SIGN,L,CAPKMN
      COMMON/CFACTIR/NSBIR,SIGOL,PHISBIR
C
      EQUIVALENCE (AV(1),AVSPAN),(AV(2),C1),(AV(3),C2),(AV(4),CD),
     1(AV(6),DCL),(AV(7),FMI),(AV(8),FME),(AV(9),FMZ),(AV(5),FMIE)
C
      EXTERNAL FACTINT
C
C
C
      ISOROS = ARMISC(5)
      ITRACE = ARMISC(6)
      IF(ITRACE .GE. 1)  WRITE(6,1010)
      IF(ISOROS-1)10,10,20
   10 PHISBIR = ARMISC(13)
      SIGN = -1.
      GO TO 30
   20 PHISBIR = ARMISC(12)
      SIGN = 1.
   30 CONTINUE
C
C              GENERATE THE EIGENVALUES
C
      NSBRB  = ARMISC(10)
      SIGMA  = ARMISC(14)
      SIGNRB = SIGMA*NSBRB
      MSBT   = ARMISC(7)
      RK    = SIGNRB*MSBT
      RKSQD= RK**2
      IF( ISOROS.EQ.1) AXIALM =AR(2,9,2)
      IF( ISOROS.EQ.2) AXIALM =AR(2,9,3)
      CMACH = 1.-AXIALM**2
      EGNBND = RK/ SQRT(CMACH)
      IF(ISOROS.EQ.1) RNOFSV = ARMISC(8)
      IF(ISOROS.EQ.2) RNOFSV = ARMISC(9)
      ETA = ARMISC(3)
      IF(ITRACE .GE. 1)  WRITE(6,1020) NSBRB,SIGMA,SIGNR3,MSBT,RK,RKSQD,
     1                                 AXIALM,CMACH,EGNBND,RNOFSV,ETA
```

```
      IF( ISOROS .NE. ISOROSO )  GO TO 110
      IF( EGNBND .NE. EGNBNDO )  GO TO 110
      IF( ETA    .NE. ETAO    )  GO TO 110
      IF( MDIM   .NE. MDIMO   )  GO TO 110
      IF( NDIM   .NE. NDIMO   )  GO TO 110
      IF( RNOFSV .NE. RNOFSVO )  GO TO 110
      IF( SIGNRB .NE. SIGNRBO )  GO TO 110
      IF(ITRACE .GE. 1)  WRITE(6,1030)
      GO TO 120

      CALL EGNVAL2(ISOROS,EGNBND,ETA,MDIM,NDIM,RNOFSV,SIGNRB,ITRACE,
     1            NOFM,MUSE,MAXN,ARMUMN,IEREGNV)
      IERROR = IEREGNV
      CONTINUE

              ERROR RETURN

      IF(IEREGNV.EQ.0) GO TO 200
      IF(IEREGNV-2) 150,130,150
      IF(ITRACE.NE.0) WRITE(6,140)
      FORMAT(//1H0,70(1H*)//1H0,*A REDUCED SET OF EIGENVALUES IS AVAILAB
     1LE*/1H0,*COMPUTATIONS WILL PROCEDE*/1H0,70(1H*) )
      GO TO 200
      IF(IEREGNV-4) 200,160,200
      IF(ITRACE.NE.0) WRITE(6,180)
      FORMAT(//1H0,70(1H*)//1H0,*THERE ARE NO PROPAGATING RADIAL MODES*/
     11H0,*NO COMPUTATIONS CAN BE MADE*/1H0,(1H*) )
      GO TO 1000
      CONTINUE


      IF( ISOROS.EQ.1) ZSBIR = ARMISC(17)
      IF( ISOROS.EQ.2) ZSBIR = ARMISC(16)
      IFLOW=ARMISC(4)
       DSPAC=ARMISC(ISOROS)


              LOOP ON M

      DO 300 IOFM=1,NOFM

              SET M,L, AND NOFN

      M = MUSE(IOFM)
      L = (M-SIGNRB)/RNOFSV
      SIGOL = L
      IF(ISOROS.EQ.2) SIGOL=SIGMA
      NSBIR = NSBRB
      IF(ISOROS.EQ.1) NSBIR=ARMISC(8)
      NOFN = MAXN(IOFM)
      IF(ITRACE.GE.1.AND.IOFM.GT.1) WRITE(6,1005)
```

```
      IF(ITRACE .GE. 1)  WRITE(6,1040) M,L,SIGOL,NSBIR,NOFN
C
C             LOOP ON N
C
      DO 700 N=1,NOFN
C
C             CALCULATE PROPAGATION FACTORS
C
      RMUMN = ARMUMN(N,IOFM)
      BETAMN = SQRT(RKSQD-CMACH*RMUMN**2)
      CAPKMN = (-RK*AXIALM + IFLOW*BETAMN)/CMACH
      CAPNMN = EGNNORM(M,RMUMN,ETA)
      IF(ITRACE .GE. 1)  WRITE(6,1050) N,RMUMN,BETAMN,CAPKMN,CAPNMN
C
C
C             COMPUTE MODAL AMPLITUDES
C
C
C             CALCULATE CONSTANT FACTOR, FACTCON
C
      ARGEXP = -CAPKMN*ZSBIR
      FACTCON =- (.125*NSBRB*RNOFSV      /BETAMN)*
     1CMPLX( COS(ARGEXP), SIN(ARGEXP) )
C
C             SET THE NUMBER OF SUB-INTERVALS FOR WHICH THE
C             LAST TWO TERMS ARE EVALUATED
C
      IORDGS = 2
      INTEGRL = (0.,0.)
C
C             SET NUMBER OSCILLATIONS
C
      NOSCE  = ABS(SIGOL)*NSBIR*(PHISBIR+DSPAC)/PI
      NOSCE = MAXO(NOSCE,N)
      NOSCE = 1.5 * NOSCE
      NOSCE = MAXO(NOSCE,2)
      WIDTHI = (1.-ETA)/NOSCE
      IF(ITRACE .GE. 1)  WRITE(6,1060) FACTCON,NOSCE
C
C             LOOP ON NUMBER OF SUBINTERVALS
C
      DO 600 ITRVL=1,NOSCE
      RLOW = ETA + (ITRVL-1)*WIDTHI
      RUP  = RLOW + WIDTHI
  270 CONTINUE
C
C             EVALUATE TERM TO BE AVERAGED
C
C             SET AVERAGE SPAN
C
      AVSPAN = (RLOW + RUP )*.5
```

```
      AV(1) = AVSPAN
C
C               SET K INDEX
C
      K = 3
      IF( ISOROS .EQ.1 ) K=2
C
C               SET AVERAGE VALUES
C
      DO 400 IAV=2,11
C
C               SET J INDEX
C
      J = IAV
      IF(IAV.EQ.10) J=9+ARMISC(18+K)+1
      IF(IAV.EQ.11) J=9+ARMISC(18+K)+2
      IF( IAV.EQ.3 ) J=2
      IF(IAV.EQ.5) J=8
C
C               SET K INDEX TO BE USED
C
      KU = K
      IF( IAV.EQ.2 ) KU=K-1
      IF( IAV.EQ.4)  KU=K-1
      IF(IAV.EQ.5) KU=K-1
C
C               SET SPAN WHEN J=1
C
      NSPN = AR(1,J,KU)
C
C               AVERAGE VALUE IS INPUT
C
      IF( NSPN ) 400,330,340
  330 AV( IAV) = AR(2,J,KU)
      GO TO 400
C
C               INTERPOLATE FOR AVERAGE VALUE
C
  340 IPA=-1
      CALL MTLUP(AVSPAN,AV(IAV),1,NSPN,NSPN,1,IPA,AR(3,1,KU),AR(3,J,KU))
  400 CONTINUE
      IF(ITRACE .GE. 1)  WRITE(6,1070) RLOW,RUP,AVSPAN,C1,C2,CD,DCL,FMI,
     1                                 FME,FMZ,AV(10),AV(11),FM1E
C
C               CALCULATE THE AVERAGE FACTOR, FACTAV
C
C
      FACTAV =      C1*C2*DCL
      IF(ITRACE .GE. 1)  WRITE(6,1080) FACTAV
C
C               COMPUTE MACH NUMBER RELATED VARIABLES
C
```

```
      TEMP1 = SQRT( FMI**2 - FMZ**2 )
      TEMP2 = SQRT( FME**2 - FMZ**2 )
      TEMP3 = .25*( TEMP1 + TEMP2 )**2
      TEMP4 = SQRT( FM1E**2 - FMZ**2 )
      FMM = SQRT( FMZ**2 + TEMP3 )
      COSTHS = FMZ/FMM
      SINTHS = SQRT( 1.-COSTHS**2 )
       COSTHS=-SIGN*COSTHS
       TEMP1=SQRT(FMM**2 - FMZ**2)
      SINBETA = FMZ*( TEMP1 + TEMP4 )/( FMM*FM1E )
      COSPSI = FMZ/FM1E
      COTBETA = (FMZ-TEMP1*TEMP4/FMZ)/( TEMP1 + TEMP4 )
C
C               UPDATE AVERAGE FACTOR
C
      TEMP3 = SINBETA/(AVSPAN*COSPSI)
      FACTAV = FACTAV*FMM *FM1E*TEMP3
      IF( ITRACE.GE.1) WRITE(6,1085)TEMP1,TEMP2,TEMP3,FMM,COSTHS,
     1                                SINBETA,COSPSI,FACTAV
C
C               COMPUTE THE REDUCED FREQUENCY
C
       TEMP3=MSBT/FMM
      RNU = .5*SIGOL*NSBIR*C2*TEMP3
      IF( ISOROS.EQ.2 ) RNU = -RNU
      B1 = 1.
      B2 = -AV(11)*COTBETA
      B3 = -AV(10)*COTBETA
C
C               COMPUTE COMPACT OPTION - NAUMANN-YEH
C
      IF( ARMISC(38).NE.0.) GO TO 410
      CALL LIFTFN3(RNU,B1,B2,B3,CAPLT)
      GO TO 420
C
C               COMPUTE NON-COMPACT OPTION
C
  410 CONTINUE
      CALL NONCPT(B1,B2,B3,C2,CAPKMN,COSTHS,M,AVSPAN,RNU,SINTHS,CAPLT)
C
C               UPDATE AVERAGE FACTOR
C
  420 FACTAV = FACTAV*CAPLT
      IF( ITRACE.GE.1) WRITE(6,1090) CAPLT,FACTAV
C
      TEMP2  = M*COSTHS/AVSPAN + CAPKMN*SINTHS
      FACTAV = FACTAV*TEMP2
      IF(ITRACE .GE. 1)  WRITE(6,1100) TEMP2,FACTAV
C
      X=(DSPAC-.25*C2*FMZ/FMI)/COSPSI
      TEMP1 = 1./(X/C1 -0.2)
      TEMP2 = 1.65*CD*SQRT( TEMP1-0.15*TEMP1**2)
```

```
      FACTAV = FACTAV*TEMP2
      IF(ITRACE .GE. 1)  WRITE(6,1110) TEMP1,TEMP2,FACTAV
C
      TEMP1 = 1.36*NSBIR*C1/(TWOPI*AVSPAN    *COSPSI   )
      TEMP1 = TEMP1*SQRT(CD*(X/C1-0.35))*SIGOL
      IF( TEMP1.NE.1. ) GO TO 450
      TEMP2 = .5
      GO TO 460
  450 TEMP2 =  PI*TEMP1
      IF(TEMP2 .EQ. 0.)  TEMP2 = 1.
      IF(TEMP2 .NE. 0.)  TEMP2 = SIN(TEMP2)/TEMP2
      TEMP2 = TEMP2/(1.-(TEMP1)**2)
  460 FACTAV = FACTAV*TEMP2
      IF(ITRACE .GE. 1)  WRITE(6,1110) TEMP1,TEMP2,FACTAV
C
C
C                PERFORM GAUSSIAN INTEGRATION
C
      CALL  GAUSS2(RLOW,RUP,IORDGS,VALUINT,FACTINT,ARMISC,MAXDIM,MAXJ,AR)
C
C                ACCUMULATE THE TERMS
C
      INTEGRL = INTEGRL + FACTAV*VALUINT
C
C                END INTERVAL LOOP
C
      IF(ITRACE .GE. 1)  WRITE(6,1120) INTEGRL ,VALUINT,FACTAV
  600 CONTINUE
C
C                APPLY FIRST TERM AND STORE
C
      INTEGRL = FACTCON*INTEGRL
      ALPHAMN(N,IOFM)=INTEGRL
      IF(ITRACE .GE. 1)  WRITE(6,1130) INTEGRL
C
C                END N AND M LOOPS
C
  700 CONTINUE
  800 CONTINUE
 1000 CONTINUE
C
C                SAVE THE EIGENVALUE DETERMINING PARAMETERS
C
      ISOROSO = ISOROS
      EGNBNDO = EGNBND
      ETAO    = ETA
      MDIMO   = MDIM
      NDIMO   = NDIM
      RNOFSVO = RNOFSV
      SIGNRBO = SIGNRB
C
C                RETURN
```

```
      RETJRN
C
 1005 FOR4AT(1H1)
 1010 FORMAT(1H1,* OPTIONAL PRINTOUT FROM SUBROUTINE AAAAA*)
 1020 FORMAT(1H0,* EIGENVALUE PARAMETERS GENERATED*/1H ,2X,*NSBRB  = *,
     1I2,10X,*SIGMA = *,F3.0,9X,*SIGNRB = *,F5.0,7X,* MS3T   = *,F10.4,
     22X,*RK   = *,F10.4,2X,*RKSQD   = *,F10.4/1H ,2X,*AXIALM = *,F10.4,
     32X,*CMACH = *,F10.4,2X,*EGNBND = *,F10.5,2X,*RNJFSV = *,F10.4,2X,
     4*ETA  = *,F10.4)
 1030 FORMAT(1H0,* THE EIGENVALUES FROM PREVIOUS CALL TO SUBROUTINE *,
     1*AAAAA ARE REUSED FOR THIS CASE*)
 1040 FORMAT(1H0,* M = *,I2,3X,*L = *,I2,9X,*SIGOL = *,F6.2,3X,*NSBIR*,
     1* = *,I2,6X,*NOFN = *,I2)
 1050 FORMAT(1H0,* N = *,I2,3X,*RMUMN = *,F10.4,3X,*BETAMN = *,F10.4,3X,
     1*CAPKMN = *,F10.4,3X,*CAPNMN = *,F10.4)
 1060 FORMAT(1H0,* FACTCON = *,2F10.4,5X,*NOSCE = *,I2)
 1070 FORMAT(1H0,* RLOW = *,F9.4,3X,*RUP = *, F9.4,3X,*AVSPAN = *,F9.4,
     13X,*C1 = *,F10.4,3X,*C2 = *,F10.4,3X,*CD  = *,F10.4/1H ,* DCL = *,
     2F10.4,3X,*FMI = *, F9.4,3X,*FME = *,F9.4,6X,*FMZ = *,F9.4/
     311X,*AV(10) = *,F9.4,2X,*AV(11) = *,F9.4,2X,*FMIE = *,F9.4)
 1080 FORMAT(1H0,80X,*FACTAV = *,2E12.4 )
 1085 FORMAT(1H0,* TEMP1 = *,F9.4,3X,*TEMP2 = *,F9.4,3X,*TEMP3 = *,F9.4/
     1        1X ,* FMM   = *,F9.4,2X,*COSTHS =*,F9.4,3X,*SINBETA=*,F9.4/
     2        1X ,* COSPSI= *,F9.4,3X,*FACTAV =*,2E12.4)
 1090 FORMAT(1H0,* CAPLT = *,2E12.4,3X,*FACTAV = *,2E12.4)
 1100 FORMAT(1H0,23X,*TEMP2 = *,F10.4,39X,*FACTAV = *,2E12.4)
 1110 FORMAT(1H0,* TEMP1 = *,F10.4,3X,* TEMP2 = *,F10.4,39X,*FACTAV = *,
     12E12.4)
 1120 FORMAT(1H0, * INTEGRL = *,2E12.4,4X,*VALUINT = *,2E12.4,4X,
     1* FACTAV = *,2E12.4)
 1130 FORMAT(1H0,* INTEGRL = *,2E12.4)
      END
```

3.1.2 Subroutine AABAA

Purpose: This subroutine computes the mode amplitudes for a given harmonic. The noise is due to the nonstationary lift on the rotor or stator blades resulting from the interaction of the potential flow field of two adjacent blade rows in relative motion. Four basic interactions are possible: (1,2) interactions between the inlet guide vanes and the rotor and (3,4) interactions between the rotor and outlet guide vanes. For these cases interactions in the upstream (1,3) and downstream direction (2,4) are possible. The computation essentially consists of determining which modes contribute significantly to the sum, computing the required modal parameters, and evaluating a definite integral per mode. The integral is equation (9) from appendix I of volume I expressed for numerical evaluation:

$$A^{\pm}_{mn\sigma} = \left\{ \begin{matrix} \text{CONSTANT} \\ \text{FACTOR} \end{matrix} \right\} \sum_{j=1}^{N_{SUB}} \int_{a_j}^{b_j} \left\{ \begin{matrix} \text{OSCILLATORY} \\ \text{FACTOR} \end{matrix} \right\} d\rho$$

with

$$\left\{ \begin{matrix} \text{CONSTANT} \\ \text{FACTOR} \end{matrix} \right\} = -2\pi \frac{N_{K1}}{2\beta_{mn\sigma}} e^{-iK^{\pm}_{mn\sigma} Z_{K1}}$$

$$\left\{ \begin{matrix} \text{OSCILLATORY} \\ \text{FACTOR} \end{matrix} \right\} = M_{M,K1}(\rho)\Gamma^{o}_{K2}(\rho)a_{\kappa,K1}(\rho)H_{\kappa,K2}(\rho)\frac{\left(\frac{dC_L}{d\alpha}\right)_{K1}}{2\pi}$$

$$* \left(\frac{me_{\phi}}{\rho} + K^{\pm}_{mn\sigma}e_Z\right) * e^{-id_{\kappa,K1}(\rho)} K_{\kappa,K1}(\rho)\, \mathcal{R}_m\left(\mu_{mn}\rho\right)$$

Method: The procedure is as follows:

1) Set the parameters to K1, K2, N_{K1}, and N_{K2}

2) Obtain the eigenvalue generation parameters (the input to EGNVAL2).

3) Compare these parameters to stored values to determine if the required eigenvalues are already available. If values are equal, proceed to step 6.

4) Compute the mode indexes and the corresponding eigenvalues.

5) Error return if correct eigenvalues have not been computed.

6) Loop on the spinning mode index.

7) Set values of required integers and C's.

8) Loop on the radial mode index.

9) Compute the propagation constants and the normalization of the duct radial eigenfunction.

10) Compute the constant factor in the mode amplitude expression.

11) Initialize the value of the integral to zero.

12) Compute the number of equal subintervals required which is determined by the total number of zeros of the oscillatory factor on the full integration interval.

13) Loop on subintervals.

14) Compute the lower and upper bound and the midpoint of the subinterval.

15) Integrate the oscillatory factor over the subinterval.

16) Accumulate the integrated oscillatory factor in the integral value, completing the loop on the subintervals.

17) Multiply the constant factor into the integral value giving the mode amplitude for the current spinning mode index and radial mode index.

18) Save the current eigenvalue generation parameters from step 2. The eigenvalues will not have to be recomputed in the next execution if these parameters remain unchanged.

Usage: CALLING SEQUENCE

```
      DIMENSION MUSE(MDIM),MAXN(MDIM),ARMUMN(NDIM,MDIM)
     *    ARMISC(40),AR(MAXDIM,MAXJ,3)
      COMPLEX ALPHAMN(NDIM,MDIM)
          .
          .
          .
      CALL AABAA(ARMISC,MAXDIM,MAXJ,AR,MDIM,NDIM,ARMUMN,NOFM,
     *    MUSE,MAXN,ALPHAMN,IERROR)
```

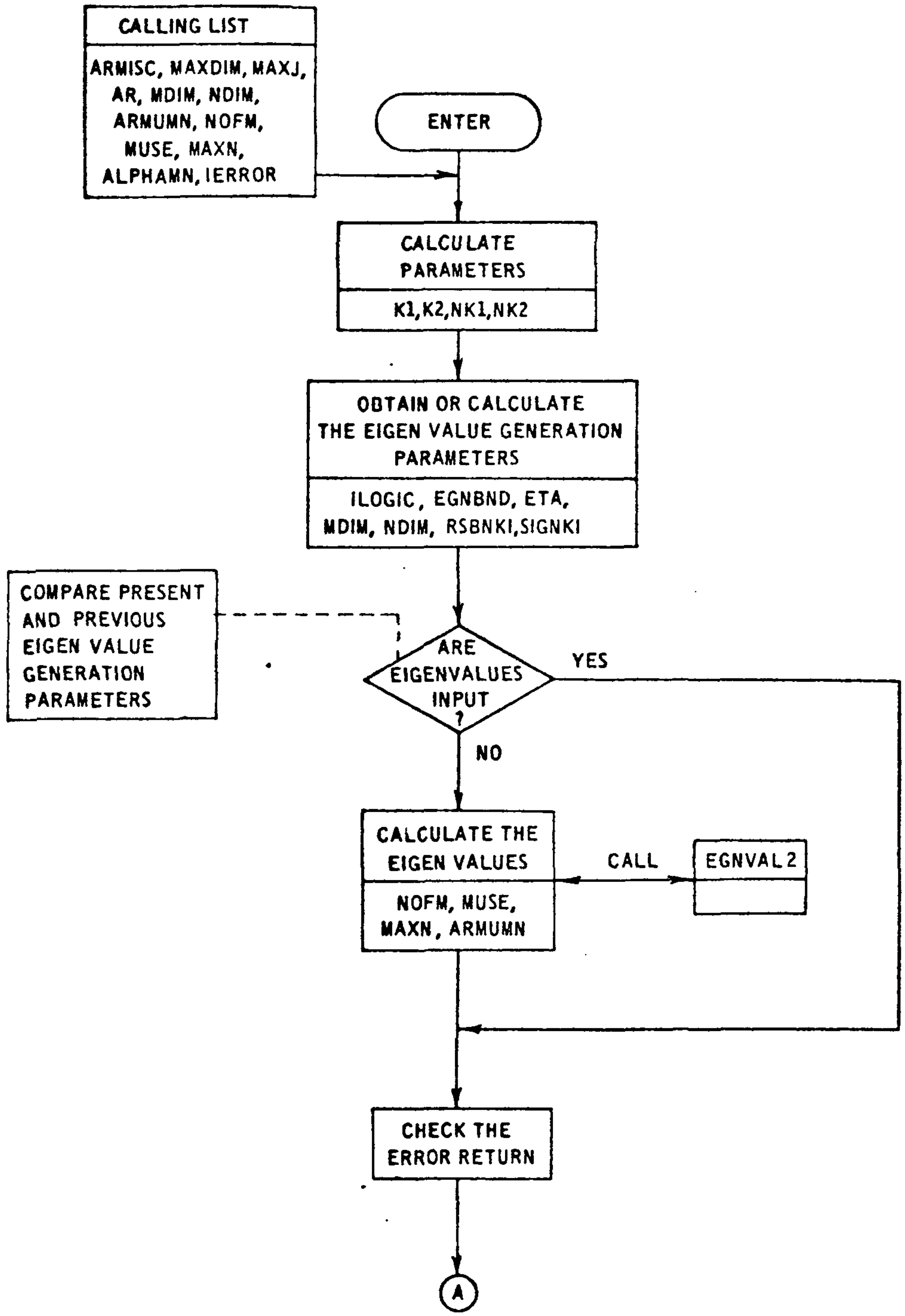
Restrictions: The use and restrictions on the input arrays ARMISC and AR and the input/output NOFM,MUSE,MAXN,ARMUMN are given in section 2.2.

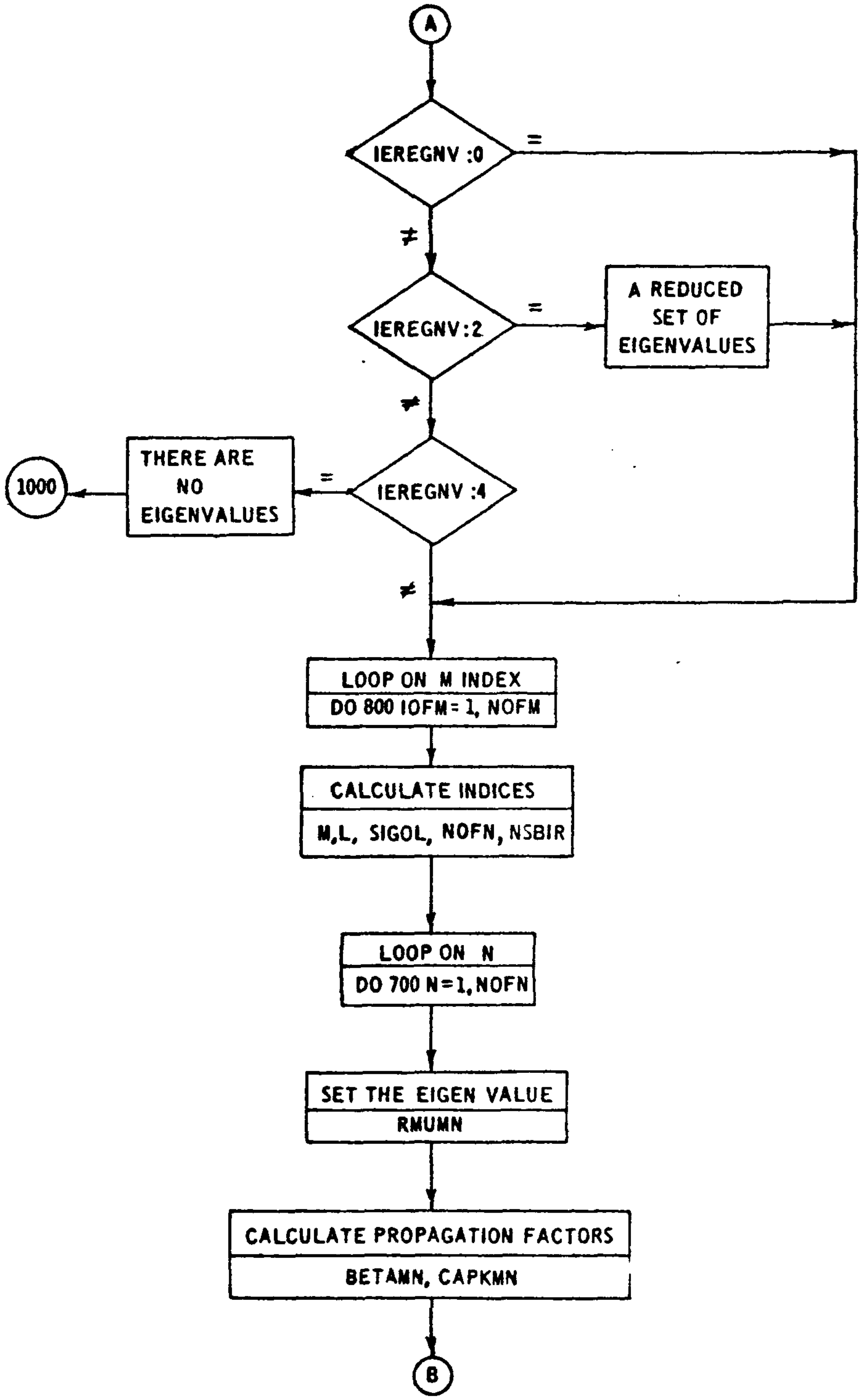
The maximum spinning mode is limited (see subroutine EGNVAL2) in absolute value to 100, and the maximum radial mode index as a result is at most 40.

Error Return: IERROR (see the FORTRAN dictionary, sec. 2.2)

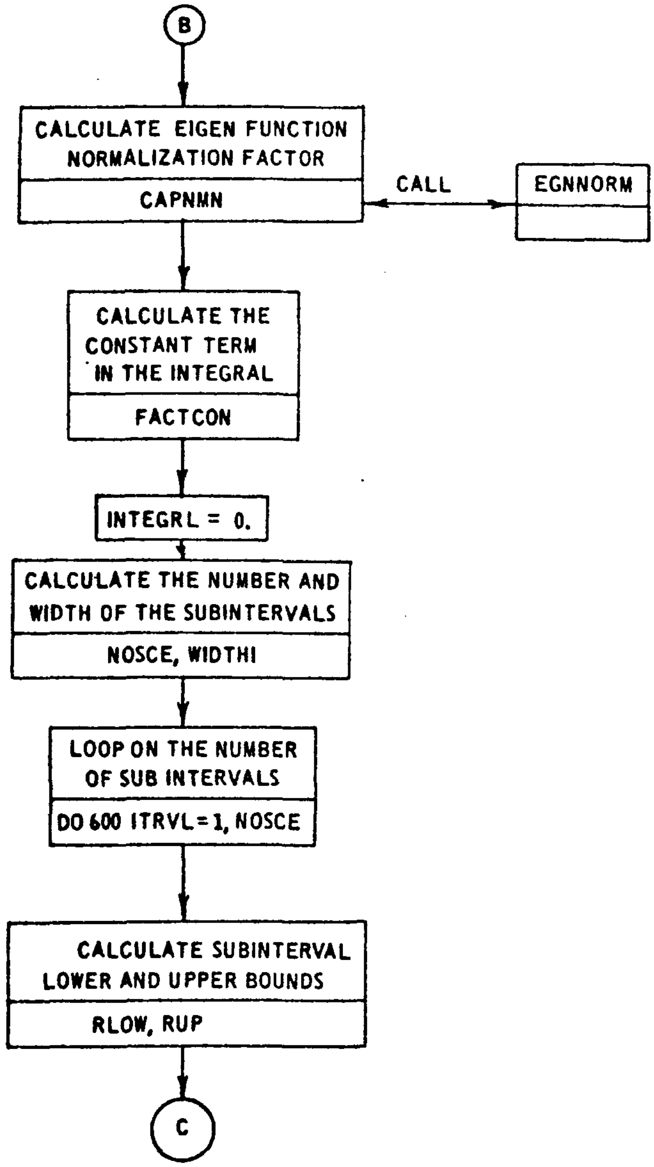
Printout and Diagnostics: See the definition of ARMISC(6),ITRACE, in the dictionary.

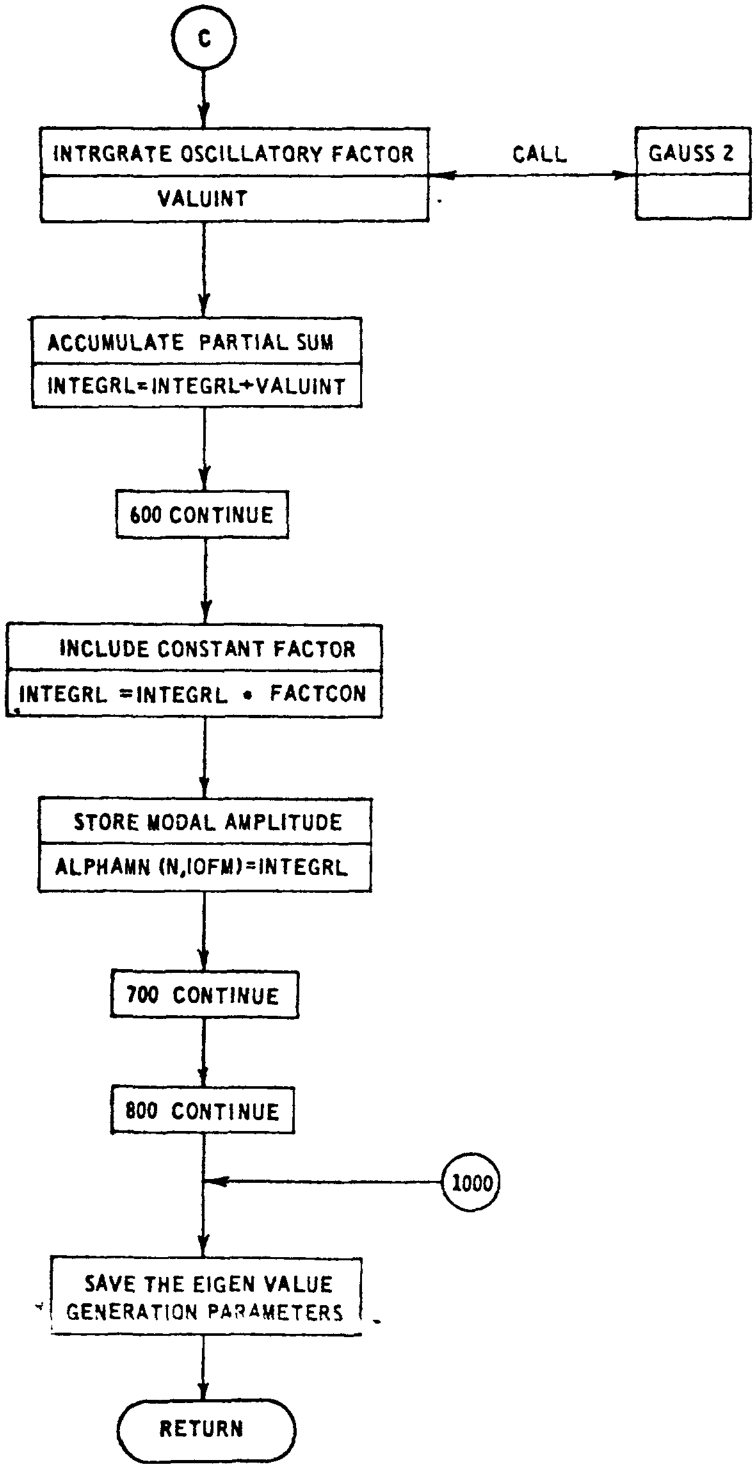
Timing: Of the cases run, the average time was 62 seconds per case.


CALLING LIST
ARMISC, MAXDIM, MAXJ, AR, MDIM, NDIM, ARMUMN, NOFM, MUSE, MAXN, ALPHAMN, IERROR
ENTER
CALCULATE PARAMETERS
K1,K2,NK1,NK2
OBTAIN OR CALCULATE THE EIGEN VALUE GENERATION PARAMETERS
ILOGIC, EGNBND, ETA, MDIM, NDIM, RSBNKI,SIGNKI
COMPARE PRESENT AND PREVIOUS EIGEN VALUE GENERATION PARAMETERS
ARE EIGENVALUES INPUT ?
YES
NO
CALCULATE THE EIGEN VALUES
NOFM, MUSE, MAXN, ARMUMN
CALL
EGNVAL 2
CHECK THE ERROR RETURN
A



A
IEREGNV :0
=
≠
IEREGNV:2
=
A REDUCED SET OF EIGENVALUES
≠
IEREGNV :4
=
THERE ARE NO EIGENVALUES
1000
≠
LOOP ON M INDEX
DO 800 IOFM= 1, NOFM
CALCULATE INDICES
M,L, SIGOL, NOFN, NSBIR
LOOP ON N
DO 700 N=1,NOFN
SET THE EIGEN VALUE
RMUMN
CALCULATE PROPAGATION FACTORS
BETAMN, CAPKMN
B


B
CALCULATE EIGEN FUNCTION NORMALIZATION FACTOR
CAPNMN
CALL
EGNNORM
CALCULATE THE CONSTANT TERM IN THE INTEGRAL
FACTCON
INTEGRL = 0.
CALCULATE THE NUMBER AND WIDTH OF THE SUBINTERVALS
NOSCE, WIDTHI
LOOP ON THE NUMBER OF SUB INTERVALS
DO 600 ITRVL=1, NOSCE
CALCULATE SUBINTERVAL LOWER AND UPPER BOUNDS
RLOW, RUP
C


C
INTRGRATE OSCILLATORY FACTOR
VALUINT
CALL
GAUSS 2
ACCUMULATE PARTIAL SUM
INTEGRL=INTEGRL+VALUINT
600 CONTINUE
INCLUDE CONSTANT FACTOR
INTEGRL =INTEGRL * FACTCON
STORE MODAL AMPLITUDE
ALPHAMN (N,IOFM)=INTEGRL
700 CONTINUE
800 CONTINUE
1000
SAVE THE EIGEN VALUE
GENERATION PARAMETERS
RETURN

```
      SUBROUTINE AABAA(ARMISC,MAXDIM,MAXJ,AR,MDIM,NDIM,ARMUMN,
     1NOFM,MUSE,MAXN,ALPHAMN,IERROR)
C
      REAL MSBT
      COMPLEX ALPHAMN(NDIM,MDIM)
      COMPLEX          FACTCON,FACTIN2,INTEGRL,          VALUINT
C
      DIMENSION ARMISC(21),AR(MAXDIM,MAXJ,3),ARMUMN(NDIM,MDIM),
     1MUSE(MDIM),MAXN(MDIM)
C
      DATA ILOGICO,EGNBNDO,ETAO,MDIMO,NDIMO,RSBNKIO,SIGNKIO/
     1     C.,-1.,-1.,0,0,0.,0./
      DATA PI,TWOPI          /3.14159265358979,6.28318530717959/
C
      COMMON/CFACT2/B,CAPKMN,CAPNMN,C3,C6,C7,C8,C9,C11,C12,C13,C14,K1,
     1               K2,L,M,N,NK2,RMUMN,SIGOL

      EXTERNAL FACTIN2


      IAERO = ARMISC(18)
      ISOROS = ARMISC(5)
      ILOGIC = ISOROS
      IF( IAERO.EQ.-1.AND.ISOROS.EQ.2) ILOGIC=1
      IF( IAERO.EQ.-1.AND.ISOROS.EQ.1) ILOGIC=2
      INDX = ILOGIC + IAERO
      ITRACE = ARMISC(6)
      IF(ITRACE .GE. 1)  WRITE(6,1010)

           COMPUTE K1, K2,      NK2, B

      K1 = ILOGIC + 1
      IF(INDX.EQ.1) K1=1
      K2 = K1-IAERO
      NK2=ARMISC(10)
      IF(INDX.EQ.2) NK2=ARMISC(8)
      IF(INDX.EQ.0) NK2=ARMISC(9)
      NK1 = ARMISC(10)
      IF(INDX.EQ.3) NK1 = ARMISC(9)
      IF(INDX.EQ.1) NK1 = ARMISC(8)
      IF(IAERO.EQ.1)B =ARMISC(K2)
      IF(IAERO.EQ.-1) B=ARMISC(K1)
      IF(ITRACE.GE.1) WRITE(6,1015) K1,K2,NK1,NK2,B
C
C          GENERATE THE EIGENVALUES
C
      IF(ILOGIC.EQ.1) NSBNKI = NK1
      IF(ILOGIC.EQ.2) NSBNKI = NK2
      SIGMA   = ARMISC(14)
      SIGNKI = SIGMA*NSBNKI
      MSBT    = ARMISC(7)
```

```
      RK    = SIGNKI*MSBT
      RKSQD= RK**2
      AXIALM = AR(2,9,K1)
      CMACH = 1.-AXIALM**2
      EGNBND = RK/ SQRT(CMACH)
      IF(ILOGIC.EQ.1) RSBNKI =  NK2
      IF(ILOGIC.EQ.2) RSBNKI =  NK1
      ETA = ARMISC(3)
      IF(ITRACE .GE. 1)  WRITE(6,1020)NSBNKI,SIGMA,SIGNKI,MSBT,RK,RKSQD,
     1                            AXIALM,CMACH,EGNBND,RSBNKI,ETA
      IF( ILOGIC .NE. ILOGICO )  GO TO 110
      IF( EGNBND .NE. EGNBNDO )  GO TO 110
      IF( ETA    .NE. ETAO    )  GO TO 110
      IF( MDIM   .NE. MDIMO   )  GO TO 110
      IF( NDIM   .NE. NDIMO   )  GO TO 110
      IF( RSBNKI .NE. RSBNKIO )  GO TO 110
      IF( SIGNKI .NE. SIGNKIO )  GO TO 110
      IF(ITRACE .GE. 1)  WRITE(6,1030)
      GO TO 120
  110 CALL EGNVAL2(ILOGIC,EGNBND,ETA,MDIM,NDIM,RSBNKI,SIGNKI,ITRACE,
     1           NOFM,MUSE,MAXN,ARMUMN,IEREGNV)
      IERROR = IEREGNV
  120 CONTINUE

               ERROR RETURN

      IF(IEREGNV.EQ.0) GO TO 200
      IF(IEREGNV-2) 150,130,150
  130 IF(ITRACE.NE.0) WRITE(6,140)
  140 FORMAT(//1H0,70(1H*)//1H0,*A REDUCED SET OF EIGENVALUES IS AVAILAB
     1LE*/1H0,*COMPUTATIONS WILL PROCEDE*/1H0,70(1H*) )
      GO TO 200
  150 IF(IEREGNV-4) 200,160,200
  160 IF(ITRACE.NE.0) WRITE(6,180)
  180 FORMAT(//1H0,70(1H*)//1H0,*THERE ARE NO PROPAGATING RADIAL MODES*/
     11H0,*NO COMPUTATIONS CAN BE MADE*/1H0,(1H*) )
      GO TO 1000
  200 CONTINUE


      ZSBIR = ARMISC(17)
      IF( INDX.EQ.3 ) ZSBIR = ARMISC(16)
      IF( INDX.EQ.1 ) ZSBIR = ARMISC(15)
      IFLOW=ARMISC(4)


               LOOP ON M

      DO 300 IOFM=1,NOFM

               SET M,L, AND NOFN
```

```
C
      M = MUSE(IOFM)
      L = (M - SIGNKI) / RSBNKI
      SIGOL = L
      IF(ILOGIC.EQ.2) SIGOL=SIGMA
      NOFN = MAXN(IOFM)
      IF( ITRACE.GE.1.AND.IOFM.GT.1) WRITE(6,1005)
      IF(ITRACE .GE. 1)  WRITE(6,1040) M,L,SIGOL,      NOFN
C
C             COMPUTE ALL THE C VALUES
C
      C7 = SIGN(1.,SIGOL)
      C11 = -C7*FLOAT(IAERO)
      C6 = 2.*FLOAT(ILOGIC) - 3.
      C3 = (2.*FLOAT(ISORDS) - 3.) * (-C7)
      C12 = C7
      C13 = C7
      C9 = C3*C7
      C14 = C9
      C8 = -C3
      IF( ITRACE .GE. 1 )  WRITE(6,1045) C3,C6,C7,C8,C9,C11,C12,C13,C14

              LOOP ON N

      DO 700 N=1,NOFN

              CALCULATE PROPAGATION FACTORS

      RMUMN = ARMUMN(N,IOFM)
      BETAMN = SQRT(RKSQD-CMACH*RMUMN**2)
      CAPKMN = (-RK*AXIALM + IFLOW*BETAMN)/CMACH
      CAPNMN = EGNNORM(M,RMUMN,ETA)
      IF(ITRACE .GE. 1)  WRITE(6,1050) N,RMUMN,BETAMN,CAPKMN,CAPNMN
C
C
C             COMPUTE MODAL AMPLITUDES
C
C
C
C             CALCULATE CONSTANT FACTOR, FACTCON
C
      ARGEXP =-CAPKMN*ZSBIR
      FACTCON = (-.25*NK1/BETAMN)*CMPLX(COS(ARGEXP),SIN(ARGEXP))
     1          *(TWOPI)
C
C             SET THE NUMBER OF SUB-INTERVALS FOR WHICH THE
C             LAST TWO TERMS ARE EVALUATED
C
      NORDGS = 2
      INTEGRL = (0.,0.)
C
C             SET NUMBER OSCILLATIONS
```

```
      NOSCE = ABS(SIGOL)*NK2
      NOSCE = MAXO(NOSCE,N)
      NOSCE = 1.5 * NOSCE
      NOSCE = MAXO(NOSCE,2)
      WIDTHI = (1.-ETA)/NOSCE
      IF(ITRACE .GE. 1)  WRITE(6,1060) FACTCON,NOSCE

                LOOP ON NUMBER OF SUBINTERVALS

      DO 500 ITRVL=1,NOSCE
      RLOW = ETA + (ITRVL-1)*WIDTHI
      RUP  = RLOW + WIDTHI
 [illegible]0 CONTINUE
      IF(ITRACE .GE. 1)  WRITE(6,1070) RLOW,RUP

                PERFORM GAUSSIAN INTEGRATION

      IF(ITRACE.EQ.3) WRITE(6,1115)
      CALL GAUSS2(RLOW,RUP,IORDGS,VALUINT,FACTIN2,ARMISC,MAXDIM,MAXJ,AR)

                ACCUMULATE THE TERMS

      INTEGRL = INTEGRL +        VALUINT

                END INTERVAL LOOP

      IF(ITRACE .GE. 1)  WRITE(6,1120) INTEGRL ,VALUINT
 [illegible]0 CONTINUE

                APPLY FIRST TERM AND STORE

      INTEGRL = FACTCON*INTEGRL
      ALPHAMN(N,IOFM)=INTEGRL
      IF(ITRACE .GE. 1)  WRITE(6,1130) INTEGRL

                END N AND M LOOPS

 [illegible]00 CONTINUE
 [illegible]00 CONTINUE
 1000 CONTINUE

                SAVE THE EIGENVALUE DETERMINING PARAMETERS

      ILOGICO = ILOGIC
      EGNBNDO = EGNBND
      ETAO    = ETA
      MDIMO   = MDIM
      NDIMO   = NDIM
      RSBNKIO = RSBNKI
      SIGNKIO = SIGNKI
```

```
C              RETURN
      RETJRN
C
 1005 FORMAT(1H1)
 1010 FORMAT(1H1,* JPTIONAL PRINTOJT FROM SUBROUTINE AABAA*)
 1015 FORMAT(1HC,* K1 = *,I2,8X,*K2 = *,I2,6X,*NK1 = *,I3,6X,*NK2 = *,
     1I3,3X,*3 = *,F10.4)
 1020 FORMAT(1HJ,* EIGENVALUE PARAMETERS GENERATED*/1H ,2X,*NSBNKI = *,
     1I2,10X,*SIGMA = *,F3.0,9X,*SIGNKI = *,F5.0,7X,* MSBT  = *,F10.4,
     22X,*RK   = *,F10.4,2X,*RKSQD  = *,F10.4/1H ,2X,*AXIALM = *,F10.4,
     32X,*CMACH = *,F10.4,2X,*EGNBND = *,F10.5,2X,*RS3NKI = *,F10.4,2X,
     4*ETA  = *,F10.4)
 1030 FORMAT(1HO,* THE EIGENVALUES FROM PREVIOUS CALL TO SUBRJUTINE *,
     1*AA3AA ARE REUSED FOR THIS CASE*)
 1040 FORMAT(1HO,* M = *,I4,3X,*L = *,I2,9X,*SIGOL = *,F6.2,
     1             6X,*NOFN = *,I2)
 1045 FORMAT(/11X,*C3 = *,F2.0,2X,*C6 = *,F2.0,2X,*C7 = *,F2.0,2X,*C8 =
     1*,F2.C,2X,*C9 = *,F2.0,2X,*C11 = *,F2.0,2X,*C12 = *,F2.0,2X,
     2*C13 = *,F2.0,2X,*C14 = *,F2.C)
 1050 FORMAT(1HO,* N = *,I4,3X,*RMJMN = *,F10.4,3X,*BETAMN = *,F10.4,3X,
     1*CAPKMN = *,F10.4,3X,*CAPNMN = *,F10.4)
 1060 FORMAT(1HO,* FACTCON = *,2F10.4,5X,*NOSCE = *,I2)
 1070 FORMAT(1HO,* RLOW = *,F9.4,3X,*RUP = *, F9.4)
 1115 FORMAT(1HO,3X,*RHO*,3X,*MMK1*,3X,*GAMMA*,6X,*ARHO*,9X,*CAPHRHO*,
     112X,*EXPORHO*,12X,*CAPKRHO*,9X,*FACT*,4X,*SCPTRMN*,9X,*FACTIN2*/)
 1120 FORMAT(1HO, * INTEGRL = *,2E12.4,4X,*VALUINT = *,2E12.4)
 1130 FORMAT(1HC,* INTEGRL = *,2E12.4)
      END
```

3.1.3 Subroutine BCDAA

Purpose: This subroutine computes the mode amplitudes, for a given harmonic, when a rotor operates in steady distortion. The computation essentially consists of determining which modes contribute significantly to the sum, computing the required modal parameters, and evaluating a definite integral per mode. This integral is equation (9) from appendix 1 of volume I expressed for numerical evaluation:

$$A^{\pm}_{mn\sigma} = \left\{\begin{matrix}\text{CONSTANT}\\ \text{FACTOR}\end{matrix}\right\} \sum_{j=1}^{N_{SUB}} \left\{\begin{matrix}\text{AVERAGE OF}\\ \text{NON-OSCILLATORY}\\ \text{FACTOR}\end{matrix}\right\}_j \int_{a_j}^{b_j} \left\{\begin{matrix}\text{OSCILLATORY}\\ \text{FACTOR}\end{matrix}\right\} d\rho$$

with

$$\left\{\begin{matrix}\text{CONSTANT}\\ \text{FACTOR}\end{matrix}\right\} = \frac{-N_R}{4\beta_{mn}} \quad e^{-iK^{\pm}_{mn}Z}$$

$$\left\{\begin{matrix}\text{NON-OSCILLATORY}\\ \text{FACTOR}\end{matrix}\right\}_j = C_2 \left(\frac{dC_L}{d\alpha}\right) M_M \; M_z \quad \sin\beta$$

$$* \left\{ \frac{m e_\phi}{\rho} + K^{\pm}_{mn} \; e_Z \right\} \quad \text{CAPLT}$$

$$\left\{\begin{matrix}\text{OSCILLATORY}\\ \text{FACTOR}\end{matrix}\right\} = W_\ell(\rho)\, \mathcal{R}_m\left(\mu_{mn}\rho\right)$$

See the FORTRAN dictionary (sec. 2.2) for CAPLT.

Method: The procedure is as follows:

1) Obtain the eigenvalue generation parameters (the input to EGNVAL2).

2) Compare these parameters to stored values to determine if the required eigenvalues are already available. If values are equal, proceed to step 6.

3) Compute the mode indexes and the corresponding eigenvalues.

4) Error return if correct eigenvalues have not been computed.

5) Loop on the spinning mode index.

6) Set values of required integers.

7) Loop on the radial mode index.

8) Compute the propagation constants and the normalization of the duct radial eigenfunction.

9) Compute the constant factor in the mode amplitude expression.

10) Initialize the value of the integral to zero.

11) Compute the number of equal subintervals required, which is determined by the total number of zeros of the oscillatory factor on the full integration interval.

12) Loop on subintervals.

13) Compute the lower and upper bound and the midpoint of the subinterval.

14) Set up for accessing the input geometric and aerodynamic data.

15) If the average value over the full interval of a geometric or aerodynamic variable is input, use it and proceed to step 17.

16) Compute an average value on the subinterval for the geometric or aerodynamic variable.

17) Initialize the nonoscillatory factor to the product of the average value of the first two variables appearing in that factor.

18) Compute flow angles and multiply the average value for the next three variables in the nonoscillatory factor into that factor.

19) Compute the reduced frequency and the lift function coefficients (used for noncompact factor also).

20) When the compact option is specified, compute the value for the frequency response function of the lift and multiply this into the nonoscillatory factor.

21) When the noncompact option is specified, compute the noncompact factor and multiply this into the nonoscillatory factor.

22) Compute the inner product, or projection, factor and multiply into the nonoscillatory factor.

23) Integrate the oscillatory factor over the subinterval.

24) Multiply the nonoscillatory and the integrated oscillatory factors together and accumulate in the integral value, completing the loop on the subintervals.

25) Multiply the constant factor into the integral value giving the mode amplitude for the current spinning mode index and radial mode index.

26) Save the current eigenvalue generation parameters from step 1. The eigenvalues will not have to be recomputed in the next execution if these parameters remain unchanged.

Usage: CALLING SEQUENCE

```
      DIMENSION MUSE(MDIM),MAXN(MDIM),ARMUMN(NDIM,MDIM),
     *    ARMISC(40),AR(MAXDIM,MAXJ,3)
      COMPLEX ALPHAMN(NDIM,MDIM)
          .
          .
          .
      CALL BCDAA(ARMISC,MAXDIM,MAXJ,AR,MDIM,NDIM,ARMUMN,NOFM,
     *    MUSE,MAXN,ALPHAMN,IERROR)
```

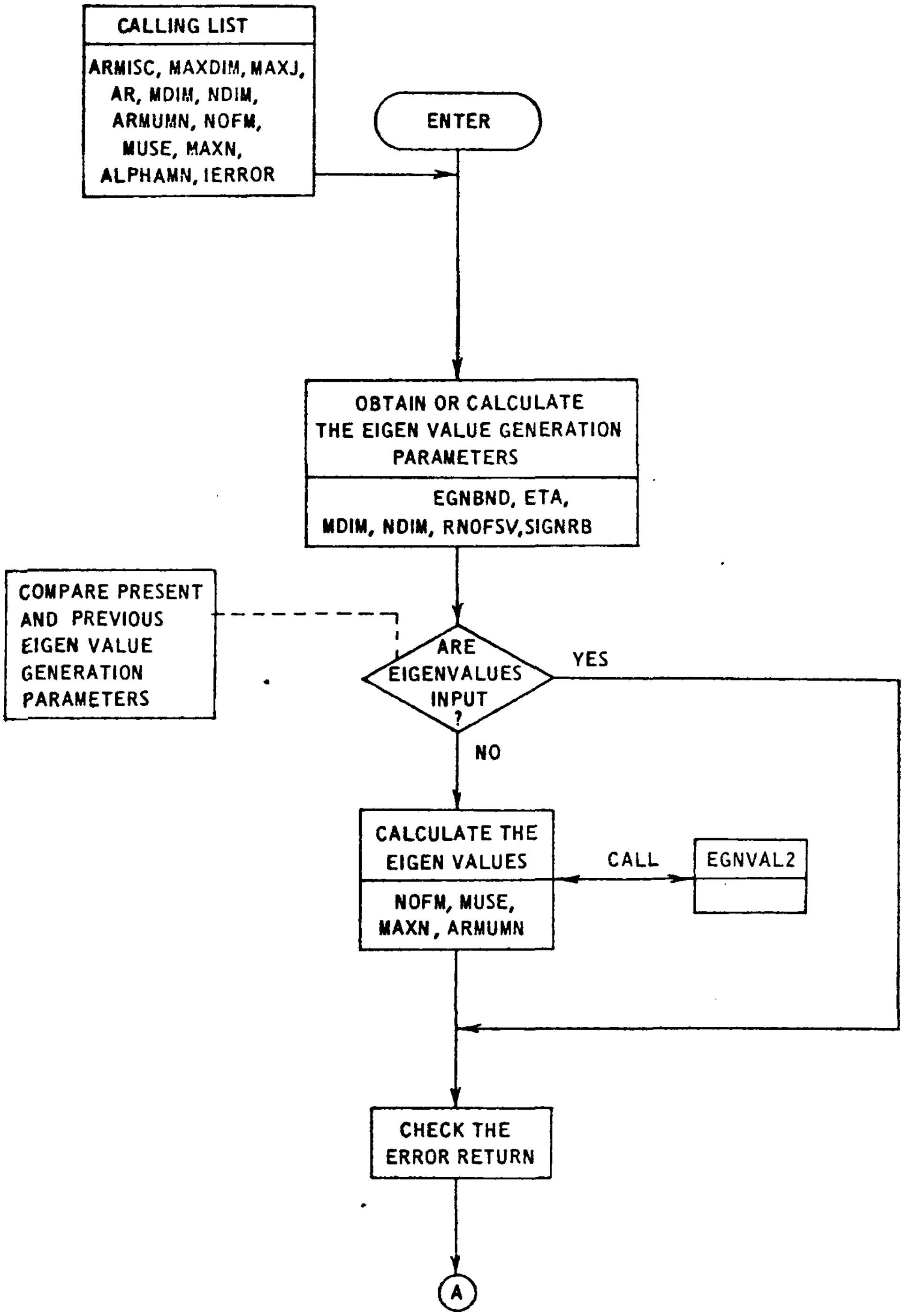
Restrictions: The use and restrictions on the input arrays ARMISC and AR and the input/output NOFM,MUSE,MAXN,ARMUMN are given in section 2.2.

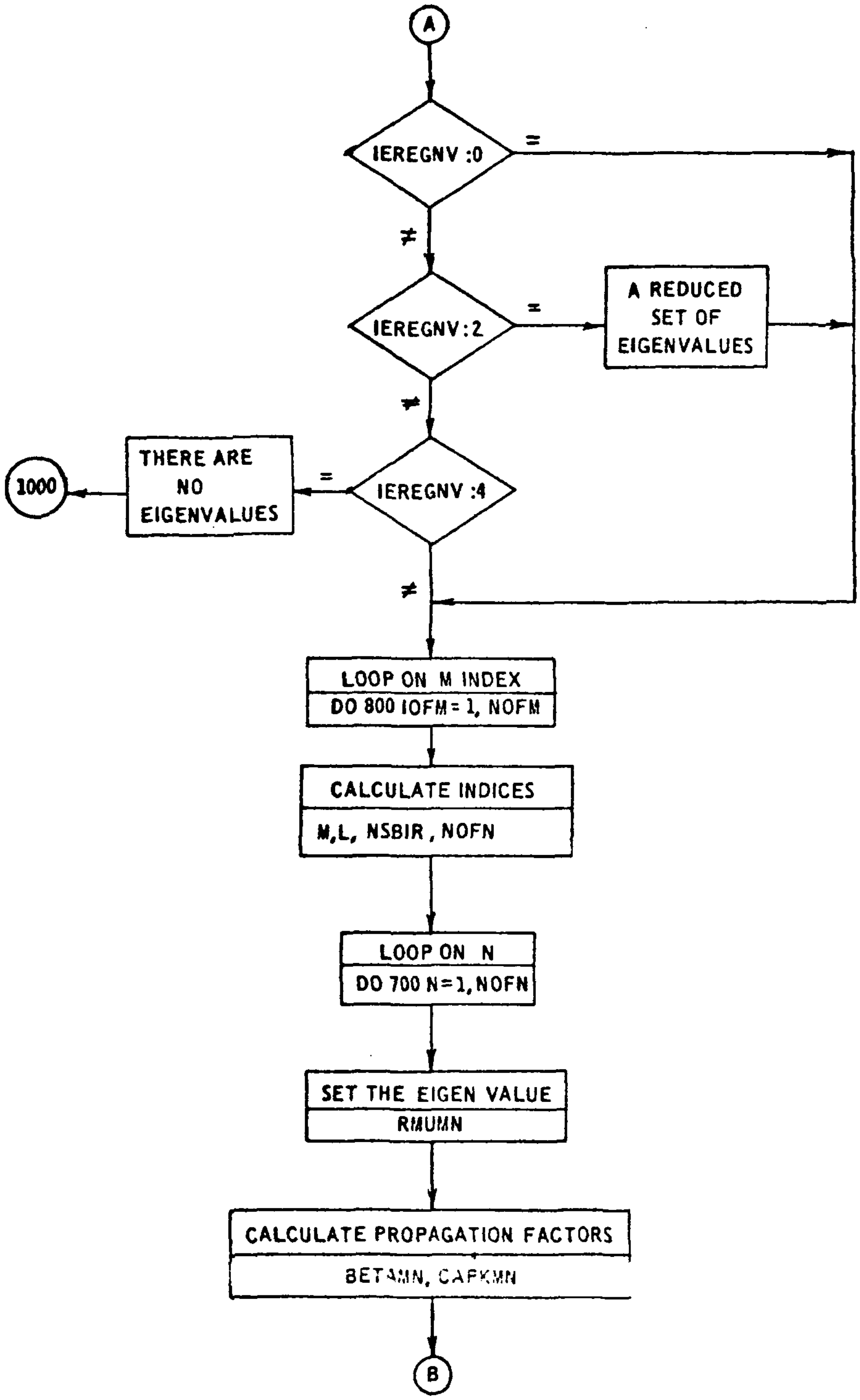
The maximum spinning mode is limited (see subroutine EGNVAL2) in absolute value to 100, and the maximum radial mode index as a result is at most 40.

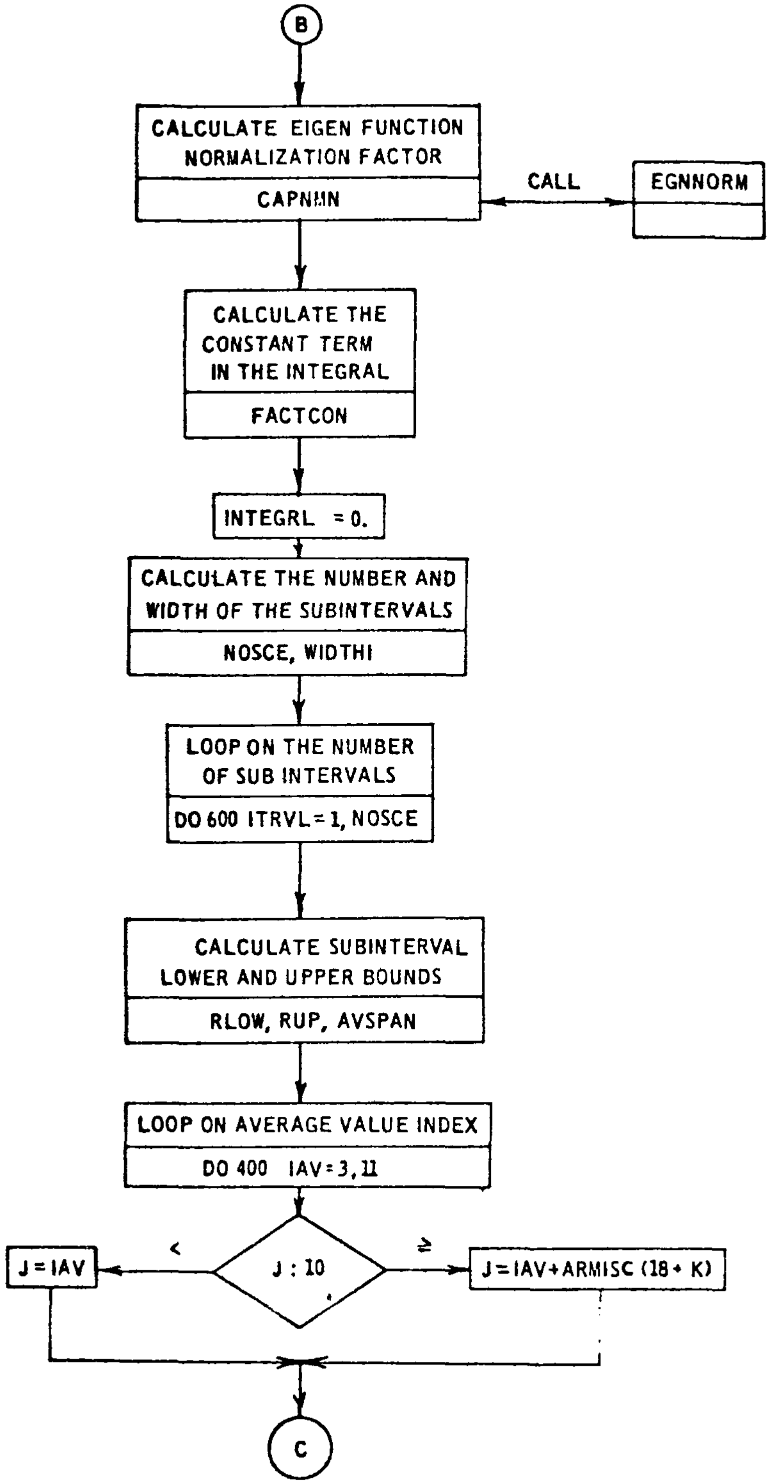
Error Return: IERROR (see the FORTRAN dictionary, sec. 2.2)

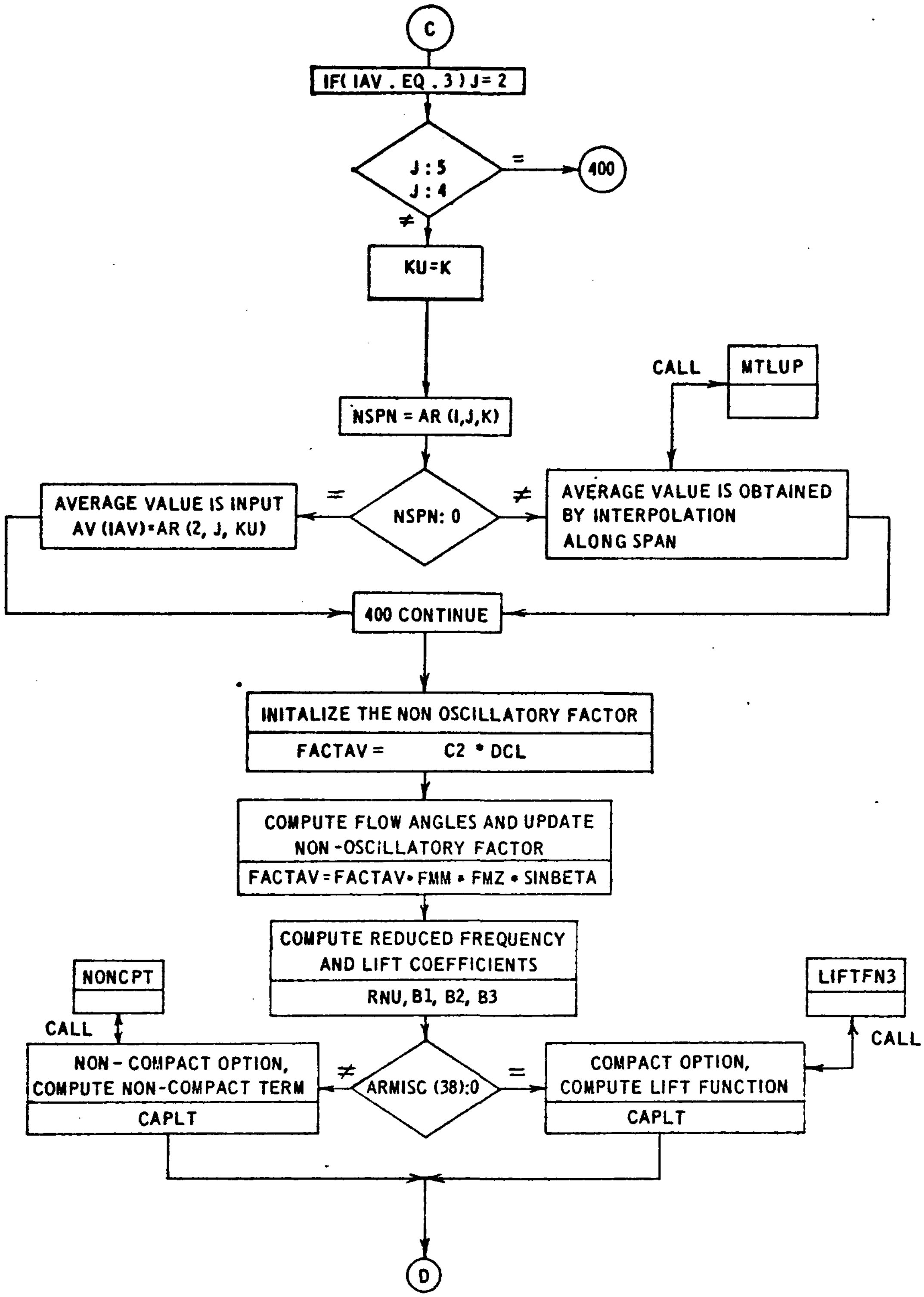
Printout and Diagnostics: See the definition of ARMISC(6),ITRACE, in the dictionary.

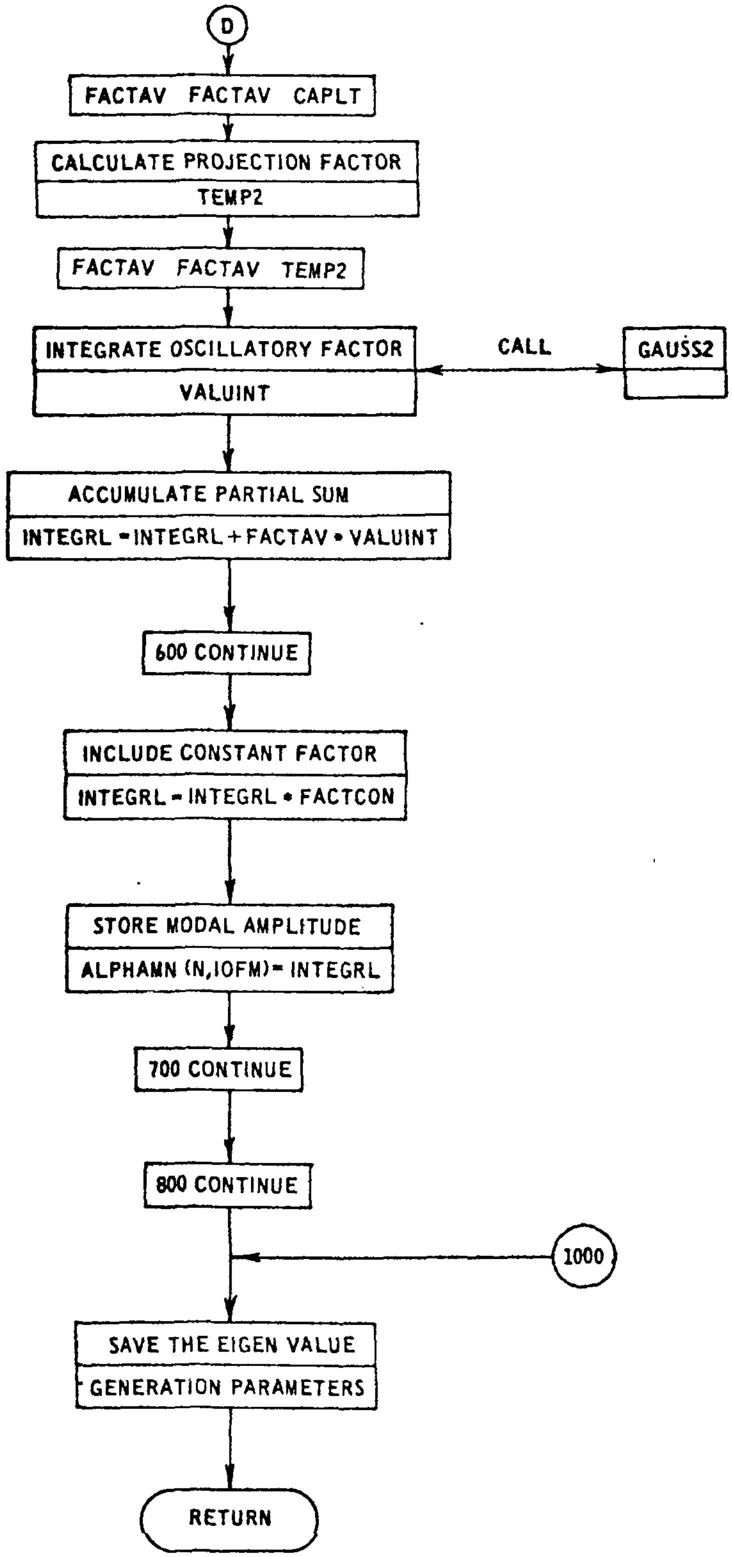
Timing: Of the cases run, the average time was 117 seconds per case.


CALLING LIST
ARMISC, MAXDIM, MAXJ, AR, MDIM, NDIM, ARMUMN, NOFM, MUSE, MAXN, ALPHAMN, IERROR
ENTER
OBTAIN OR CALCULATE THE EIGEN VALUE GENERATION PARAMETERS
EGNBND, ETA, MDIM, NDIM, RNOFSV, SIGNRB
COMPARE PRESENT AND PREVIOUS EIGEN VALUE GENERATION PARAMETERS
ARE EIGENVALUES INPUT ?
YES
NO
CALCULATE THE EIGEN VALUES
NOFM, MUSE, MAXN, ARMUMN
CALL
EGNVAL2
CHECK THE ERROR RETURN
A


A
IEREGNV :0
=
≠
IEREGNV:2
=
A REDUCED SET OF EIGENVALUES
≠
THERE ARE NO EIGENVALUES
=
IEREGNV :4
1000
≠
LOOP ON M INDEX
DO 800 IOFM = 1, NOFM
CALCULATE INDICES
M,L, NSBIR , NOFN
LOOP ON N
DO 700 N=1,NOFN
SET THE EIGEN VALUE
RMUMN
CALCULATE PROPAGATION FACTORS
BETAMN, CAPKMN
B


B
CALCULATE EIGEN FUNCTION NORMALIZATION FACTOR
CAPNMN
CALL
EGNNORM
CALCULATE THE CONSTANT TERM IN THE INTEGRAL
FACTCON
INTEGRL = 0.
CALCULATE THE NUMBER AND WIDTH OF THE SUBINTERVALS
NOSCE, WIDTHI
LOOP ON THE NUMBER OF SUB INTERVALS
DO 600 ITRVL = 1, NOSCE
CALCULATE SUBINTERVAL LOWER AND UPPER BOUNDS
RLOW, RUP, AVSPAN
LOOP ON AVERAGE VALUE INDEX
DO 400 IAV = 3, 11
J = IAV
<
J : 10
≥
J = IAV+ARMISC (18 + K)
C


C
IF(IAV . EQ . 3) J= 2
J : 5
J : 4
=
≠
400
KU=K
CALL
MTLUP
NSPN = AR (1,J,K)
AVERAGE VALUE IS INPUT
AV (IAV)=AR (2, J, KU)
=
NSPN: 0
≠
AVERAGE VALUE IS OBTAINED
BY INTERPOLATION
ALONG SPAN
400 CONTINUE
INITALIZE THE NON OSCILLATORY FACTOR
FACTAV = C2 * DCL
COMPUTE FLOW ANGLES AND UPDATE
NON -OSCILLATORY FACTOR
FACTAV = FACTAV * FMM * FMZ * SINBETA
COMPUTE REDUCED FREQUENCY
AND LIFT COEFFICIENTS
RNU, B1, B2, B3
NONCPT
CALL
NON - COMPACT OPTION,
COMPUTE NON-COMPACT TERM
CAPLT
≠
ARMISC (38):0
=
COMPACT OPTION,
COMPUTE LIFT FUNCTION
CAPLT
LIFTFN3
CALL
D


D
FACTAV FACTAV CAPLT
CALCULATE PROJECTION FACTOR
TEMP2
FACTAV FACTAV TEMP2
INTEGRATE OSCILLATORY FACTOR
VALUINT
CALL
GAUSS2
ACCUMULATE PARTIAL SUM
INTEGRL = INTEGRL + FACTAV * VALUINT
600 CONTINUE
INCLUDE CONSTANT FACTOR
INTEGRL = INTEGRL * FACTCON
STORE MODAL AMPLITUDE
ALPHAMN (N,IOFM) = INTEGRL
700 CONTINUE
800 CONTINUE
1000
SAVE THE EIGEN VALUE
GENERATION PARAMETERS
RETURN

```
      SUBROUTINE BCDAA(ARMISC,MAXDIM,MAXJ,AR,MDIM,NDIM,ARMUMN,NOFM,MUSE,
     1MAXN,ALPHAMN,IERROR)
C
      REAL  MSBT
      COMPLEX ALPHAMN(NDIM,MDIM)
      COMPLEX FACTAV,FACTCON,FACTIN3,INTEGRL,CAPLT ,VALUINT
C
      DIMENSION ARMISC(1),AR(MAXDIM,MAXJ,3),ARMUMN(NDIM,MDIM),
     1MUSE(MDIM),MAXN(MDIM)
      DIMENSION AV(11)
C
      DATA          EGNBND0,ETA0,MDIMO,NDIMO,RNOFSVO,SIGNRBO/
     1        -1.,-1.,0,0,0.,0./
      DATA PI,TWOPI        /3.14159265358979,6.28318530717959/
C
      COMMON/CFACT/  M,N,RMUMN,CAPNMN,ETA,SIGN,L,CAPKMN
C
      EQUIVALENCE (AV(1),AVSPAN),              (AV(3),C2),
     1(AV(6),DCL),(AV(7),FMI),(AV(8),FME),(AV(9),FMZ)
C
      EXTERNAL FACTIN3
C
C
      ITRACE = ARMISC(6)
      IF(ITRACE .GE. 1)  WRITE(6,1010)
C
C             GENERATE THE EIGENVALUES
C
      NSBRB  = ARMISC(10)
      SIGMA  = ARMISC(14)
      SIGNRB = SIGMA*NSBRB
      MSBT   = ARMISC(7)
      RK    = SIGNRB*MSBT
      RKSQD= RK**2
      AXIALM = AR(2,9,2)
      CMACH = 1.-AXIALM**2
      EGNBND = RK/ SQRT(CMACH)
      RNOFSV = 1
      ETA = ARMISC(3)
      IF(ITRACE .GE. 1)  WRITE(6,1020) NSBRB,SIGMA,SIGNRB,MSBT,RK,RKSQD,
     1                                 AXIALM,CMACH,EGNBND,RNOFSV,ETA
      IF( EGNBND  .NE. EGNBND0 )  GO TO 110
      IF( ETA     .NE. ETA0    )  GO TO 110
      IF( MDIM    .NE. MDIMO   )  GO TO 110
      IF( NDIM    .NE. NDIMO   )  GO TO 110
      IF( RNOFSV .NE. RNOFSV0 )  GO TO 110
      IF( SIGNRB .NE. SIGNRB0 )  GO TO 110
      IF(ITRACE .GE. 1)  WRITE(6,1030)
      GO TO 120
C
  110 CALL EGNVAL2( 2     ,EGNBND,ETA,MDIM,NDIM,RNOFSV,SIGNRB,ITRACE,
     1              NOFM,MUSE,MAXN,ARMUMN,IEREGNV)
```

```
      IERROR = IEREGNV
 120  CONTINUE
C
C               ERROR RETURN
C
      IF(IEREGNV.EQ.0) GO TO 200
      IF(IEREGNV-2) 150,130,150
 130  IF(ITRACE.NE.0) WRITE(6,140)
 140  FORMAT(//1H0,70(1H*)//1H0,*A REDUCED SET OF EIGENVALUES IS AVAILAB
     1LE*/1H0,*COMPUTATIONS WILL PROCEDE*/1H0,70(1H*) )
      GO TO 200
 150  IF(IEREGNV-4) 200,160,200
 160  IF(ITRACE.NE.0) WRITE(6,180)
 180  FORMAT(//1H0,70(1H*)//1H0,*THERE ARE NO PROPAGATING RADIAL MODES*/
     11H0,*NO COMPUTATIONS CAN BE MADE*/1H0,70(1H*) )
      GO TO 1000
 200  CONTINUE
C
C
C
      ZSBIR = ARMISC(17)
      IFLOW=ARMISC(4)
C
C
C               LOOP ON M
C
      DO 800 IOFM=1,NOFM
C
C               SET M,L, AND NOFN
C
      M = MUSE(IOFM)
      L = (M-SIGNRB)/RNOFSV
      NOFN = MAXN(IOFM)
      IF(IOFM.GT.1.AND.ITRACE.GE.1) WRITE(6,1005)
      IF(ITRACE .GE. 1)  WRITE(6,1040) M,L,       NSBRB,NOFN
C
C               LOOP ON N
C
      DO 700 N=1,NOFN
C
C               CALCULATE PROPAGATION FACTORS
C
      RMUMN = ARMUMN(N,IOFM)
      BETAMN = SQRT(RKSQD-CMACH*RMUMN**2)
      CAPKMN = (-RK*AXIALM + IFLOW*BETAMN)/CMACH
      CAPNMN = EGNNORM(M,RMUMN,ETA)
      IF(ITRACE .GE. 1)  WRITE(6,1050) N,RMUMN,BETAMN,CAPKMN,CAPNMN
C
C
C               COMPUTE MODAL AMPLITUDES
C
C
C
```

```
C
C             CALCULATE CONSTANT FACTOR, FACTCON
C
      TEMP1 = -CAPKMN*ZSBIR
      FACTCON = - .25*NSBRB/(BETAMN    )*(
     1CMPLX( COS(TEMP1), SIN(TEMP1) ) )
C
C             SET THE NUMBER OF SUB-INTERVALS FOR WHICH THE
C             LAST TWO TERMS ARE EVALUATED
C
      IORDGS = 2
      INTEGRL = (0.,0.)
C
C             SET NUMBER OSCILLATIONS
C
      NOSCE  = 1
      NOSCE = MAX0(NOSCE,N)
      NOSCE = 1.5 * NOSCE
      NOSCE = MAX0(NOSCE,2)
      WIDTHI = (1.-ETA)/NOSCE
      IF(ITRACE .GE. 1)  WRITE(6,1060) FACTCON,NOSCE

C             LOOP ON NUMBER OF SUBINTERVALS

      DO 600 ITRVL=1,NOSCE
      RLOW = ETA + (ITRVL-1)*WIDTHI
      RUP  = RLOW + WIDTHI
  570 CONTINUE
C
C             EVALUATE TERM TO BE AVERAGED
C
C             SET AVERAGE SPAN
C
      AVSPAN = (RLOW + RUP )*.5
      AV(1) = AVSPAN
C
C             SET K INDEX
C
      K = 2
C
C             SET AVERAGE VALUES
C
      DO 400 IAV=3,11
C
C             SET J INDEX
C
      J = IAV
      IF(IAV.EQ.3) J=2
      IF(IAV.EQ.10)J=9+ARMISC(19+K)+1
      IF(IAV.EQ.11)J=9+ARMISC(13+K)+2
      IF( J.EQ.4 ) GO TO 400
      IF( J.EQ.5 ) GO TO 400
```

```
C
C             SET K INDEX TO BE USED
C
      KU = K
C
C             SET SPAN WHEN J=1
C
      NSPN = AR(1,J,KU)
C
C             AVERAGE VALUE IS INPUT
C
      IF( NSPN ) 400,330,340
  330 AV( IAV) = AR(2,J,KU)
      GO TO 400
C
C             INTERPOLATE FOR AVERAGE VALUE
C
  340 IPA=-1
      CALL MTLUP(AVSPAN,AV(IAV),1,NSPN,NSPN,1,IPA,AR(3,1,KU),AR(3,J,KU))
  400 CONTINUE
      IF(ITRACE .GE. 1)  WRITE(6,1070) RLOW,RUP,AVSPAN    ,C2    ,DCL,FMI,
     1                                 FME,FMZ,AV(10),AV(11)
C
C             CALCULATE THE AVERAGE FACTOR, FACTAV
C
C
      FACTAV =     C2*DCL
      IF(ITRACE .GE. 1)  WRITE(6,1080) FACTAV
C
C             COMPUTE  MACH NUMBER RELATED VARIABLES
C
      TEMP1 = SQRT( FMI**2 -FMZ**2 )
      TEMP2 = SQRT( FME**2 -FMZ**2 )
      TEMP3 =0.25*(TEMP1+TEMP2)**2
      FMM   = SQRT( FMZ**2 + TEMP3 )
      COSTHS = FMZ/FMM
      SINTHS = SQRT(1. - COSTHS**2)
      SINBETA = SINTHS
      COTBETA = COSTHS/SINTHS
C
C             UPDATE AVERAGE FACTOR
C
      FACTAV = FACTAV*FMM*FMZ*SINBETA
      IF( ITRACE.GE.1) WRITE(6,1085) TEMP1,TEMP2,TEMP3,FMM,COSTHS,
     1                               SINBETA,FACTAV
C
C             COMPUTE REDUCED FREQUENCY
C
      RNU = .5*L*C2*MSBT/FMM
      B1 = 1.
      B2 = -AV(11)*COTBETA
      B3 = -AV(10)*COTBETA
```

```
C
C               COMPUTE COMPACT OPTION
C
      IF( ARMISC(38).NE.O.) GO TO 410
      CALL LIFTFN3(RNU,B1,B2,B3,CAPLT)
      GO TO 420
C
C               COMPUTE NON-COMPACT OPTION
C
  410 CONTINUE
      CALL NONCPT(B1,B2,B3,C2,CAPKMN,COSTHS,M,AVSPAN,RNU,SINTHS,CAPLT)
C
C               UPDATE AVERAGE FACTOR
C
  420 FACTAV = FACTAV*CAPLT
      IF( ITRACE.GE.1 ) WRITE(6,1090) FACTAV,CAPLT
C
      TEMP2  =        M*COSTHS/AVSPAN + CAPKMN*SINTHS
      FACTAV = FACTAV*TEMP2
      IF(ITRACE .GE. 1)  WRITE(6,1100) FACTAV,TEMP2
C
C               PERFORM GAUSSIAN INTEGRATION
C
      CALL GAUSS2(RLOW,RUP,IORDGS,VALUINT,FACTIN3,ARMISC,MAXDIM,MAXJ,AR)
C
C               ACCUMULATE THE TERMS
C
      INTEGRL = INTEGRL + FACTAV*VALUINT
C
C               END INTERVAL LOOP
C
      IF(ITRACE.GE.1) WRITE(6,1120)FACTAV,VALUINT,INTEGRL
  600 CONTINUE
C
C               APPLY FIRST TERM AND STORE
C
      INTEGRL = FACTCON*INTEGRL
      ALPHAMN(N,IOFM)=INTEGRL
      IF(ITRACE .GE. 1)  WRITE(6,1130) INTEGRL
C
C               END N AND M LOOPS
C
  700 CONTINUE
  800 CONTINUE
 1000 CONTINUE
C
C               SAVE THE EIGENVALUE DETERMINING PARAMETERS
C
      EGN3NDO = EGN3ND
      ETAO    = ETA
      MDIMO   = MDIM
      NDIMO   = NDIM
```

```
      RNOFSVO = RNOFSV
      SIGNRBO = SIGNRB
C
      RETURN
C
 1005 FORMAT(1H1//)
 1010 FORMAT(1H1//11X,* OPTIONAL PRINTOUT FROM SUBROUTINE BCOAA*)
 1020 FORMAT(1HO,10X,*EIGENVALUE PARAMETERS GENERATED*/13X,*NSBRB = *,
     1I3,2X,*SIGMA = *,F3.0,2X,*SIGNRB = *,F6.0,2X,*MSBT = *,F1C.4/13X,
     2*RK = *,F10.4,2X,*RKSQD = *,F10.4,2X,*AXIALM = *,F10.4/13X,
     3*CMACH = *,F10.4,2X,*EGNBND = *,2X,F10.5,2X,*RNOFSV = *,F10.4/13X,
     4*ETA = *,F10.5)
 1030 FORMAT(1HO,10X,*THE EIGENVALUES FROM PREVIOUS CALL TO SUBROUTINE B
     1BCAA*/11X,*ARE REUSED FOR THIS CASE*)
 1040 FORMAT(1HO,10X,*M = *,I4,2X,*L = *,I4,2X,
     1*NSBRB = *,I3,2X,*NOFN = *,I4)
 1050 FORMAT(1HO,10X,*N = *,I4,2X,*RMUMN = *,F10.4,2X,*BETAMN = *,F10.4/
     120X,*CAPKMN = *,F1C.4,2X,*CAPNMN = *,F10.4)
 1060 FORMAT(1HO,10X,*FACTCON = *,2E12.4,2X,*NOSCE = *,I4)
 1070 FORMAT(1HO,10X,*RLOW = *,F9.4,2X,*RUP = *,F9.4,2X,*AVSPAN = *,F9.4
     1/11X,*C2 = *,F10.4,2X,                     *DCL = *,F10.4/11X,
     2*FMI = *F9.4,2X,*FME = *,F9.4,2X,*FMZ = *,F9.4/
     311X,*AV(10) = *,F9.4,2X,*AV(11) = *,F9.4)
 1080 FORMAT(1HO,10X,*FACTAV = *,2E12.4)
 1085 FORMAT(1HO,* TEMP1 = *,F9.4,2X,*TEMP2 = *,F9.4,2X,*TEMP3 = *,F9.4/
     1          1X,* FMM   = *,F9.4,2X,*COSTHS= *,F9.4,2X,*SINBETA=*,F9.4/
     2          1X,* FACTAV= *,2E12.4)
 1090 FORMAT(1HO,10X,*FACTAV = *,2E12.4,2X,*CAPLT = *,2E12.4)
 1100 FORMAT(1HO,10X,*FACTAV = *,2E12.4,2X,*TEMP2 = *,2E12.4)
 1120 FORMAT(1HO,10X,*FACTAV =*,2E12.4,2X,*VALUINT =*,2E12.4/
     111X,*INTEGRL =*2E12.4)
 1130 FORMAT(1HO,10X,*INTEGRL =*,2E12.4)
      END
```

3.1.4 Subroutine BBCAA

Purpose: This subroutine computes the mode amplitudes for a given harmonic. The pressure results from the nonstationary lift induced on the rotor blades as they cut through an eddy which is convected with the flow. The computation consists of determining which modes contribute significantly to the sum, computing the required modal parameters, and evaluating a definite integral per mode. The integral is equation (9) from appendix I of volume I and is expressed for numerical evaluation:

$$A^{\pm}_{mn\sigma} = \left\{\begin{matrix}\text{CONSTANT}\\ \text{FACTOR}\end{matrix}\right\} \sum_{j=1}^{N_{SUB}} \left\{\begin{matrix}\text{AVERAGE OF}\\ \text{NON-OSCILLATORY}\\ \text{FACTOR}\end{matrix}\right\}_j \int_{a_j}^{b_j} \left\{\begin{matrix}\text{OSCILLATORY}\\ \text{FACTOR}\end{matrix}\right\} d\rho$$

with

$$\left\{\begin{matrix}\text{CONSTANT}\\ \text{FACTOR}\end{matrix}\right\} = \frac{-N}{4\beta_{mn}} \quad e^{-iK^{\pm}_{mn}Z}$$

$$\left\{\begin{matrix}\text{NON-OSCILLATORY}\\ \text{FACTOR}\end{matrix}\right\}_j = C\left(\frac{dC_L}{d\alpha}\right) M_Z \left\{\frac{me_\phi}{\rho} + K^{\pm}_{mn}\, e_Z\right\}$$

$$\left\{\begin{matrix}\text{OSCILLATORY}\\ \text{FACTOR}\end{matrix}\right\} = \text{FACTIN4}$$

See the FORTRAN dictionary (sec. 2.2) for FACTIN4.

Method: The procedure is as follows:

1) Obtain the eigenvalue generation parameters (the input to EGNVAL2).

2) Compare these parameters to stored values to determine if the required eigenvalues are already available. If values are equal, proceed to step 4.

3) Compute the mode indexes and the corresponding eigenvalues.

4) Error return if correct eigenvalues have not been computed.

5) Loop on the spinning mode index.

6) Set values of required integers.

7) Loop on the radial mode index.

8) Compute the propagation constants and the normalization of the duct radial eigenfunction.

9) Compute the constant factor in the mode amplitude expression.

10) Initialize the value of the integral to zero.

11) Compute the number of equal subintervals required, which is determined by the total number of zeros of the oscillatory factor on the full integration interval.

12) Loop on subintervals.

13) Compute the lower and upper bound and the midpoint of the subinterval.

14) Set up for accessing the input geometric and aerodynamic data.

15) If the average value over the full interval of a geometric or aerodynamic variable is input, use it and proceed to step 17.

16) Compute an average value on the subinterval for the geometric or aerodynamic variable.

17) Initialize the nonoscillatory factor to the product of the average value of the first two variables appearing in that factor.

18) Multiply the average axial Mach number into the nonoscillatory factor.

19) Compute the inner product, or projection, factor and multiply into the nonoscillatory factor.

20) Integrate the oscillatory factor over the subinterval.

21) Multiply the nonoscillatory and the integrated oscillatory factors together and accumulate in the integral value, completing the loop on the subintervals.

22) Multiply the constant factor into the integral value giving the mode amplitude for the current spinning mode index and radial mode index.

23) Save the current eigenvalue generation parameters from step 1. The eigenvalues will not have to be recomputed in the next execution if these parameters remain unchanged.

Usage: CALLING SEQUENCE

```
 DIMENSION MUSE(MDIM),MAXN(MDIM),ARMUMN(NDIM,MDIM)
*    ARMISC(40),AR(MAXDIM,MAXJ,3)
 COMPLEX ALPHAMN(NDIM,MDIM)
     .
     .
     .
 CALL BBCAA(ARMISC,MAXDIM,MAXJ,AR,MDIM,NDIM,ARMUMN,NOFM,
*    MUSE,MAXN,ALPHAMN,IERROR)
```

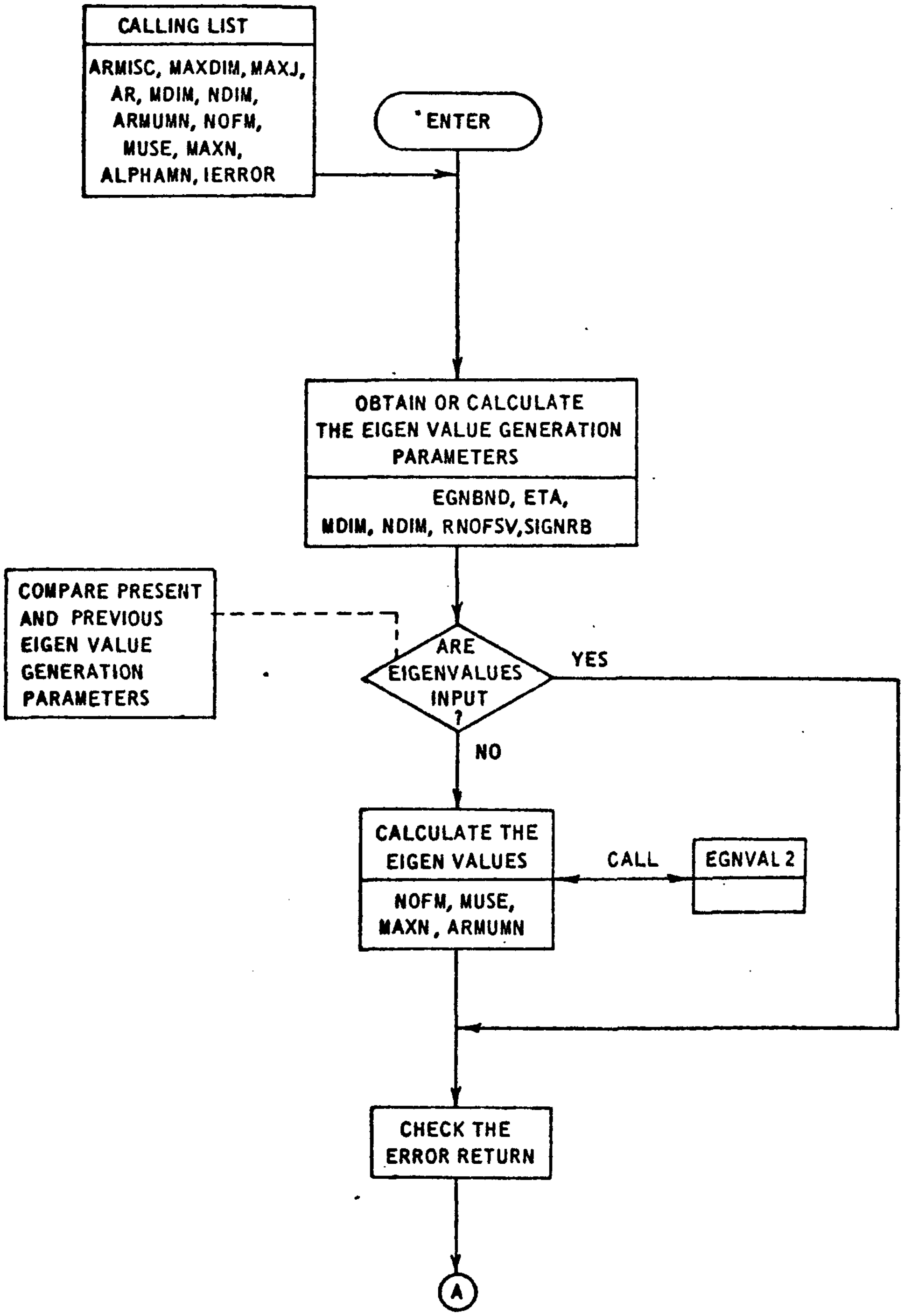
Restrictions: The use and restrictions on the input arrays ARMISC and AR and the input/output NOFM,MUSE,MAXN,ARMUMN are given in section 2.2.

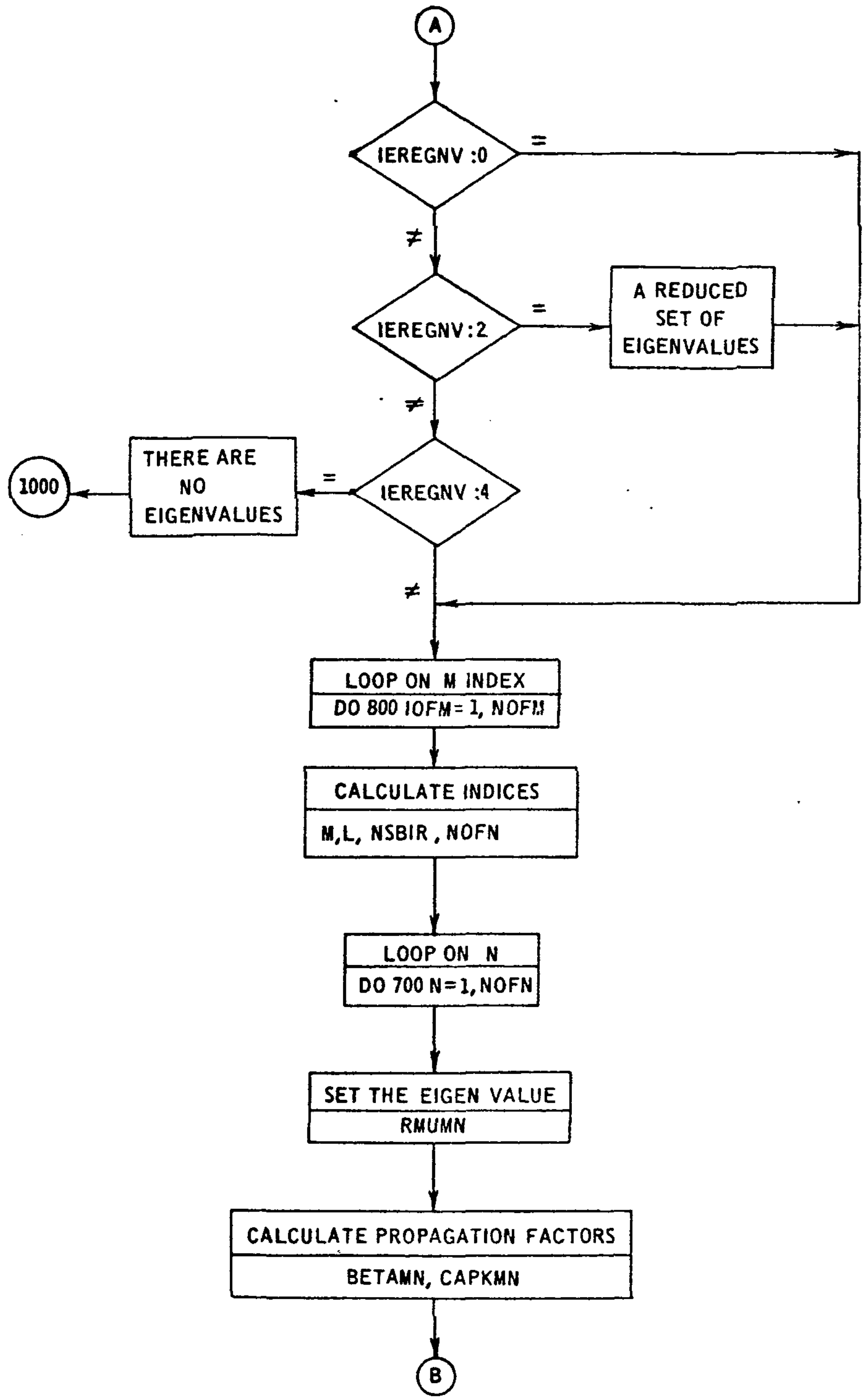
The maximum spinning mode is limited (see subroutine EGNVAL2) in absolute value to 100, and the maximum radial mode index as a result is at most 40.

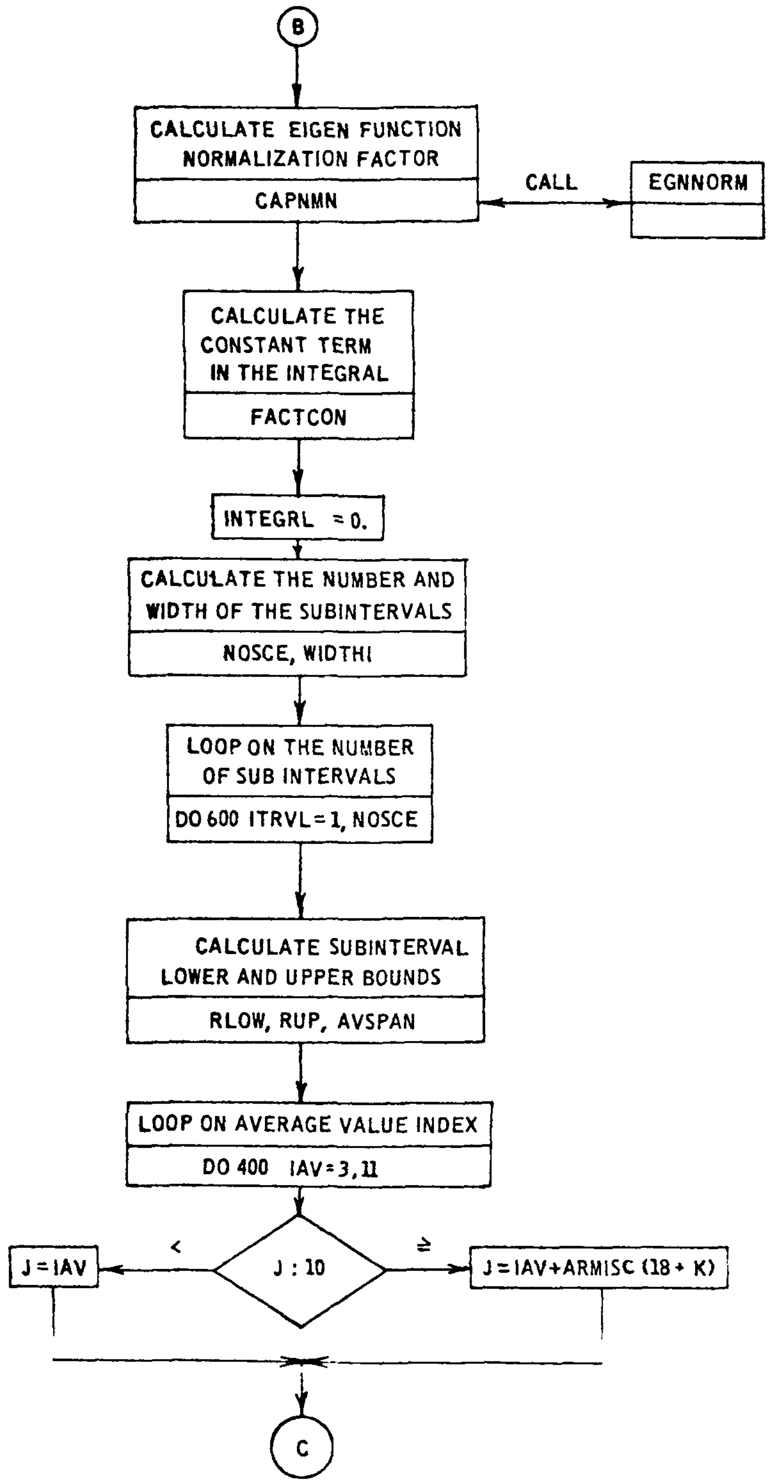
Error Return: IERROR (see the FORTRAN dictionary, sec. 2.2)

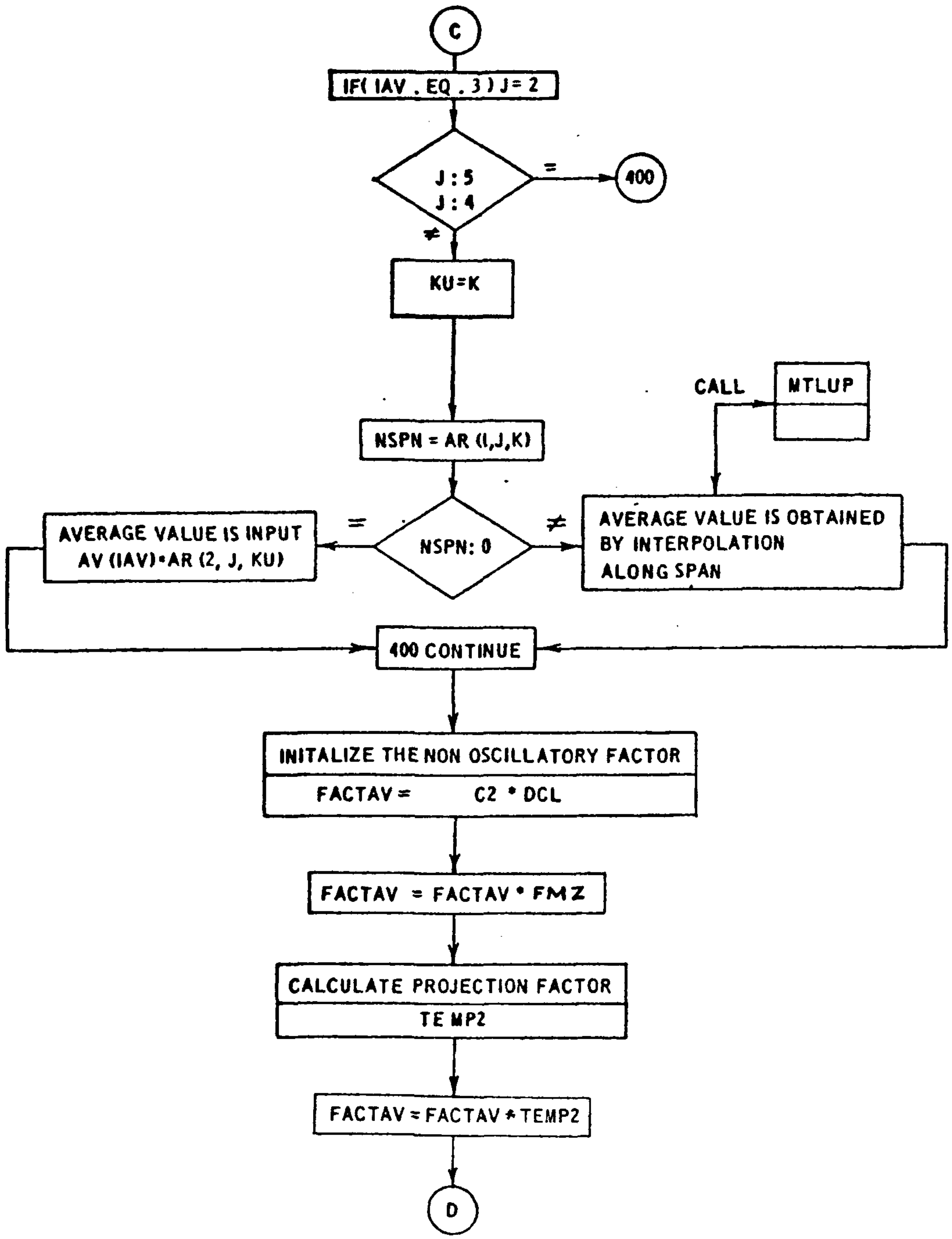
Printout and Diagnostics: See the definition of ARMISC(6),ITRACE, in the dictionary.

Timing: Of the cases run, the average time was 145 seconds per case.

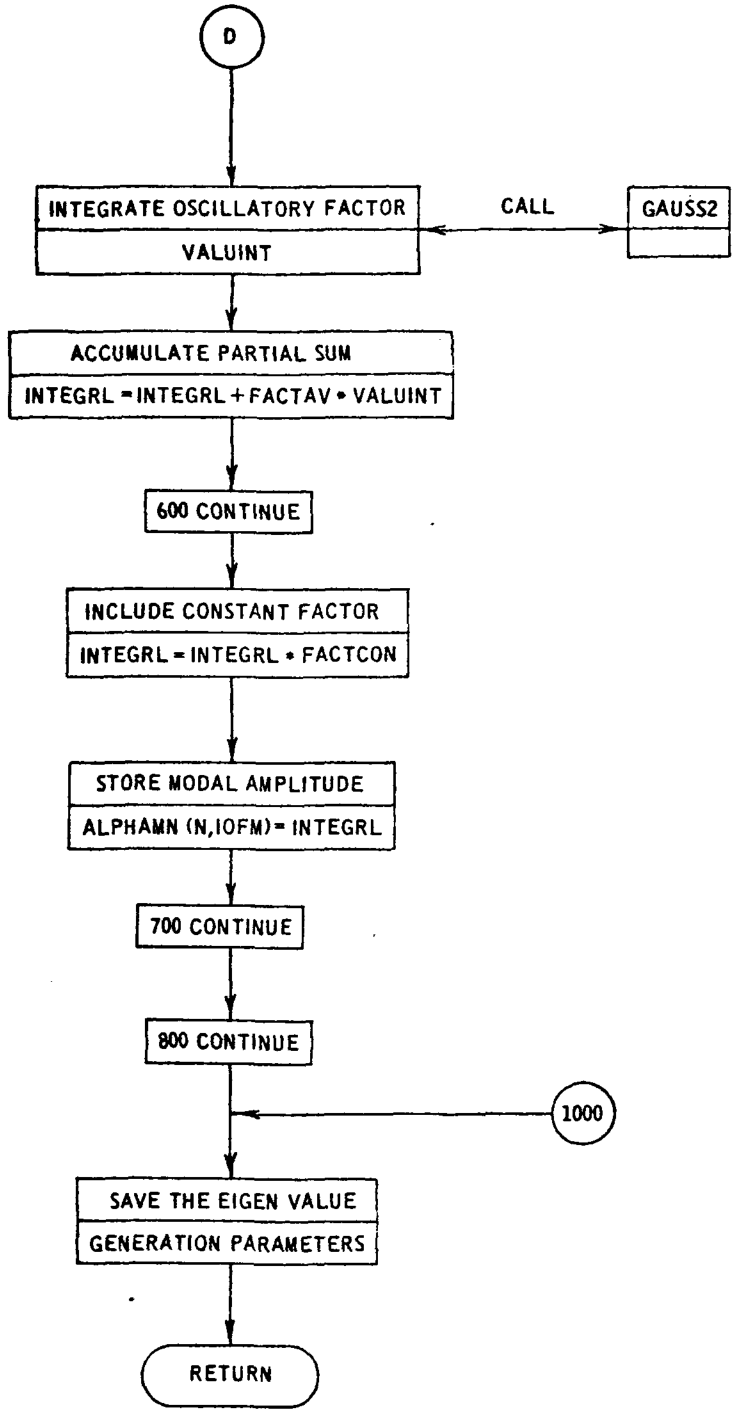

CALLING LIST
ARMISC, MAXDIM, MAXJ,
AR, MDIM, NDIM,
ARMUMN, NOFM,
MUSE, MAXN,
ALPHAMN, IERROR
ENTER
OBTAIN OR CALCULATE THE EIGEN VALUE GENERATION PARAMETERS
EGNBND, ETA,
MDIM, NDIM, RNOFSV, SIGNRB
COMPARE PRESENT AND PREVIOUS EIGEN VALUE GENERATION PARAMETERS
ARE EIGENVALUES INPUT ?
YES
NO
CALCULATE THE EIGEN VALUES
NOFM, MUSE,
MAXN, ARMUMN
CALL
EGNVAL 2
CHECK THE ERROR RETURN
A


A
IEREGNV :0
=
≠
IEREGNV:2
=
A REDUCED SET OF EIGENVALUES
≠
IEREGNV :4
=
THERE ARE NO EIGENVALUES
1000
≠
LOOP ON M INDEX
DO 800 IOFM=1, NOFM
CALCULATE INDICES
M,L, NSBIR, NOFN
LOOP ON N
DO 700 N=1,NOFN
SET THE EIGEN VALUE
RMUMN
CALCULATE PROPAGATION FACTORS
BETAMN, CAPKMN
B


B
CALCULATE EIGEN FUNCTION NORMALIZATION FACTOR
CAPNMN
CALL
EGNNORM
CALCULATE THE CONSTANT TERM IN THE INTEGRAL
FACTCON
INTEGRL = 0.
CALCULATE THE NUMBER AND WIDTH OF THE SUBINTERVALS
NOSCE, WIDTHI
LOOP ON THE NUMBER OF SUB INTERVALS
DO 600 ITRVL = 1, NOSCE
CALCULATE SUBINTERVAL LOWER AND UPPER BOUNDS
RLOW, RUP, AVSPAN
LOOP ON AVERAGE VALUE INDEX
DO 400 IAV = 3, 11
J : 10
<
≥
J = IAV
J = IAV+ARMISC (18 + K)
C



C
IF(IAV . EQ . 3) J = 2
J : 5
J : 4
=
400
≠
KU=K
CALL
MTLUP
NSPN = AR (I,J,K)
AVERAGE VALUE IS INPUT
AV (IAV)=AR (2, J, KU)
=
NSPN: 0
≠
AVERAGE VALUE IS OBTAINED
BY INTERPOLATION
ALONG SPAN
400 CONTINUE
INITALIZE THE NON OSCILLATORY FACTOR
FACTAV = C2 * DCL
FACTAV = FACTAV * FMZ
CALCULATE PROJECTION FACTOR
TEMP2
FACTAV = FACTAV * TEMP2
D


D
INTEGRATE OSCILLATORY FACTOR
VALUINT
CALL
GAUSS2
ACCUMULATE PARTIAL SUM
INTEGRL = INTEGRL + FACTAV * VALUINT
600 CONTINUE
INCLUDE CONSTANT FACTOR
INTEGRL = INTEGRL * FACTCON
STORE MODAL AMPLITUDE
ALPHAMN (N,IOFM) = INTEGRL
700 CONTINUE
800 CONTINUE
1000
SAVE THE EIGEN VALUE
GENERATION PARAMETERS
RETURN

```
      SUBROUTINE BBCAA(ARMISC,MAXDIM,MAXJ,AR,MDIM,NDIM,ARMUMN,NOFM,MUSE,
     1MAXN,ALPHAMN,IERROR)
C
      REAL MSBT
      COMPLEX ALPHAMN(NDIM,MDIM)
      COMPLEX CAPLT,FACTAV,FACTCON,FACTIN4,INTEGRL,VALUINT
C
      DIMENSION ARMISC(1),AR(MAXDIM,MAXJ,3),ARMUMN(NDIM,MDIM),
     1MUSE(MDIM),MAXN(MDIM)
      DIMENSION AV(11)
C
      DATA            EGNBNDO,ETAO,MDIMO,NDIMO,RNOFSVO,SIGNRBO/
     1         -1.,-1.,0,0,0.,0./
      DATA PI,TWOPI         /3.14159265358979,6.28318530717959/
C
      COMMON/CFACT/  M,N,RMUMN,CAPNMN,ETA,SIGN,L,CAPKMN
C
      EQUIVALENCE (AV(1),AVSPAN),              (AV(3),C2),
     1(AV(6),DCL),(AV(7),FMI),(AV(8),FME),(AV(9),FMZ)
C
      EXTERNAL FACTIN4
C

      ITRACE = ARMISC(6)
      IF(ITRACE .GE. 1)  WRITE(6,1010)
C
C              GENERATE THE EIGENVALUES
C
      NSBRB  = ARMISC(10)
      SIGMA  = ARMISC(14)
      SIGNRB = SIGMA*NSBRB
      MSBT   = ARMISC(7)
      RK    = SIGNRB*MSBT
      RKSQD= RK**2
      AXIALM = AR(2,9,2)
      CMACH = 1.-AXIALM**2
      EGNBND = RK/ SQRT(CMACH)
      RNOFSV = 1
      ETA = ARMISC(3)
      IF(ITRACE .GE. 1)  WRITE(6,1020) NSBRB,SIGMA,SIGNRB,MSBT,RK,RKSQD,
     1                                 AXIALM,CMACH,EGNBND,RNOFSV,ETA
      IF( EGNBND .NE. EGNBNDO )  GO TO 110
      IF( ETA    .NE. ETAO    )  GO TO 110
      IF( MDIM   .NE. MDIMO   )  GO TO 110
      IF( NDIM   .NE. NDIMO   )  GO TO 110
      IF( RNOFSV .NE. RNOFSVO )  GO TO 110
      IF( SIGNRB .NE. SIGNRBO )  GO TO 110
      IF(ITRACE .GE. 1)  WRITE(6,1030)
      GO TO 120
C
  110 CALL EGNVAL2( 2    ,EGNBND,ETA,MDIM,NDIM,RNOFSV,SIGNRB,ITRACE,
     1              NOFM,MUSE,MAXN,ARMUMN,IEREGNV)
```

```
      IERROR = IEREGNV
 120  CONTINUE
C
C             ERROR RETURN
C
      IF(IEREGNV.EQ.0) GO TO 200
      IF(IEREGNV-2) 150,130,150
  130 IF(ITRACE.NE.0) WRITE(6,140)
  140 FORMAT(//1H0,70(1H*)//1H0,*A REDUCED SET OF EIGENVALUES IS AVAILAB
     1LE*/1H0,*COMPUTATIONS WILL PROCEDE*/1H0,70(1H*) )
      GO TO 200
 150  IF(IEREGNV-4) 200,160,200
  160 IF(ITRACE.NE.0) WRITE(6,180)
  180 FORMAT(//1H0,70(1H*)//1H0,*THERE ARE NO PROPAGATING RADIAL MODES*/
     11H0,*NO COMPUTATIONS CAN BE MADE*/1H0,70(1H*) )
      GO TO 1000
  200 CONTINUE
C
C
C
C
      ZSBIR = ARMISC(17)
      IFLOW=ARMISC(4)
C
C
C             LOOP ON M
C
      DO 300 IOFM=1,NOFM
C
C             SET M,L, AND NOFN
C
      M = MUSE(IOFM)
      L = (M-SIGNRB)/RNOFSV
      NOFN = MAXN(IOFM)
      IF(IOFM.GT.1.AND.ITRACE.GE.1) WRITE(6,1005)
      IF(ITRACE .GE. 1)  WRITE(6,1040) M,L,      NSBRB,NOFN
C
C             LOOP ON N
C
      DO 700 N=1,NOFN
C
C             CALCULATE PROPAGATION FACTORS
C
      RMUMN = ARMUMN(N,IOFM)
      BETAMN = SQRT(RKSQD-CMACH*RMUMN**2)
      CAPKMN = (-RK*AXIALM + IFLOW*BETAMN)/CMACH
      CAPNMN = EGNNORM(M,RMUMN,ETA)
      IF(ITRACE .GE. 1)  WRITE(6,1050) N,RMUMN,BETAMN,CAPKMN,CAPNMN
C
C
C
C             COMPUTE MODAL AMPLITUDES
C
C
C
```

```
C
C                CALCULATE CONSTANT FACTOR, FACTCON
C
      TEMP1 = -CAPKMN*ZSBIR
      FACTCON = (-NSBRB*CMPLX(COS(TEMP1),SIN(TEMP1))) /
     1             (         4.*BETAMN)
C
C
C                SET THE NUMBER OF SUB-INTERVALS FOR WHICH THE
C                LAST TWO TERMS ARE EVALUATED
C
      IORDGS = 2
      INTEGRL = (0.,0.)
C
C
C                SET NUMBER OSCILLATIONS
C
      NOSCE  = 1
      NOSCE = MAXO(NOSCE,N)
      NOSCE = 1.5 * NOSCE
      NOSCE = MAXO(NOSCE,2)
      WIDTHI = (1.-ETA)/NOSCE
      IF(ITRACE .GE. 1)  WRITE(6,1060) FACTCON,NOSCE
C
C                LOOP ON NUMBER OF SUBINTERVALS
C
      DO 600 ITRVL=1,NOSCE
      RLOW  = ETA + (ITRVL-1)*WIDTHI
      RUP   = RLOW + WIDTHI
  270 CONTINUE
C
C                EVALUATE TERM TO BE AVERAGED
C
C                SET AVERAGE SPAN
C
      AVSPAN = (RLOW + RUP )*.5
      AV(1) = AVSPAN
C
C                SET K INDEX
C
      K = 2
C
C                SET AVERAGE VALUES
C
      DO 400 IAV=3,11
C
C                SET J INDEX
C
      J = IAV
      IF(IAV.EQ.3) J=2
      IF(IAV.EQ.10)J=9+ARMISC(18+K)+1
      IF(IAV.EQ.11)J=9+ARMISC(18+K)+2
      IF( J.EQ.4 ) GO TO 400
      IF( J.EQ.5 ) GO TO 400
```

```
C
C               SET K  INDEX  TO  BE  USED
C
      KU = K
C
C
C               SET SPAN WHEN J=1
C
      NSPN  = AR(1,J,KU)
C
C               AVERAGE VALUE IS INPUT
C
      IF( NSPN ) 400,330,340
  330 AV( IAV) = AR(2,J,KU)
      GO TO 400
C
C               INTERPOLATE FOR AVERAGE VALUE
C
  340 IPA=-1
      CALL MTLUP(AVSPAN,AV(IAV),1,NSPN,NSPN,1,IPA,AR(3,1,KU),AR(3,J,KU))
  400 CONTINUE
      IF(ITRACE .GE. 1)  WRITE(6,1070) RLOW,RUP,AVSPAN    ,C2    ,DCL,FMI,
     1                                 FME,FMZ,AV(10),AV(11)
C
C
C               CALCULATE THE AVERAGE FACTOR, FACTAV
C
      FACTAV =   . C2*DCL
      IF(ITRACE .GE. 1)  WRITE(6,1080) FACTAV
      TEMP1 = SQRT( FMI**2 -FMZ**2 )
      TEMP2 = SQRT( FME**2 -FMZ**2 )
      TEMP1 =0.25*(TEMP1+TEMP2)**2
      TEMP1 = SQRT( FMZ**2 + TEMP1 )
      FACTAV = FACTAV*FMZ
      IF(ITRACE .GE. 1)  WRITE(6,1090) TEMP1,TEMP2,FACTAV
      COSTHS = FMZ/TEMP1
      SINTHS = SQRT(1. - COSTHS**2)
      TEMP2  =         M*COSTHS/AVSPAN + CAPKMN*SINTHS
      FACTAV = FACTAV*TEMP2
      IF(ITRACE .GE. 1)  WRITE(6,1100) FACTAV,TEMP2
C
C
C
C               PERFORM GAUSSIAN INTEGRATION
C
      CALL  GAUSS2(RLOW,RUP, IORDGS,VALUINT,FACTIN4,ARMISC,MAXDIM,MAXJ,AR)
C
C               ACCUMULATE THE TERMS
C
      INTEGRL = INTEGRL + FACTAV*VALUINT
C
C
C               END INTERVAL LOOP
C
      IF(ITRACE.GE.1)  WRITE(6,1120)FACTAV,VALUINT,INTEGRL
```

```
  600 CONTINUE
C
C              APPLY FIRST TERM AND STORE
C
      INTEGRL = FACTCON*INTEGRL
      ALPHAMN(N,IOFM)=INTEGRL
      IF(ITRACE .GE. 1)  WRITE(6,1130) INTEGRL
C
C              END N AND M LOOPS
C
  700 CONTINUE
  800 CONTINUE
 1000 CONTINUE
C
C              SAVE THE EIGENVALUE DETERMINING PARAMETERS
C
      EGNBNDO = EGNBND
      ETAO    = ETA
      MDIMO   = MDIM
      NDIMO   = NDIM
      RNOFSVO = RNOFSV
      SIGNRBO = SIGNRB
C
      RETURN
C
 1005 FORMAT(1H1//)
 1010 FORMAT(1H1//11X,* OPTIONAL PRINTOUT FROM SUBROUTINE BBCAA*)
 1020 FORMAT(1HO,10X,*EIGENVALUE PARAMETERS GENERATED*/13X,*NSBRB = *,
     1I3,2X,*SIGMA = *,F3.0,2X,*SIGNRB = *,F6.0,2X,*MSBT = *,F10.4/13X,
     2*RK = *,F10.4,2X,*RKSQD = *,F10.4,2X,*AXIALM = *,F10.4/13X,
     3*CMACH = *,F10.4,2X,*EGNBND = *,2X,F10.5,2X,*RNOFSV = *,F10.4/13X,
     4*ETA = *,F10.5)
 1030 FORMAT(1HO,10X,*THE EIGENVALUES FROM PREVIOUS CALL TO SUBROUTINE B
     1BCAA*/11X,*ARE REUSED FOR THIS CASE*)
 1040 FORMAT(1HO,10X,*M = *,I4,2X,*L = *,I4,2X,
     1*NSBRB = *,I3,2X,*NOFN = *,I4)
 1050 FORMAT(1H ,10X,*N = *,I4,2X,*RMUMN = *,F10.4,2X,*BETAMN = *,F10.4/
     120X,*CAPKMN = *,F10.4,2X,*CAPNMN = *,F10.4)
 1060 FORMAT(1H ,10X,*FACTCON = *,2E12.4,2X,*NOSCE = *,I4)
 1070 FORMAT(1H ,10X,*RLOW = *,F9.4,2X,*RUP = *,F9.4,2X,*AVSPAN = *,F9.4
     1/11X,*C2 = *,F10.4,2X,                  *DCL = *,F10.4/11X,
     2*FMI = *F9.4,2X,*FME = *,F9.4,2X,*FMZ = *,F9.4/
     311X,*AV(10) = *,F9.4,2X,*AV(11) = *,F9.4)
 1080 FORMAT(1H ,10X,*FACTAV = *,2E12.4)
 1090 FORMAT(1H ,10X,*TEMP1 = *,F10.4,2X,*TEMP2 = *,F10.4,5X,*FACTAV = *
     1       ,2E12.4)
 1100 FORMAT(1H ,10X,*FACTAV = *,2E12.4,2X,*TEMP2 = *,2E12.4)
 1120 FORMAT(1H ,10X,*FACTAV =*,2E12.4,2X,*VALUINT =*,2E12.4/
     111X,*INTEGRL =*2E12.4)
 1130 FORMAT(1H ,10X,*INTEGRL =*,2E12.4)
      END
```

3.2 Secondary Special-Purpose Subprogram Descriptions

3.2.1 Subroutine EGNVAL2

Purpose: This subroutine computes the double subscripted array of hardwall, annular duct eigenvalues required by the modal representation of the acoustic pressure in such a duct. The first subscript is referred to as the spinning mode index, while the second is referred to as the radial mode index. For each member of a set of spinning mode indexes, $m = m_1, m_2, \ldots$, determined by the cutoff criterion (equation (4) of appendix I, volume I), the eigenvalues are the ordered set of zeros of the transcendental function:

$$F(x) = J_m'(x) - Y_m'(x)\,\frac{J_m'(\eta x)}{Y_m'(\eta x)}$$

solved for by the subroutine ZEROS, i.e.,

$$x = \mu_{mn}, \quad n = 1, 2, \ldots$$

with J_m and Y_m the Bessel and Neumann functions, respectively; the primes denoting differentiation with respect to the argument; and η denoting the hub-to-tip ratio.

Method: The procedure is as follows:

1) Establish the spinning mode index having the largest absolute value, m_{max}, from the inequality (see equation (4), appendix I, of volume I):

$$|m_{max}| \leq \mu_{m_{max},n} \leq E_B$$

where E_B is EGNBND (see the FORTRAN dictionary, sec. 2.2).

This inequality is satisfied by the integer which is less than or equal to the real number E_B.

2) Restrict the above bound, m_{max}, to be at most 100 (based on the restriction on the Bessel function evaluation BSSLS, sec. 3.3.6).

3) Calculate the smallest negative ℓ, ℓ_{lower}, according to the above bound, which is derived as follows:

Since $$m = \sigma N_R + \ell N_S$$

$$\ell = \left| \frac{m - \sigma N_R}{N_S} \right|$$

$$\leq \frac{|m| + \sigma N_R}{N_S} ,$$

then $$|\ell| \leq \frac{E_B + \sigma N_R}{N_S}$$

and $$\ell_{lower} = - \frac{E_B + \sigma N_R}{N_S}$$

4) Determine all m's according to the above equation defining m and within the bounds on m and ℓ given above by starting with the lowest ℓ and stepping through the ℓ's, calculating the m's, and storing those m's within the established bounds.

5) Set an error counter in the case that either the list of m's is not exhausted or no m's were obtained, continuing only in the former case.

6) Calculate an upper bound, n_{max}, on the radial mode index n derived as follows. From reference 30, formula (9.5.31) (see also APROX1, sec. 3.3.1), the eigenvalues are ultimately spaced by $\pi/1-\eta$. The bound used is

$$n_{max} = \left(\frac{1-\eta}{\pi}\right)(m_{max} + 1)$$

7) Calculate the eigenvalues for the m's determined above and n = 1 to n_{max} for each m.

8) Restrict the eigenvalues according to the bound $\mu_{mn} \leq E_B$, counting the number of eigenvalues within the bound, if any, for each m.

9) Eliminate any m for which there are no eigenvalues less than the bound, updating the stored arrays of m's, n's, and μ_{mn}'s.

Usage: CALLING SEQUENCE

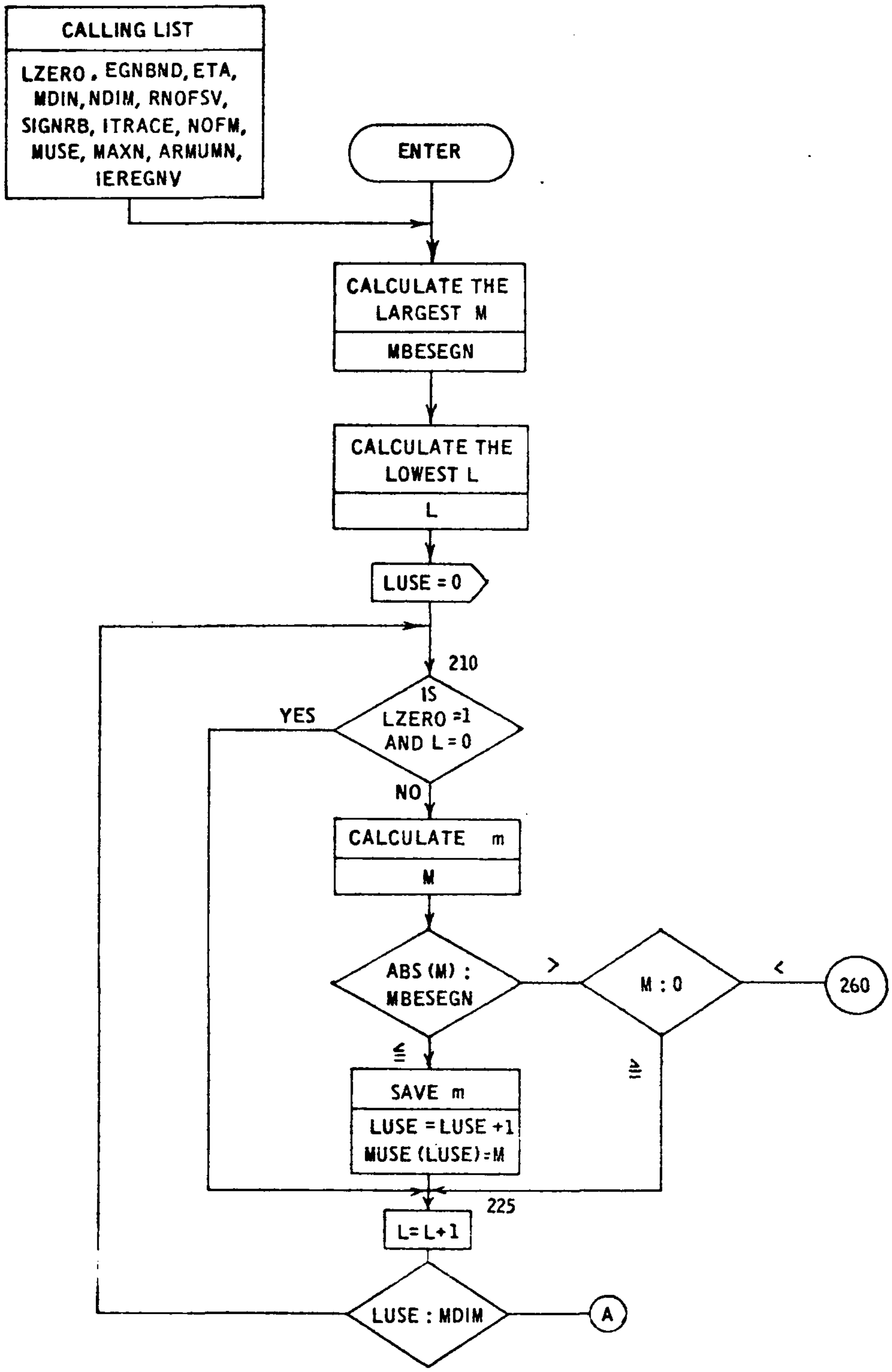
```
DIMENSION MUSE(MDIM),MAXN(MDIM),ARUMN(NDIM,MDIM)
.
.
.
CALL EGNVAL2(LZERO,EGNBND,ETA,MDIM,NDIM,RNOFSV,SIGNRB,
*  ITRACE,NOFM,MUSE,MAXN,ARMUMN,IEREGNV)
```

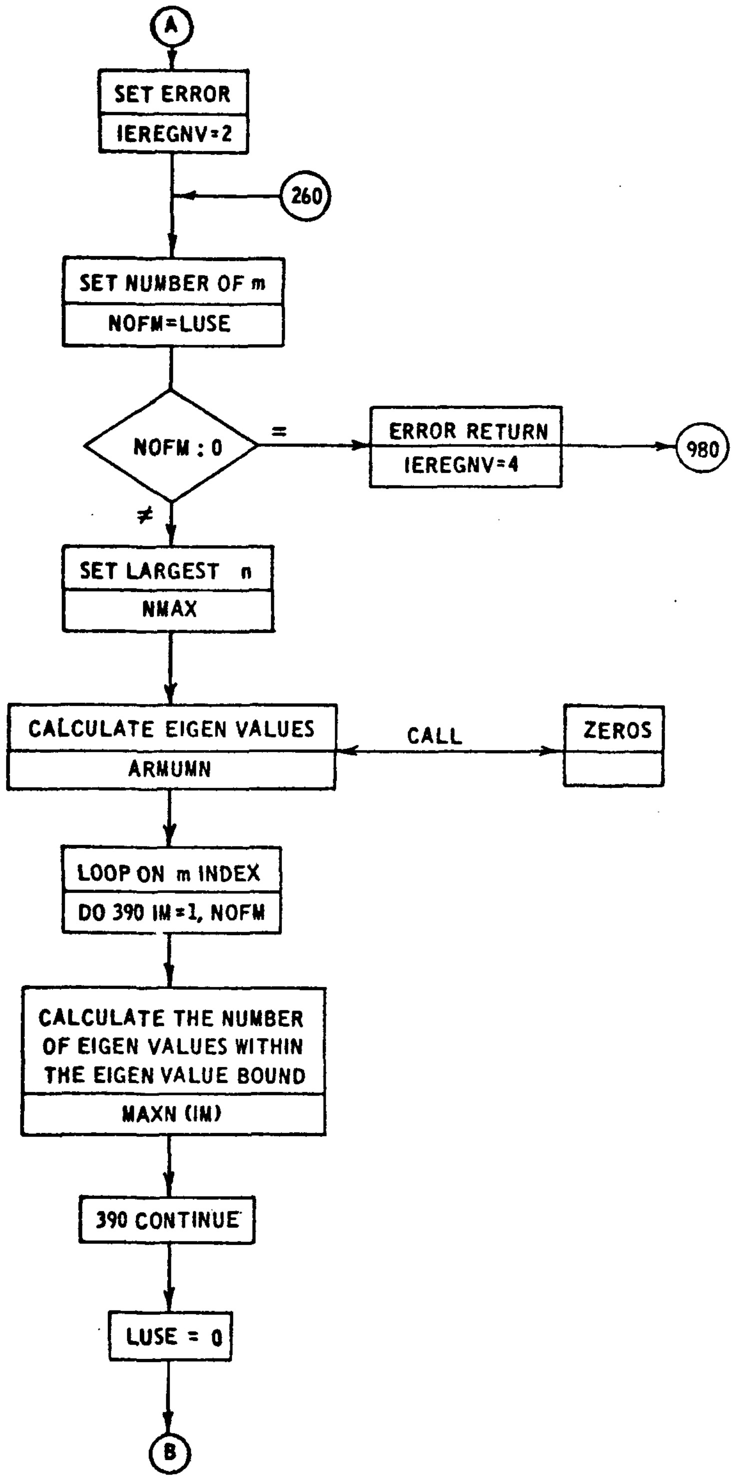
Error Return: IEREGNV

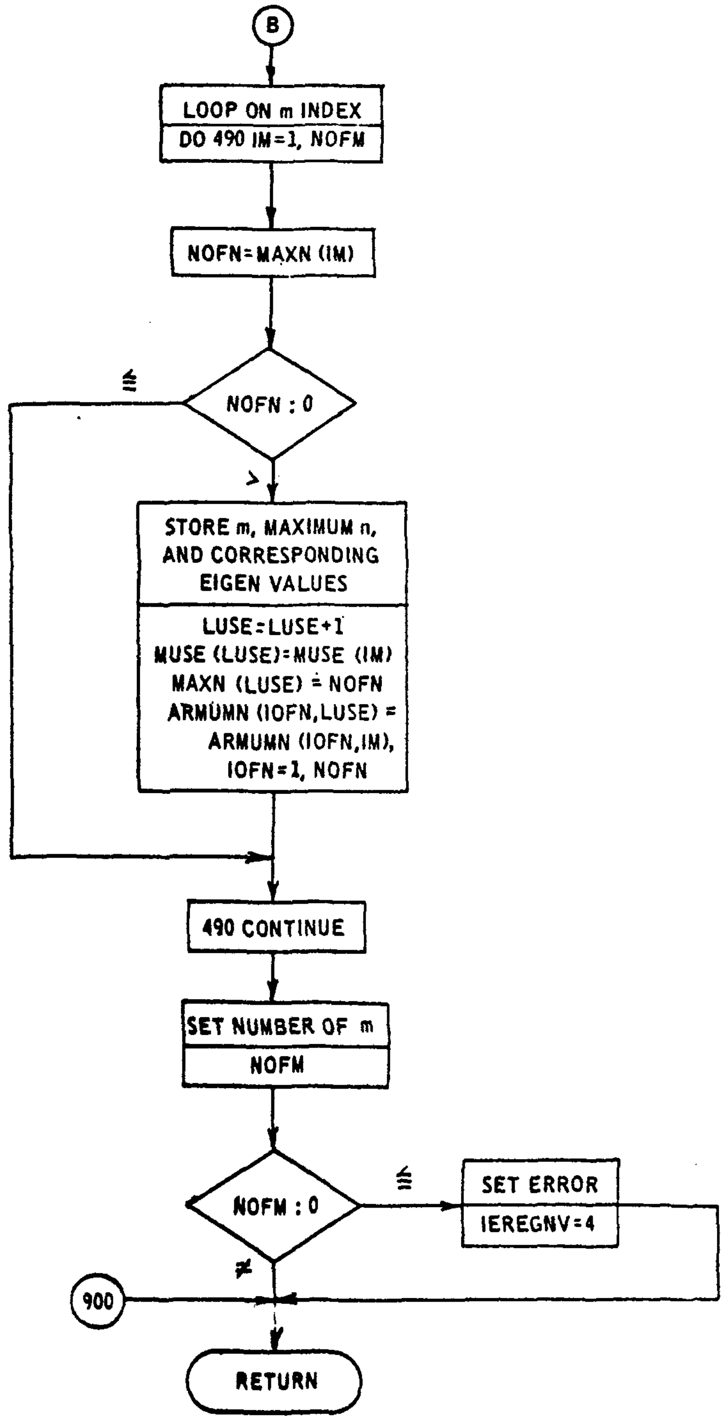
Timing: The timing is dominated by the eigenvalue calculation, subroutine ZEROS (sec. 3.2.3). According to sample runs, the time is

2 x EGNBND x (1. - ETA) + 2

Accuracy: See subroutine ZEROS for the accuracy of the eigenvalues.


CALLING LIST
LZERO, EGNBND, ETA, MDIN, NDIM, RNOFSV, SIGNRB, ITRACE, NOFM, MUSE, MAXN, ARMUMN, IEREGNV
ENTER
CALCULATE THE LARGEST M
MBESEGN
CALCULATE THE LOWEST L
L
LUSE = 0
210
IS LZERO =1 AND L = 0
YES
NO
CALCULATE m
M
ABS (M) : MBESEGN
>
M : 0
<
260
≦
≧
SAVE m
LUSE = LUSE +1
MUSE (LUSE) = M
225
L = L+1
LUSE : MDIM
A


A
SET ERROR
IEREGNV=2
260
SET NUMBER OF m
NOFM=LUSE
NOFM : 0
=
ERROR RETURN
IEREGNV=4
980
≠
SET LARGEST n
NMAX
CALCULATE EIGEN VALUES
ARMUMN
CALL
ZEROS
LOOP ON m INDEX
DO 390 IM=1, NOFM
CALCULATE THE NUMBER OF EIGEN VALUES WITHIN THE EIGEN VALUE BOUND
MAXN (IM)
390 CONTINUE
LUSE = 0
B



B
LOOP ON m INDEX
DO 490 IM=1, NOFM
NOFN=MAXN (IM)
≦
NOFN : 0
>
STORE m, MAXIMUM n, AND CORRESPONDING EIGEN VALUES
LUSE=LUSE+1
MUSE (LUSE)=MUSE (IM)
MAXN (LUSE) = NOFN
ARMUMN (IOFN,LUSE) = ARMUMN (IOFN,IM), IOFN=1, NOFN
490 CONTINUE
SET NUMBER OF m
NOFM
NOFM : 0
≦
SET ERROR
IEREGNV=4
≠
900
RETURN

```
      SUBROUTINE EGNVALZ(LZERO ,EGNBND,ETA,MDIM,NDIM,RNOFSV,SIGNRB,
     1ITRACE,NOFM,MUSE,MAXN,ARMUMN,IEREGNV)
C
C
CPURPOSE       OBTAIN THE APPROPRIATE SPINNING AND RADIAL MODES.
C
C
C
C
CINPUT          VARIABLE  DEFINITION
C                EGNBND    ALL EIGENVALUES ARE FOUND WHICH DO NOT EXCEED
C                           THIS NUMBER.
C                  ETA     HUB TO TIP RATIO, WHICH IS ZERO FOR CIRCULAR
C                           DUCT
C                 MDIM     COLUMN DIMENSION OF MATRIX IN WHICH EIGEN-
C                           VALUES ARE PLACED, PROVIDENG A MAXIMUM ON
C                           THE NUMBER OF SPINNING MODES.
C                 NDIM     ROW DIMENSION OF MATRIX IN WHICH EIGENVALUES
C                           ARE PLACED, PROVIDING A MAXIMUM ON THE
C                           NUMBER OF RADIAL EIGEN VALUES.
C               RNOFSV     NUMBER OF STATOR VANES
C               SIGNRB     PRODUCT OF HARMONIC INDEX,SIGMA,
C                           AND THE NUMBER OF ROTOR BLADES,NSBRB
C
COUTPUT         ARMUMN     MATRIX OF EIGENVALUES WHERE
C                                    MUMN(IOFM,IOFN)
C                           IS THE EIGENVALUE FOR SPINNING MODE INDEX
C                           IOFM AND RADIAL MODE INDEX IOFN WHERE
C                           IOFN=1,...,MAXN(IOFM)
C                 MAXN     ARRAY OF THE NUMBER OF RADIAL MODES
C                                              WHERE MAXN(IOFM)
C                          CORRESPONDS TO MUSE(IOFM), IOFM=1,...,NOFM
C                 MUSE     ARRAY OF THE NOFM SPINNING MODE INDICES
C                 NOFM     NUMBER OF SPINNING MODE INDICES
C
CERROR RETURN  IEREGNV     = 0 ALL EIGENVALUES REQUIRED ARE RETURNED
C                            2 THERE ARE MORE EIGENVALUES REQUIRED THAN
C                               THERE IS SPACE FOR, AS MANY AS POSSIBLE
C                               ARE RETURNED
C                            4 THERE ARE NO EIGENVALUES
C
C
C
      DIMENSION MUSE(MDIM),MAXN(MDIM),ARMUMN(NDIM,MDIM)
      DIMENSION SC(40)
      DATA MBES/100/
C
C
C
      IEREGNV = 0
C
C               SET THE MAXIMUM SPINNING MODE THAT CAN POSSIBLY PROPAGATE
C
```

```
      MEGN = EGNBND + 1
      MBESEGN = MINO(MEGN,MBES)
C
C
C              NOW COMPUTE CANDIDATE SPINNING MODES ACCORDING TO
C
C
C              SET THE LOWEST L THAT IS USABLE
C
      L = -ABS( ( EGNBND    +SIGNRB)/RNOFSV )
      LUSE = 0
  210 IF( LZERO .EQ.1 .AND.L.EQ.0 ) GO TO 225
      M = SIGNRB +L*RNOFSV
      IABSM = IABS( M )
      IF( IABSM - MBESEGN ) 220,220,215
  215 IF(M) 225,225,260
  220 LUSE = LUSE + 1
      MUSE(LUSE) = M
  225  L = L + 1
      IF( LUSE - MDIM ) 210,250,250
  250 IEREGNV = 2
  260 NOFM = LUSE
C
  300 CONTINUE
C
C              CHECK TO BE SURE THERE ARE SPINNING MODES
C
      IF( NOFM ) 310,310,320
  310 IEREGNV=4
      GO TO 900
  320 CONTINUE
C
C              OBTAIN THE EIGENVALUES
C
C
C              SET BOUNDS FOR EIGENVALUE CALCULATION
C
      NBESEGN = (1.-ETA)*(MBES+1.)/3.14159265 + 1.
      NMAX = MINO( NBESEGN,    NDIM-1)
C
C
C
      CALL  ZEROS(ETA,NOFM,MUSE,NMAX,NDIM,ITRACE,SC,ARMUMN)
C
C              COMPUTE THE NUMBER OF RADIAL MODES FOR EACH SPINNING MODE
C
  330 DO  390 IM=1,NOFM
      M = MUSE(IM)
      MP1 = IABS(M) + 1
      N = 1
  340 IF( ARMUMN(N,IM ) - EGNBND ) 350,350,370
  350 N = N + 1
```

```
      IF( N - NMAX    ) 340,340,360
  360 IEREGNV = 2
  370 NM1 = N - 1
      MAXN(IM) = NM1
      IF( NM1 - NDIM ) 390,390,380
  380 IEREGNV = 2
      MAXN(IM) = NDIM
  390 CONTINUE
C
  400 CONTINUE
C
C                 ELIMINATE SPINNING MODES FOR WHICH THERE ARE NO
C                 RADIAL MODES
C
      LUSE = 0
      DO 490 IM=1,NOFM
      NOFN=MAXN(IM)
      IF( NOFN      ) 490,490,410
  410 LUSE = LUSE + 1
      MUSE(LUSE) = MUSE(IM)
      MAXN(LUSE) = NOFN
      DO 470 IOFN=1,NOFN
  470 ARMUMN(IOFN,LUSE) = ARMUMN(IOFN,IM)
  490 CONTINUE
      NOFM = LUSE
      IF( NOFM ) 500,500,1000
  500 IEREGNV = 4
C
C                 ERROR RETURN AT THIS POINT
C
  900 CONTINUE
C
C
 1000 RETURN
C
C
      END
```

3.2.2 Subroutine ZEROS

Purpose: This subroutine computes the first NMAX zeros, in increasing order starting with the lowest, of the function:

$$F(x) = J'_m(x) - Y'_m(x) \frac{J'_m(\eta x)}{Y'_m(\eta x)}$$

for each $m = \{m_1, m_2, \ldots\}$.

For $m = 0$, the first zero is $x = 0$; all other zeros are nonzero positive and equal to the zeros of the function:

$$G(x) = J'_m(x) Y'_m(\eta x) - Y'_m(x) J'_m(\eta x)$$

with J_m and Y_m the Bessel and Neumann functions, respectively; the primes denoting differentiation with respect to the argument; and η denoting the hub-to-tip ratio. The zeros of G(x) are computed for $m = 0, 1, \ldots,$ MMAX, the largest input m in absolute value, using subroutine JARRATT (sec. 3.3.3) with selected iteration starting values. The zeros corresponding to $\{m_1, m_2, \ldots\}$ are saved as computed with zeros corresponding to $m < 0$ being the same as $-m$.

Method: The procedure is as follows:

1) Set the tolerances and iteration limit for subroutine JARRATT (see sec. 3.3.3) used in steps 6 and 13 below.

2) Set the largest $|m|$ input, MMAX.

3) Set the first zero of F(x) to zero when $m = 0$.

4) Set the index of the i^{th} zero of F(x) for m = 0, i = 2, ..., NMAX by a DO loop.

5) For m = 0, calculate the three iteration starting values for the i^{th} zero of F(x). The first starting value is computed by subroutine APROX1 (sec. 3.3.1) for $\eta \geq .2$, and by subroutine APROX2 (sec. 3.3.2) for $\eta < .2$. The second and third starting values are the first +.1 and -.1, respectively.

6) Calculate the i^{th} zero of F(x) by solving equation G(x) = 0 using subroutine JARRATT with the values set in steps 1 and 5.

7) When m = 0 is input, save the zeros calculated in steps 3 to 6 in an output array.

8) Return if only m = 0 is input.

9) Reset the first zero for m = 0 to 1 for use in step 12.

10) Set the value of m, m = 1, 2, ..., MMAX by a DO loop.

11) Set the index of the i^{th} zero of F(x) for the m in step 9 by a DO loop.

12) Calculate the three iteration starting values for the i^{th} zero of F(x). The first value is the i^{th} zero for the previous m. For $\eta > 0$, the second and third starting values are the first -.1 and +.1; for $\eta = 0$, the values are the first +.1 and +.2.

13) Calculate the i^{th} zero of F(x) by solving the equation G(x) = 0 using subroutine JARRATT with the values set in steps 1 and 12.

14) When $|m|$, m set in step 10 is input, except for the zeros computed in steps 11 to 13 in an output array.

Usage: CALLING SEQUENCE

```
DIMENSION MUSE(MDIM),SC(40),ARMUMN(NDIM,MDIM)
.
.
.
CALL ZEROS(ETA,NOFM,MUSE,NMAX,NDIM,ITRACE,SC,ARMUMN)
```

Printout and Diagnostics: The zero, the corresponding function value, the starting guess (GUESS[1]), and the error return code IERJAR from subroutine JARRATT (see sec. 3.3.3) can be printed as calculated according to the input ITRACE (see the FORTRAN dictionary, sec. 2.2).

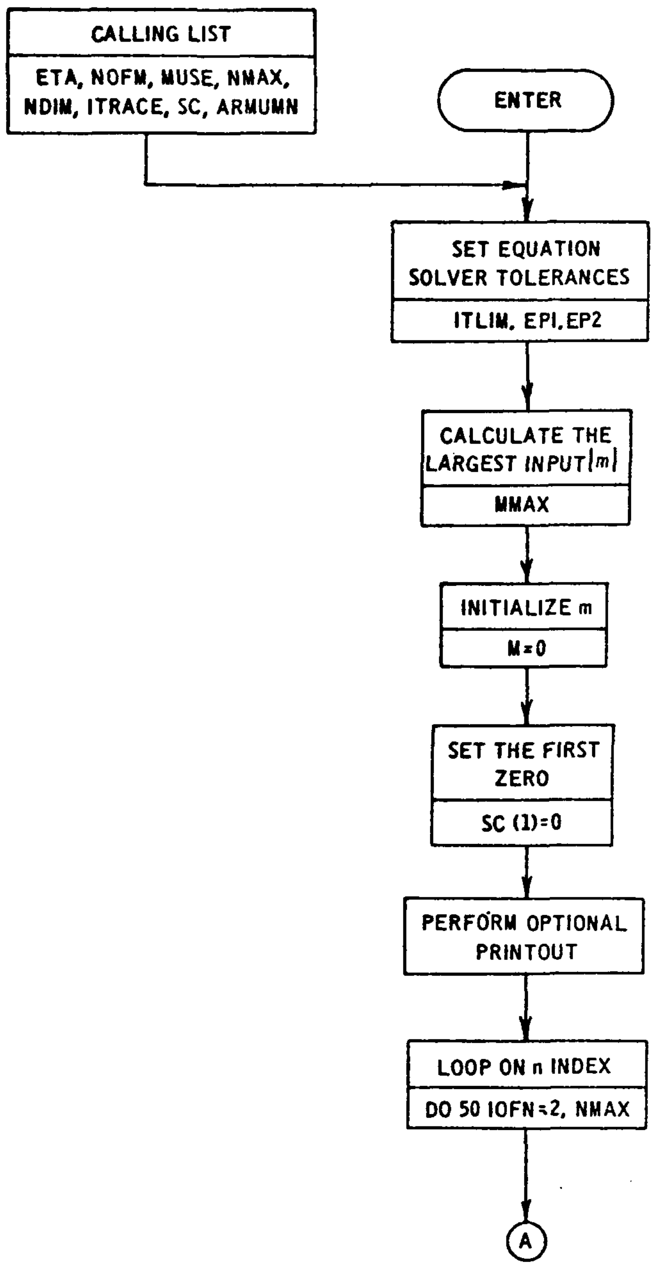
Restrictions: MUSE(1) or MUSE(NOFM) must be the largest m in absolute value; NMAX $\leq$ NDIM.

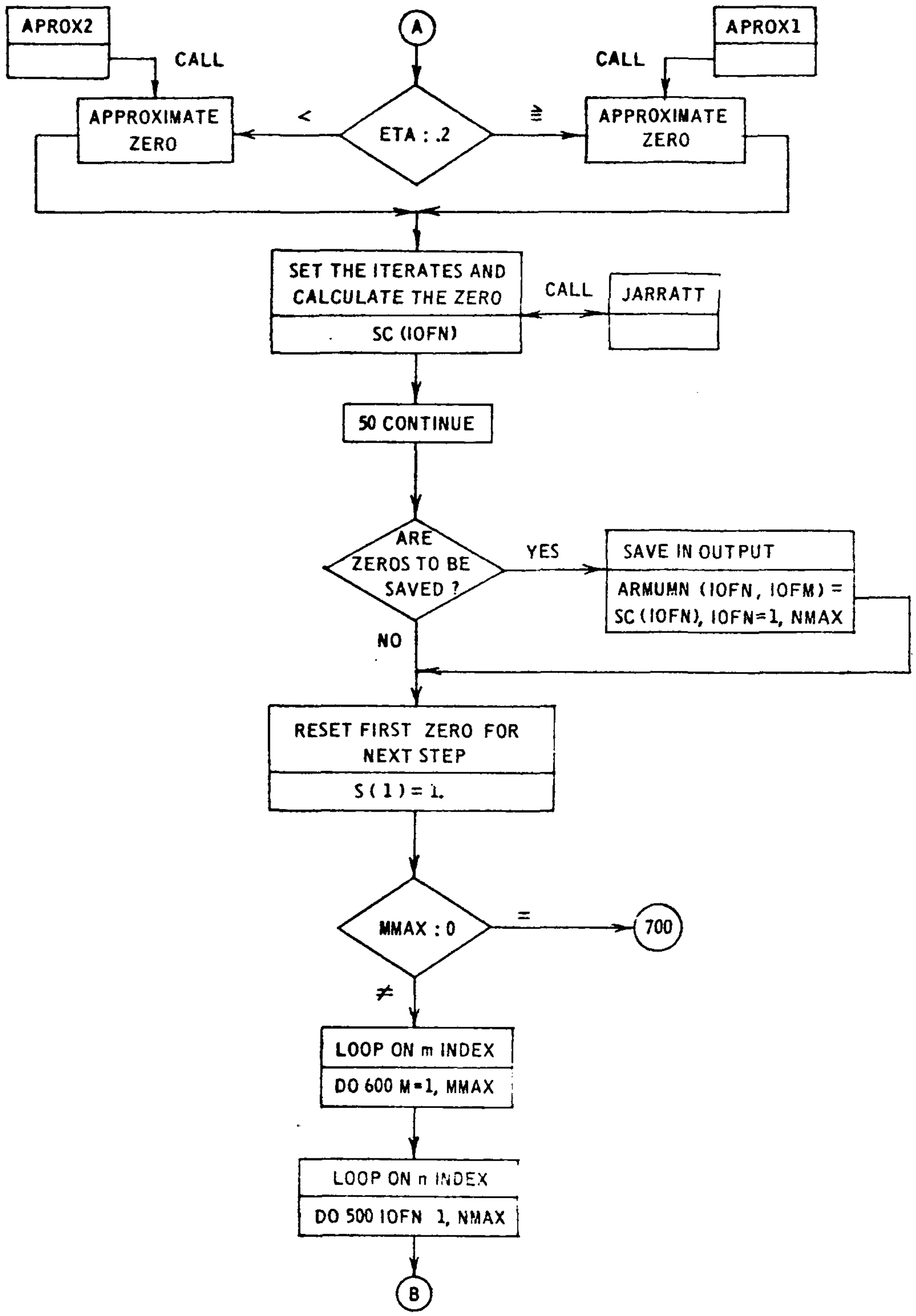
Timing: The timing is proportional to the nearest integer to

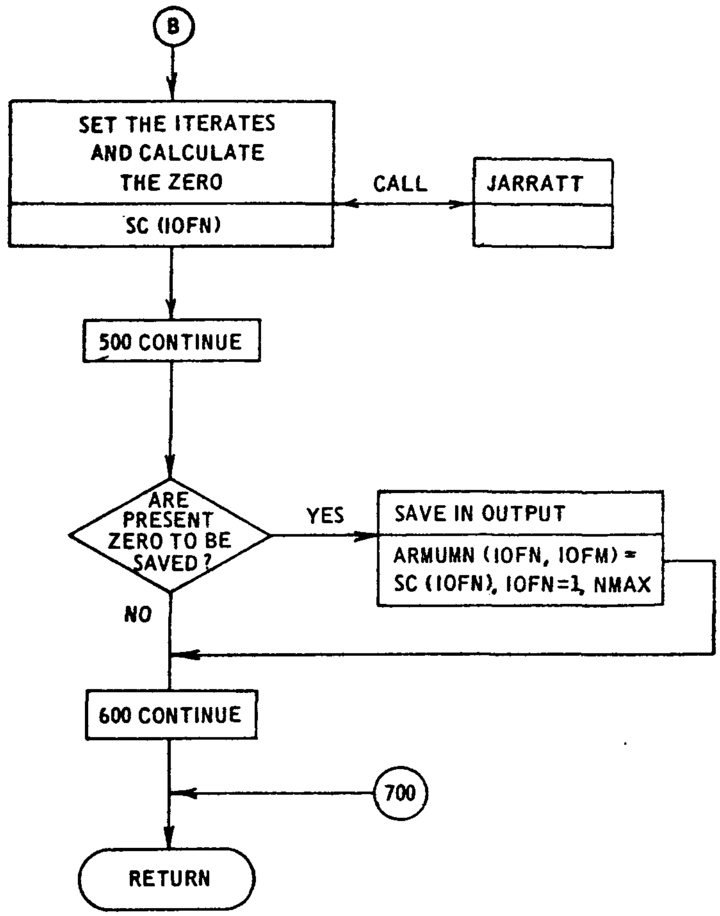
$$\frac{2(\text{EGNBND} + 1)(1 - \text{ETA})}{3.14}$$

times a unit call to subroutine JARRATT.

Accuracy: The accuracy is of the algorithmic type and, in particular, is dominated by the Bessel function evaluators BSSLS (sec.3.3.6 and BF4F [ref. 41]). The zeros are calculated by subroutine JARRATT with given starting values so that the cross product of Bessel functions (see subroutine EQATION) is less than 10^{-10}.


CALLING LIST
ETA, NOFM, MUSE, NMAX, NDIM, ITRACE, SC, ARMUMN
ENTER
SET EQUATION SOLVER TOLERANCES
ITLIM, EPI, EP2
CALCULATE THE LARGEST INPUT |m|
MMAX
INITIALIZE m
M=0
SET THE FIRST ZERO
SC (1)=0
PERFORM OPTIONAL PRINTOUT
LOOP ON n INDEX
DO 50 IOFN=2, NMAX
A


A
APROX2
CALL
APROX1
CALL
APPROXIMATE ZERO
<
ETA : .2
≧
APPROXIMATE ZERO
SET THE ITERATES AND CALCULATE THE ZERO
SC (IOFN)
CALL
JARRATT
50 CONTINUE
ARE ZEROS TO BE SAVED ?
YES
SAVE IN OUTPUT
ARMUMN (IOFN, IOFM) = SC (IOFN), IOFN=1, NMAX
NO
RESET FIRST ZERO FOR NEXT STEP
S(1) = 1.
MMAX : 0
=
700
≠
LOOP ON m INDEX
DO 600 M=1, MMAX
LOOP ON n INDEX
DO 500 IOFN 1, NMAX
B


B
SET THE ITERATES AND CALCULATE THE ZERO
SC (IOFN)
CALL
JARRATT
500 CONTINUE
ARE PRESENT ZERO TO BE SAVED?
YES
SAVE IN OUTPUT
ARMUMN (IOFN, IOFM) = SC (IOFN), IOFN=1, NMAX
NO
600 CONTINUE
700
RETURN

```
      SUBROUTINE ZEROS(ETA,NOFM,MUSE,NMAX,NDIM,ITRACE,SC,ARMUMN)
C
C PURPOSE         COMPUTE THE ZEROS OF THE EQUATION
C
C                  JP(M,X)*YP(M,ETA*X) - YP(M,X)*JP(M,ETA*X)
C
C                 WHERE JP AND YP ARE THE DERIVATIVES OF THE BESSEL
C                 FUNCTIONS OF THE FIRST AND SECOND KINDS, RESPECTIVELY,
C                 OF ORDER M AND ARGUMENT X OR ETA*X.
C
C METHOD          THE EQUATION IS TO BE SOLVED FOR THE FIRST M ZEROS
C                 FOR EACH ORDER, THE ORDER M = 0,1,2,...,MMAX. FOR THIS
C                 PURPOSE THE ZEROS FOR M = 0 ARE FOUND BY FIRST APPLYING
C                 AN APPROXIMATION FORMULA THEN A  REFINEMENT PROCEDURE
C                 USING BESSE FUNCTION EVALUATORS. THE ZEROS FOR HIGHER
C                 ORDERS ARE FOUND BY STEPPING THROUGH ORDER USING A
C                 NONLINEAR EQUATION SOLVER AND BESSEL EVALUATORS ( AS THE
C                 REFINEMENT ) WITH STARTING VALUES BEING THE ZEROS FOR
C                 THE PREVIOUS ORDER.
C
C
C SUBPROGRAMS     JARRATT                    NONLINEAR EQUATION SOLVER
C
C                 APROX1   APPROXIMATION TO EQUATION FOR ETA AT LEAST .2
C
C                 APROX2   COMBINATION OF APPROXIMATION AND INTERPOLATION
C                          FOR ETA LESS THAN .2
C
C EXTERNALS       EQATION    EVALUATES THE EQUATION
C
      DIMENSION GUESS(3),ARMUMN(NDIM,1),SC(1),MUSE(1)
      EXTERNAL EQATION
C
C                 THIS COMMON PASSES M AND ETA TO EQUATION EVALUATOR
C
      COMMON/CEQUAT/M,CETA
C
C                 EQUTION SOLVER TOLERANCES
C
      ITLIM=30
      EP1=0.
      EP2=1.E-10
C
      CETA = ETA
      MMAX = MAX0( IABS(MUSE(1)),IABS(MUSE(NOFM)) )
      MP1MAX= MMAX+1
C
C                 SOLVE THE EQUATION FOR ORDER ZERO
C
      M=0
```

```
      GUESS(1) =   SC(IOFN)
      GUESS(2)=GUESS(1)-.1
      IF(ETA.EQ.0.) GUESS(2)=GUESS(1 )+.2
      GUESS(3)=GUESS(1)+.1
      CALL  JARRATT(GUESS,ITLIM,EP1,EP2,EQATION,ZERO,FT,IERJAR)
      IF(ITRACE .EQ. 2)  WRITE(6,20) ZERO,FT,GUESS(1),IERJAR
      SC(IOFN) = ZERO
  500 CONTINUE
      DO 540 IOFM=1,NOFM
      MSAVE= IABS( MUSE(IOFM))
      IF(M-MSAVE) 540,510,540
  510 DO 530 IOFN=1,NMAX
  530 ARMJMN(IOFN,IOFM ) = SC(IOFN)
  540 CONTINUE

  600 CONTINUE

  700 RETURN
      END
```

3.2.3 Function EQATION

Purpose: This function evaluates the cross-product expression:

$$J_m'(x)\ Y_m'(\eta x) - Y_m'(x)\ J_m'(\eta x)$$

with the prime denoting differentiation with respect to the argument. J_m and Y_m denote, respectively, the Bessel and Neumann functions of integer order and real argument. The hub-to-tip ratio is given by η.

Method: The procedure is as follows:

1) Evaluate $J'_m(x)$ using the recursion relationship (ref. 30):

$$J_m'(x) = -J_{m+1}(x) + \frac{m}{x} J_m(x)$$

2) If $\eta = 0$, the cross product is $J_m'(x)$.

3) Evaluate $Y_m'(x)$, $J_m'(\eta x)$, and $Y_m'(\eta x)$, using the recursion relationship for the derivatives, as referenced above.

4) Evaluate the cross product.

Usage: CALLING SEQUENCE

```
COMMON/CEQAT/M,ETA
COMMON/SCRATCH/BES(1000)
.
.
.
CRSPRD = EQATION (X)
```

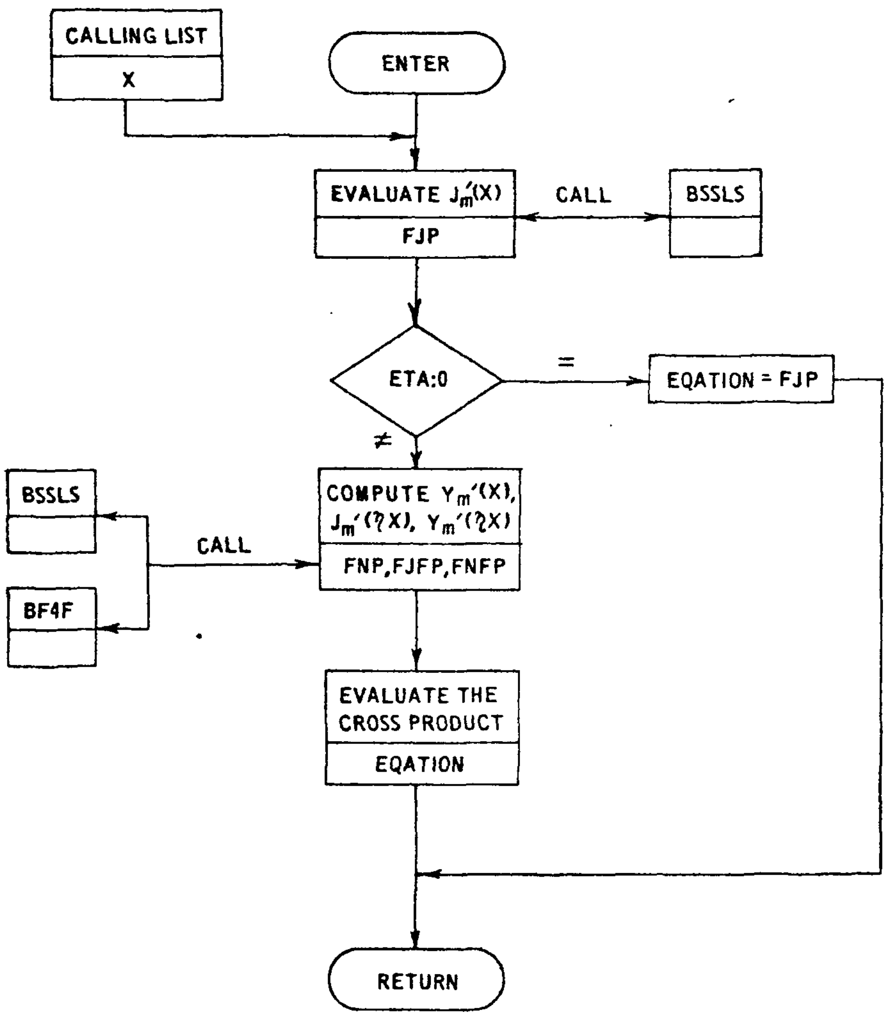
Restrictions: $3X + M + 12 \leq 1000$ (see BSSLS)

$M \geq 0$

$0 \leq \eta < 1$

Timing: The timing is dominated by the Bessel function evaluation and is approximately equal to twice the sum of the unit time for a call to BF4F (ref. 41) and BSSLS (sec. 3.3.6).

Accuracy: The accuracy is of the algorithmic type and, in particular, is dominated by subroutines BF4F and BSSLS.


CALLING LIST
X
ENTER
EVALUATE $J_m'(X)$
FJP
CALL
BSSLS
ETA:0
=
EQATION = FJP
≠
BSSLS
CALL
COMPUTE $Y_m'(X)$, $J_m'(?X)$, $Y_m'(?X)$
FNP,FJFP,FNFP
BF4F
EVALUATE THE CROSS PRODUCT
EQATION
RETURN

```
      FUNCTION EQATION(X)
C
C
CPURPOSE              EVALUATE THE FOLLOWING EQUATION USING BESSEL
C                       FUNCTION EVALUATORS
C
C                       JP(M,X)*YP(M,ETA*X) - YP(M,X)*JP(M,ETA*X)
C
C                       WHERE JP AND YP ARE THE DERIVATIVES OF THE BESSEL
C                       FUNCTIONS OF FIRST AND SECOND KIND, RESPECTIVELY,
C                       FOR INTEGER ORDER M AND REAL ARGUMENT X AND ETA*X,
C                       ETA A GIVEN PARAMETER.
C
CINPUT          VARIABLE  DEFINITION
C                     X     REAL ARGUMENT X
C                     M     INTEGER ORDER M, OBTAINED FROM COMMON
C                    ETA     PARAMETER ETA, OBTAINED FROM COMMON
C
CNOTATION       VARIABLE  DEFINITION
C                  FNFP     YP(M,ETA*X)
C                  FJFP     JP(M,ETA*X)
C                   FJP     JP(M,X)
C                   FNP     YP(M,X)
C
CSUBPROGRAMS     LRC LIBRARY ROUTINES BSSLS AND BF4F  THAT EVALUATE
C                BESSEL FUNCTIONS OF FIRST AND SECOND KIND, RESPECTIVELY.
C
CRESTRICTIONS  3*X + M + 12 CAN BE AT MOST 1000
C
      COMMON/CEQUAT/M,ETA
      COMMON/SCRATCH/BES(1000)
      DATA ISIGN/-1/
C
      MP1=M+1
      MP2=M+2
C
      CALL BSSLS(X,BES,MP1,IERR)
      A1=BES(MP1)
      A2=BES(MP2)
      FJP=-A2+(M/X)*A1
C
      IF(ETA.NE.0.) GO TO 10
      EQATION = FJP
      GO TO 30
C
   10 CALL  BF4F(X,BES,MP1,IERR,ISIGN)
      A3= BES(MP1)
      A4= BES(MP2)
      FNP= -A4 + (M/X)*A3
C
      Y = X*ETA
      CALL BSSLS(Y,BES,MP1,IERR)
```

```
      MP1=1
      RM=M
      SC(1) = 0.
C
C              TRACE FORMATS
C
      IF(ITRACE.EQ.2) WRITE(6,5)ETA
    5 FORMAT(    1H1,*TRACE THE CALCULATION FOR THE ZEROS OF THE ANNULAR
     1 EIGENFUNCTION*/1H0,2X,*THE RATIO OF THE INNER TO OUTER RADIUS IS*
     2,F10.5)
      IF(ITRACE.EQ.2) WRITE(6,10)M
   10 FORMAT(1H0,* THE ORDER M = *,I4/1H0,18X,*ZERO*, 8X,*FUNCTION VALUE
     1*, 8X,*STARTING GUESS*,2X,*CONVERGE*)
      IF(ITRACE.EQ.2) WRITE(6,15)
   15 FORMAT(1H ,*0.*,68X,*SET*)
C
C              FIND THE N-TH ZERO BY APPLYING THE APPROXIMATION
C               FORMULA THEN THE REFINEMENT PROCEDURE
C
      DO 50 IOFN=2,NMAX
      NTHZERO = IOFN - 1
      IF(ETA .GE. .2)  CALL APROX1(RM,NTHZERO,ETA,ZERO)
      IF(ETA .LT. .2)  CALL APROX2(RM,NTHZERO,ETA,ZERO)
      GUESS(1) = ZERO
      GUESS(2) = ZERO + .1
      GUESS(3) =.ZERO - .1
      CALL JARRATT(GUESS,ITLIM,EP1,EP2,EQATION,ZERO,FT,IERJAR)
      IF(ITRACE .EQ. 2)  WRITE(6,20) ZERO,FT,GUESS(1),IERJAR
      SC(IOFN) = ZERO
   20 FORMAT(1H ,3E22.14,I6)
   50 CONTINUE
C
      DO 240 IOFM=1,NOFM
      MSAVE= IABS( MUSE(IOFM))
      IF(M-MSAVE) 240,210,240
  210 DO 230 IOFN=1,NMAX
  230 ARMJMN(IOFN,IOFM ) = SC(IOFN)
  240 CONTINUE
      SC(1) = 1.
C
      IF( MMAX.EQ.0 ) GO TO 700
C
C              NOW STEP THROUGH THE ORDERS
C
      DO 500 M=1,MMAX
      RM=M
      IF(ITRACE.EQ.2) WRITE(6,10)M
      MP1=M+1
C
C              FIND THE N-TH ZERO BY ITERATION
C
      DO 500 IOFN=1,NMAX
```

```
      D1 = BES(MP1)
      D2 = BES(MP2)
      FJFP = -D2 + (M/Y)*D1

      CALL  BF4F(Y,BES,MP1,IERR,ISIGN)
      D3 = BES(MP1)
      D4 = BES(MP2)
      FNFP = -D4 + (M/Y)*D3


      EQATION=FJFP*FNP - FJP*FNFP

   20 RETJRN
      END
```

3.2.4 Function UNEGNFN

Purpose: This function subprogram computes the m^{th} order (m an integer) unnormalized radial eigenfunction for a hardwalled annular duct of hub-to-tip ratio, η, when the argument is $\mu_{mn}\rho$, where μ_{mn} is the n^{th} hardwall eigenvalue of an m^{th} order duct mode and ρ is the polar radial coordinate nondimensionalized on the duct outer radius:

$$R_m\left(\mu_{mn}\rho\right) = J_m\left(\mu_{mn}\rho\right) - \frac{J_m'\left(\eta\mu_{mn}\right)}{Y_m'\left(\eta\mu_{mn}\right)} Y_m\left(\mu_{mn}\rho\right),$$

where J_m and Y_m are the Bessel and Neumann functions, respectively; the primes denote differentiation with respect to the argument; and η indicates the hub-to-tip ratio.

Method: The procedure is as follows:

1) Set working m to absolute value of input m.

2) Test for $\mu_{mn} = 0$, and, when true, set $R_{|m|}(\mu_{mn}\rho) = 1$ (for all ρ).

3) Evaluate $J_{|m|}(\mu_{mn}\rho)$.

4) Evaluate $J_{|m|}'(\mu_{mn}\eta)$, $Y_{|m|}'(\mu_{mn}\eta)$, and $Y_{|m|}(\mu_{mn}\eta)$ using the recurrence relations (formula [9.1.27] of ref. 30).

5) Evaluate $R_{|m|}(\mu_{mn}\rho)$.

6) Set $R_m(\mu_{mn}\rho) = (-1)^m R_{|m|}(\mu_{mn}\rho)$.

Usage: CALLING SEQUENCE

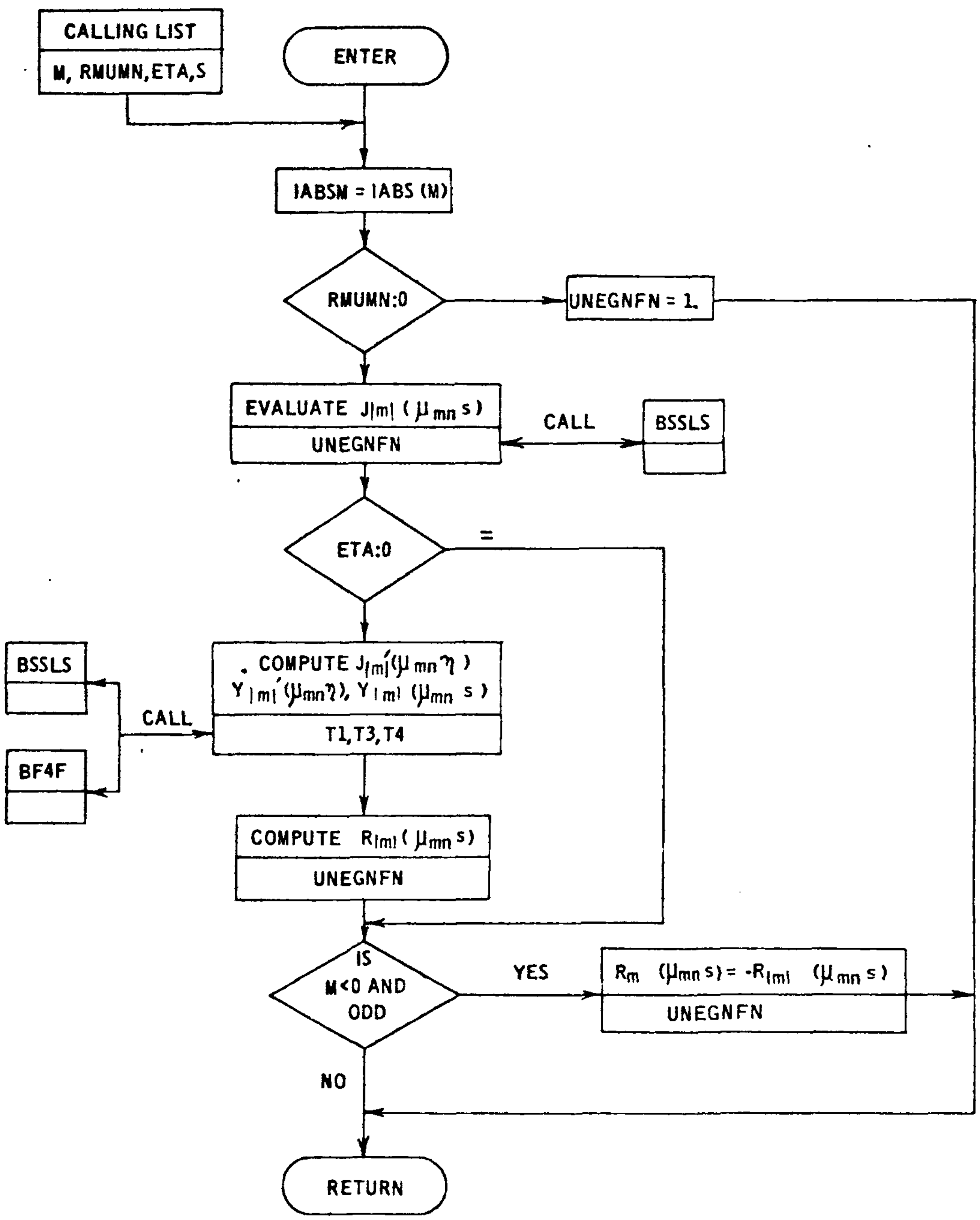
```
COMMON/SCRATCH/BES(1000)
.
.
.
CAPRMN=UNEGNFN(M,RMUMN,ETA,S)
```

Restrictions: $3\ \mu_{mn} + 12 + |m| \leqq 1000$; see subroutines BSSLS (sec. 3.3.6) and BF4F (ref. 41)

$S \geqq 0$

Timing: The timing is dominated by the Bessel function evaluators BSSLS and BF4F. It is, therefore, approximately equal to the sum of two unit calls to each.

Accuracy: The accuracy is of the algorithmic type and is dominated by that of either BSSLS or BF4F.


CALLING LIST
M, RMUMN,ETA,S
ENTER
IABSM = IABS (M)
RMUMN:0
UNEGNFN = 1.
EVALUATE J|m| (μmn s)
UNEGNFN
CALL
BSSLS
ETA:0
=
BSSLS
COMPUTE J|m|′(μmn η)
Y|m|′(μmn η), Y|m| (μmn s)
CALL
T1,T3,T4
BF4F
COMPUTE R|m| (μmn s)
UNEGNFN
IS
M<0 AND
ODD
YES
Rm (μmn s) = -R|m| (μmn s)
UNEGNFN
NO
RETURN

```
      FUNCTION UNEGNFN(M,RMUMN,ETA,S)
C
C
CPURPOSE        COMPUTE THE UNNORMALIZED ANNULAR EIGEN FUNCTION
C
C                           RMN(MUMN*S) = JM(MUMN*S) -
C                JMP(ETA*MUMN)*YM(MUMN*S)/YMP(ETA*MUMN)
C
C               WHERE S IS BETWEEN 0 AND 1
C                     MUMN IS THE N-TH EIGENVALUE OF THE ANNULAR EIGEN-
C                      VALUE EQUATION OBTAINED BY DIFFERENTIATING THE
C                      ABOVE EQUATION
C                     ETA IS THE RATIO OF THE INNER TO OUTER RADIUS OF
C                      THE ANNULUS
C                     JM AND YM ARE THE BESSEL FUNCTIONS OF THE FIRST AND
C                      SECOND KINDS OF INTEGER ORDER M
C                     JMP AND YMP ARE THE DERIVATIVES OF JM AND YM
C
CINPUT     VARIABLE DEFINITION
C                 M     INTEGER ORDER OF BESSEL FUNCTION
C             RMUMN     N-THE EIGENVALUE OF ANNULAR EIGENVALUE EQUATION,
C                            JMP(X) - JMP(ETA*X)*YMP(X)/YMP(ETA*X)
C                        MUMN ABOVE
C               ETA     RATIO OF INNER TO OUTER ANNULAR RADII
C                 S     ARGUMENT BETWEEN 0 AND 1, IN GENERAL RMN IS ONLY
C                        MEANINGFUL FOR S GREATER THAN OR EQUAL TO ETA
C
COUTPUT    UNEGNFN     THE VALUE OF RMN(MUMN*S)
C
CSUBPROGRAMS    BSSLS      EVALUATES JM FROM LRC  LIBRARY
C               BF4F       EVALUATES YM FROM LRC  LIBRARY
C
CRESTRICTION    3*RMUMN + M + 12 CAN BE AT MOST 1000 (SEEARRAY BJ)
C               ETA  BETWEEN 0 AND 1
C               S    BETWEEN 0 AND 1, GENERALLY AT LEAST ETA
C
C
      COMMON/SCRATCH/BES(1000)
      DATA ISIGN/-1/
C
C               COMPUTE BESSEL RELATED FUNCTIONS WITH POSITIVE ORDER AND
C               SWITCH SIGN FOR ODD NEGATIVE ORDER
      IABSM = IABS(M)
C
      IF(RMUMN ) 100,10,20
C
C               USE LIMITING VALUE FOR RMUMN=0 WHERE M=0
C
   10 IF(IABSM.EQ.0 ) UNEGNFN = 1.
      GO TO 100
   20 CONTINUE
      MP1 = IABSM + 1
```

```
      MP2 = MP1+1
      ARGETA = ETA*RMUMN
      ARGS   = S*RMUMN
C
C               COMPUTE JM(MUMN*S)
C
      CALL BSSLS(ARGS,BES,IABSM,IERR)
      T2 = BES(MP1)
      UNEGNFN = T2
      IF(ETA.EQ.0.) GO TO 90
C
C               COMPUTE YMP(ETA*MUMN)
C
      CALL BF4F(ARGETA,BES,MP1,IERR,ISIGN)
      T1 = -BES(MP2) + (IABSM/ARGETA)*BES(MP1)
C
C               COMPUTE JMP(ETA*MUMN)
C
      CALL BSSLS(ARGETA,BES,MP1,IERR)
      T3 = -BES(MP2) + (IABSM/ARGETA)*BES(MP1)
C
C               COMPUTE YM(MUMN*S)
C
      CALL BF4F(ARGS,BES,IABSM,IERR,ISIGN)
      T4 = BES(MP1)
C
      UNEGNFN = UNEGNFN - T3*T4/T1
   90 CONTINUE
      IF( M.LT.0 .AND. MOD(M,2).NE.0 )UNEGNFN = -UNEGNFN
C
  100 RETURN
      END
```

3.2.5 Function EGNNORM

Purpose: This function computes the normalization factor for the hard-wall duct radial eigenfunction $R_m(\mu_{mn}\rho)$ (see description of UNEGNFN):

$$N_{mn} = \left[\frac{1}{2}\left(1 - \frac{m^2}{\mu_{mn}^2}\right) R_m^2\left(\mu_{mn}\right) - \frac{1}{2}\left(\eta^2 - \frac{m^2}{\mu_{mn}^2}\right) R_m^2\left(\mu_{mn}\eta\right)\right]^{\frac{1}{2}}$$

for $m \neq 0$ and $m = 0$, $n \neq 0$ and

$$N_{oo} = \left[\frac{1}{2}\left(1-\eta\right)\right]^{\frac{1}{2}}$$

Method: The procedure is as follows:

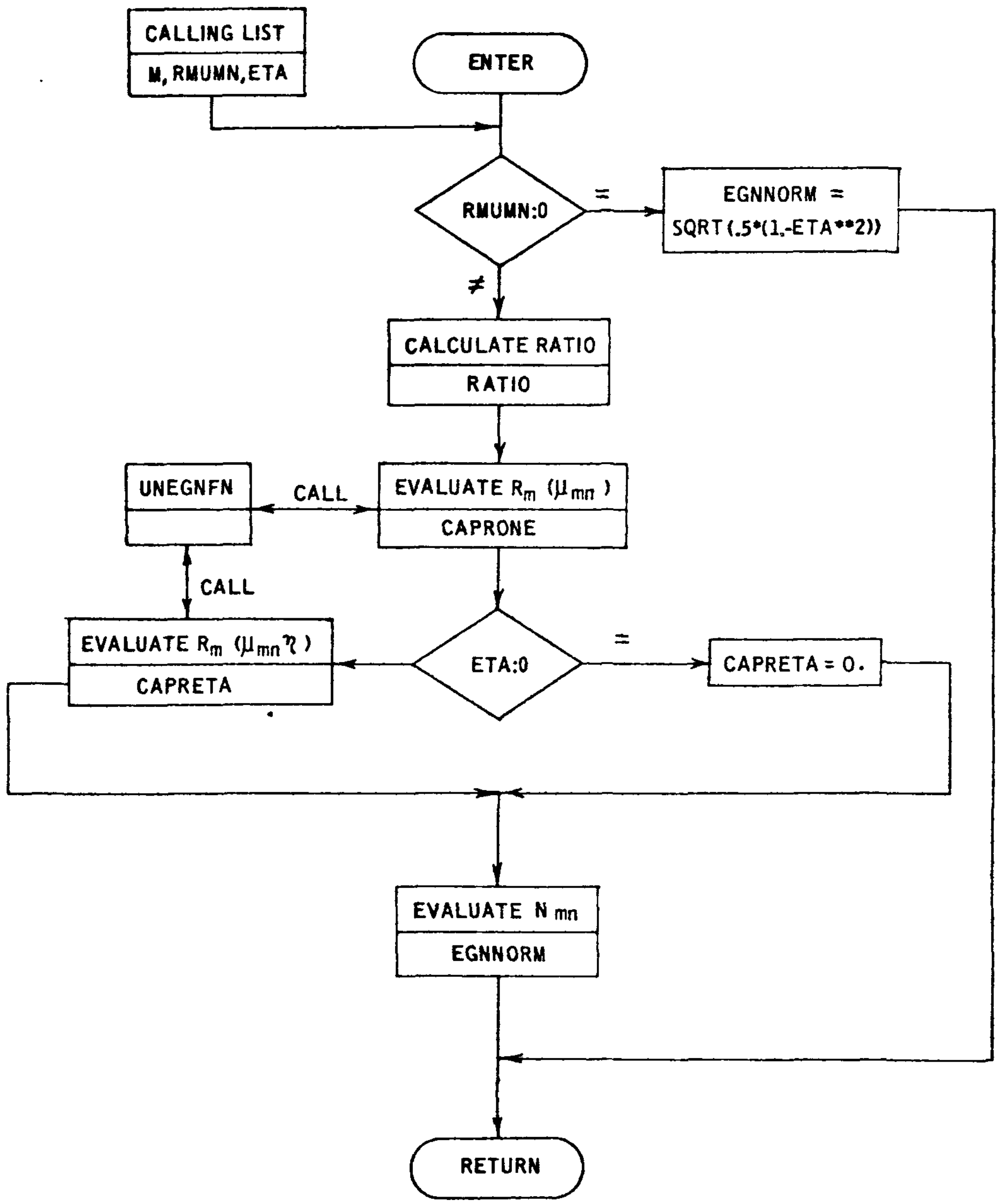
1) Set N_{oo} when $\mu_{mn} = 0$.
2) Compute the ratio $(m/\mu_{mn})^2$.
3) Evaluate $R_m(\mu_{mn})$ and $R_m(\mu_{mn}\eta)$ where $R_m(\mu_{mn}\eta)$ is set to zero when $\eta = 0$.
4) Evaluate N_{mn}.

Usage: CALLING SEQUENCE

```
CAPNMN=EGNNORM(M,RMUMN,ETA)
```

Timing: The timing is dominated by function UNEGNFN (sec. 3.2.4), approximately equal to two unit calls to that function.

Accuracy: The accuracy is of the algorithmic type, and, in particular, it is dominated by function UNEGNFN.


CALLING LIST
M, RMUMN, ETA
ENTER
RMUMN:0
=
≠
EGNNORM =
SQRT(.5*(1.-ETA**2))
CALCULATE RATIO
RATIO
UNEGNFN
CALL
EVALUATE R_m (μ_{mn})
CAPRONE
CALL
EVALUATE R_m ($\mu_{mn}\eta$)
CAPRETA
ETA:0
=
CAPRETA = 0.
EVALUATE N_{mn}
EGNNORM
RETURN

```
      FUNCTION EGNNORM(M,RMUMN,ETA)
C
C
CPURPOSE         COMPUTE THE NORMALIZATION FACTOR TO THE ANNULAR EIGEN-
C                 FUNCTION
C
C                  NMN**2 = .5*( (1.-M**2/MUMN**2)*RMN(MUMN)**2
C                               - (ETA**2-M**2/MUMN**2)*RMN(ETA*MUMN)**2 )
C
C                WHERE RMN IS THE UNNORMALIZED ANNULAR EIGEN FUNCTION
C                      M IS BESSEL FUNCTION ORDER FOR RMN
C                      MUMN IS ANNULAR EIGENVALUE
C                      ETA IS RATIO INNER TO OUTER ANNULAR DUCT RADII
C
CINPUT           VARIABLE  DEFINITION
C                      M    BESSEL FUNCTION ORDER
C                   MUMN    ANNULAR EIGENVALUE
C                    ETA    RATIO INNER TO OUTER ANNULAR DUCT RADII
C
COUTPUT          EGNNORM    VALUE OF NMN
C
CSUBPROGRAMS     UNEGNFN    EVALUATES UNNORMALIZED ANNULAR EIGENFUNCTION
C
CRESTRICTIONS    SEE FUNCTION  UNEGNFN
C
      IF(RMUMN ) 100,10,20
C
C                USE LIMITING VALUE FOR RMUMN=0 WHERE M=0
C
   10 IF( M.EQ.0 )EGNNORM = SQRT( .5*(1.-ETA**2) )
      GO TO 100
   20 CONTINUE
      RATIO = (M/RMUMN)**2
      CAPRONE  =UNEGNFN(M,RMUMN,ETA,1.)
      CAPRETA =0.
      IF(ETA.NE.0.)CAPRETA=UNEGNFN(M,RMUMN,ETA,ETA)
C
      EGNNORM= .5*( (1.-RATIO)*CAPRONE**2 - (ETA**2-RATIO)*CAPRETA**2 )
      EGNNORM= SQRT(EGNNORM)
C
  100 RETURN
      END
```

3.2.6 Function FACTINT

Purpose: This function evaluates the oscillatory factor in the integral expression for the modal amplitudes for primary subroutine AAAAA:

$$e^{-i\ell N_{IS}\,\Theta_R}\,\mathcal{R}_m\left(\mu_{mn}\rho\right)\,e^{i\ell N_{IS}\frac{d\,\sin\psi}{\rho\,\cos\psi}}$$

for the inlet stator-rotor, and

$$e^{i\sigma N_R\,\Theta_{OS}}\,\mathcal{R}_m\left(\mu_{mn}\rho\right)\,e^{i\sigma N_R\frac{d\,\sin\psi}{\rho\,\cos\psi}}$$

for the rotor-outlet stator (see equation [36], appendix I, of volume I).

Method: The procedure is as follows:

1) Evaluate the normalized duct radial eigenfunction.

2) Compute the first exponential term.

3) Compute M_z and M_{1E} from input or spanwise interpolation and the flow angle,

4) Compute the second exponential factor.

5) Evaluate the oscillatory factor.

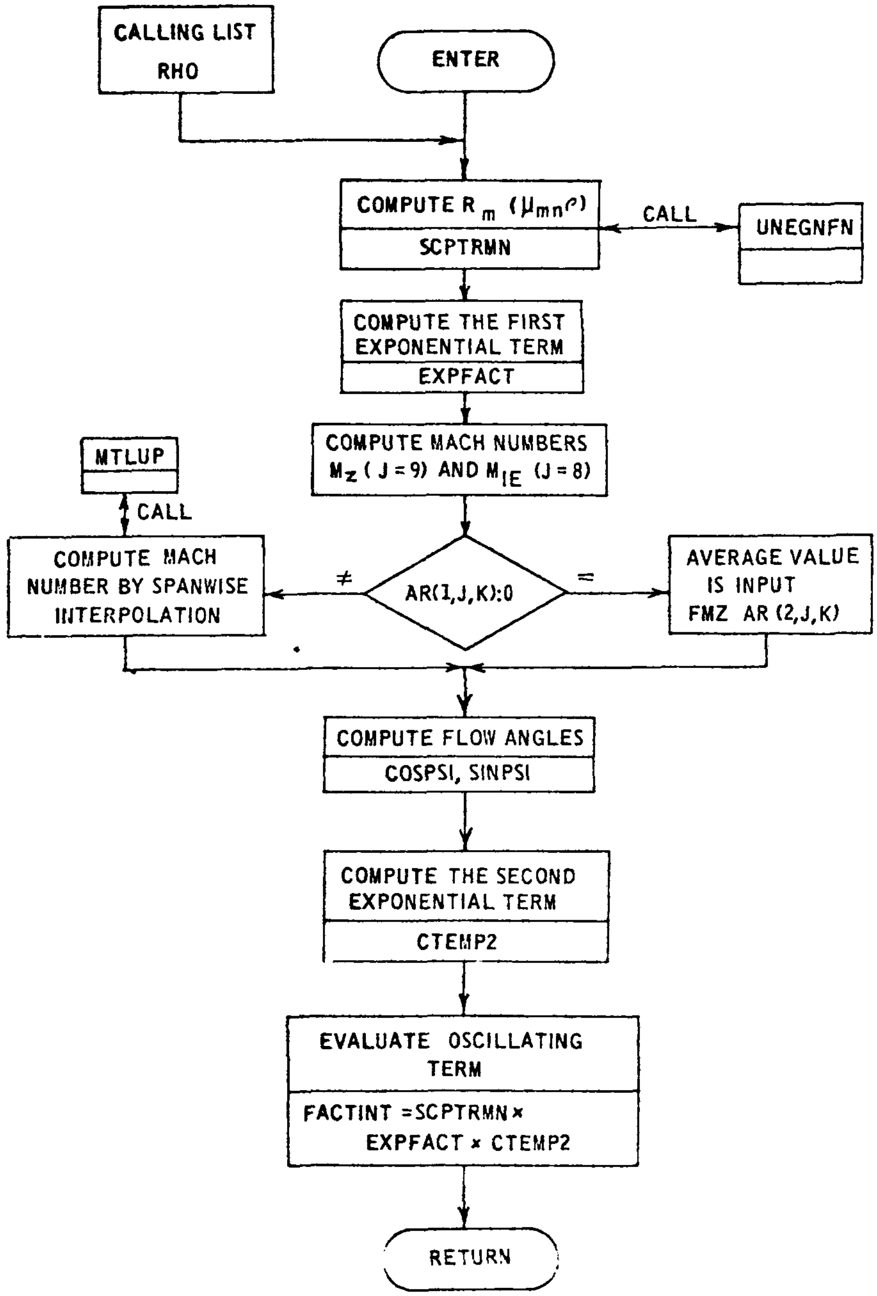
Usage: CALLING SEQUENCE

```
COMPLEX FACTINT,FACTRPD
COMMON/CFACT/M,N,RMUMN,CAPNMN,ETA,SIGN,L,CAPKMN
COMMON/CFACTIR/NSBIR,SIGOL,PHISBIR
.
.
.
FACTRPD = FACTINT (RHO)
```

Restrictions: ETA ≤ RHO ≤ 1

Timing: The timing is dominated by the eigenfunction evaluation, which is approximately equal to a unit call to the function UNEGNFN (see sec. 3.2.4).

Accuracy: The accuracy is of the algorithmic type and, in particular, is dominated by UNEGNFN.


CALLING LIST
RHO
ENTER
COMPUTE R_m ($\mu_{mn}\rho$)
SCPTRMN
CALL
UNEGNFN
COMPUTE THE FIRST EXPONENTIAL TERM
EXPFACT
COMPUTE MACH NUMBERS M_Z (J = 9) AND M_{IE} (J = 8)
MTLUP
CALL
COMPUTE MACH NUMBER BY SPANWISE INTERPOLATION
≠
AR(1,J,K):0
=
AVERAGE VALUE IS INPUT FMZ AR (2,J,K)
COMPUTE FLOW ANGLES
COSPSI, SINPSI
COMPUTE THE SECOND EXPONENTIAL TERM
CTEMP2
EVALUATE OSCILLATING TERM
FACTINT = SCPTRMN × EXPFACT × CTEMP2
RETURN

```
      COMPLEX FUNCTION FACTINT(RHO,ARMISC,MAXDIM,MAXJ,AR)
C
C PURPOSE       EVALUATE THE INTEGRAND FACTINT TO BE CALLED BY THE
C               INTEGRATOR GAUSS2 IN PRIMARY SUBROUTINE AAAAA
C
C  COMMON        CFACT
C   BLOCKS       CFACTIR
C
C SUBPROGRAMS    UNEGNFN
C  CALLED
C
      DIMENSION AR(MAXDIM,MAXJ,3),ARMISC(1)
C
      COMPLEX EXPFACT,CTEMP2
C
      COMMON/CFACT/  M,N,RMUMN,CAPNMN,ETA,SIGN,L,CAPKMN
      COMMON/CFACTIR/NSBIR,SIGOL,PHISBIR
C
C               EVALUATE NORMALIZED EIGENFUNCTION
C
      SCPTRMN = UNEGNFN(M,RMUMN,ETA,RHO)/CAPNMN
C
C               EVALUATE FIRST EXPONENTIAL TRIGNOMETRIC FACTOR
C
      ARGEXP2 = SIGN*SIGOL*NSBIR*PHISBIR*(RHO-ETA)/(1.-ETA)
      EXPFACT = CMPLX( COS(ARGEXP2),SIN(ARGEXP2) )
C
C               EVALUATE NEXT  EXPONENTIAL TRIGNOMETRIC FACTOR
C
C               COMPUTE FMZ AND FM1E
C
      ISOROS = ARMISC(5)
      J = 9
      K = ISOROS+1
      NSPN = AR(1,J,K)
      IF(NSPN ) 40,30,40
   30 FMZ = AR(2,J,K)
      GO TO 50
   40 IPA = -1
      CALL  MTLUP(RHO,FMZ,1,NSPN,NSPN,1,IPA,AR(3,1,K),AR(3,J,K)  )
   50 J = 8
      K = ISOROS
      NSPN = AR(1,J,K)
      IF(NSPN) 70,60,70
   60 FM1E = AR(2,J,K)
      GO TO 80
   70 IPA = -1
      CALL  MTLUP(RHO,FM1E,1,NSPN,NSPN,1,IPA,AR(3,1,K),AR(3,J,K)  )
C
C               EVALUATE THE FACTOR
C
   80 COSPSI = FMZ/FM1E
```

```
      SINPSI = SQRT(1.-COSPSI**2)
      DSPAC = ARMISC(ISOROS)
      TEMP1 = SIGOL*NSBIR*DSPAC*SINPSI/(RHO*COSPSI)
      CTEMP2 = CMPLX( COS(TEMP1),SIN(TEMP1) )
C
C           COMBINE TO FORM FACTINT
C
      FACTINT = EXPFACT*SCPTRMN*CTEMP2
C
      RETURN
      END
```

3.2.7 Function FACTIN2

Purpose: This function computes the oscillatory factor in the integral expression for the modal amplitudes of primary subroutine AABAA:

$$M_{M,K1}(\rho)\Gamma^{o}_{K2}(\rho)a_{\kappa,K1}(\rho)H_{\kappa,K1}(\rho)\ e^{-id_{\kappa,K1}(\rho)}\quad K_{\kappa,K1}(\rho)$$

$$*\left(\frac{me_{\phi}}{\rho}+K^{\pm}_{mn}e_{z}\right)\left(\frac{dC_{L}}{d\alpha}(\rho)\Big/2\pi\right)\mathcal{R}_{m}\left(\mu_{mn}\rho\right)$$

(See equation [47], appendix I, of volume I.)

Method: The procedure is as follows:

1) Initialize the component index to K1.

2) Initialize the table lookup position index.

3) Determine the value of the inlet, exit, and axial Mach numbers and the chord at the given span position by using the input average values, or by interpolating on the input tables of spanwise values.

4) Compute $M_{M,K1}(\rho)$ and $\theta_{K1}(\rho)$.

5) Repeat steps 2, 3, and 4 for component index K2.

6) Compute the Glauert coefficients for component K2 using the procedure described in step 3.

7) Compute $\Gamma^{o}_{K2}(\rho)$ and $a_{\kappa,K1}(\rho)$.

8) Compute the Bessel function argument $h_{K2}(\rho)$.

9) Compute $g_{K2}(\rho)$ for index n = 1, up to the number of Glauert coefficients which were input.

10) Compute the Bessel function $J_n(h_{K2})$ for zero and the n's in step 9.

11) Compute $H_{\kappa,K2}(\rho)$ by summation.

12) Compute $d_{\kappa,K1}(\rho)$ and $e^{-id_{\kappa,K1}(\rho)}$.

13) Compute γ and λ and the corresponding value of the lift function.

14) Compute the factor $\left(\frac{me_\phi}{\rho} + K_{mn}^{\pm} e_z\right)\left(\frac{dC_L}{d\alpha}(\rho)\Big/2\pi\right)$.

15) Compute the normalized duct radial eigenfunction.

16) Compute the oscillatory factor.

Usage: CALLING SEQUENCE

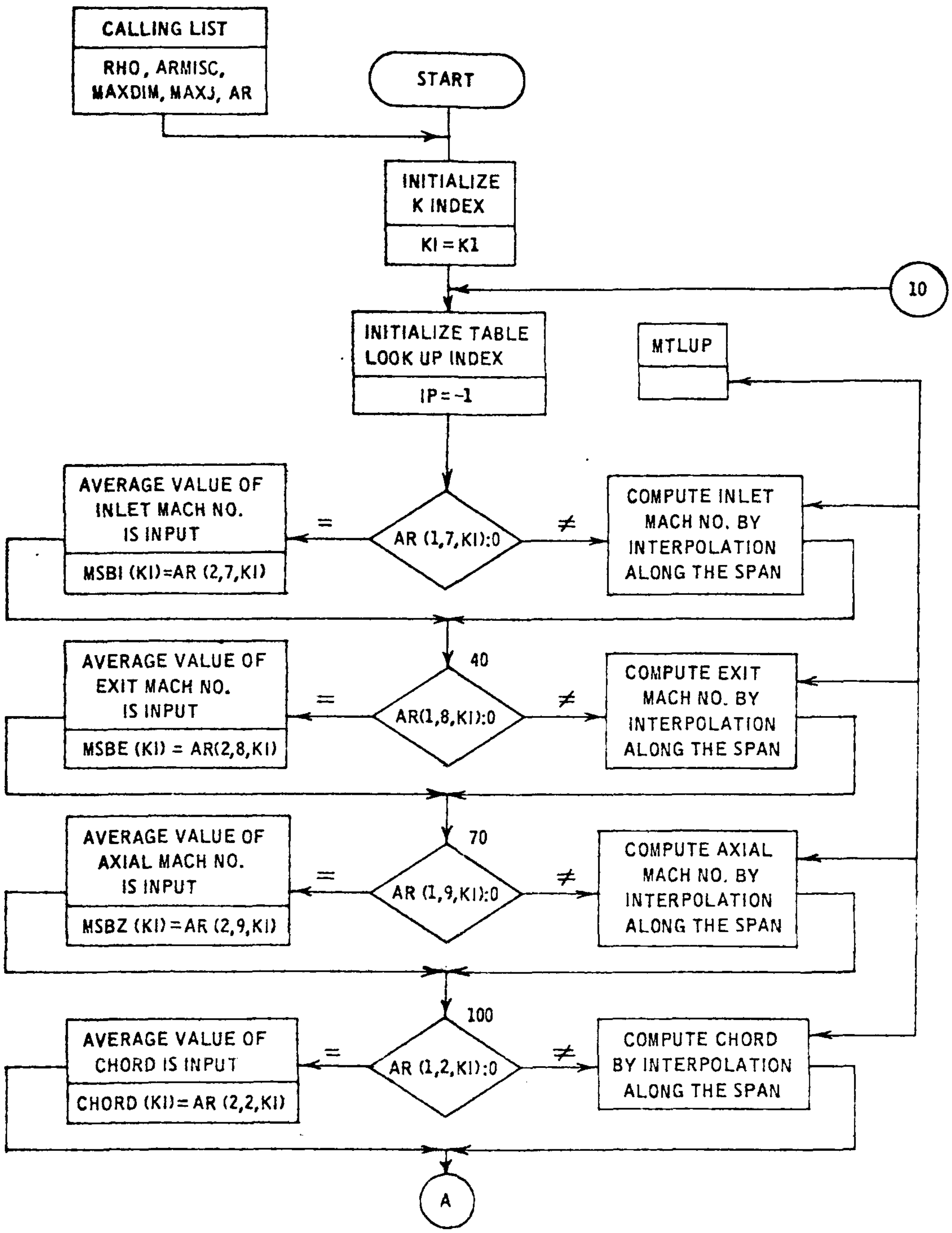
```
COMPLEX FACTIN2,FACTRPD
DIMENSION AR(MAXDIM,MAXJ,3),ARMISC(21)
COMMON/SCRATCH/BES(1000)
COMMON/CFACT2/B,CAPKMN,CAPNMN,C3,
* C7,C8,C9,C11,C12,C13,C14,K1,K2,L,M,N,NK2,RMUMN,SIGOL
.
.
.
FACTRPD = FACTIN2(RHO,ARMISC,MAXDIM,MAXJ,AR)
```

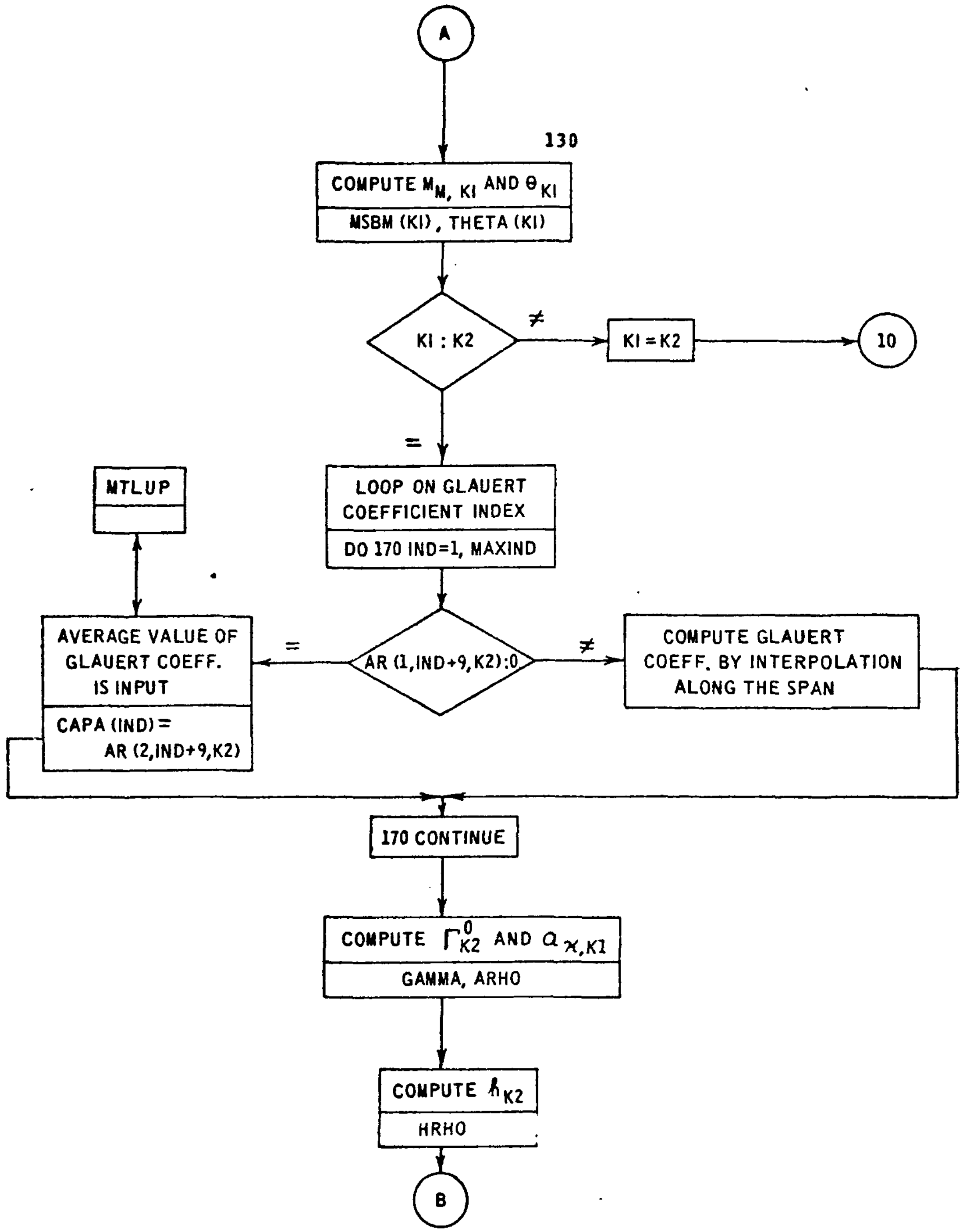
Restrictions: ETA $\leq$ RHO $\leq$ 1

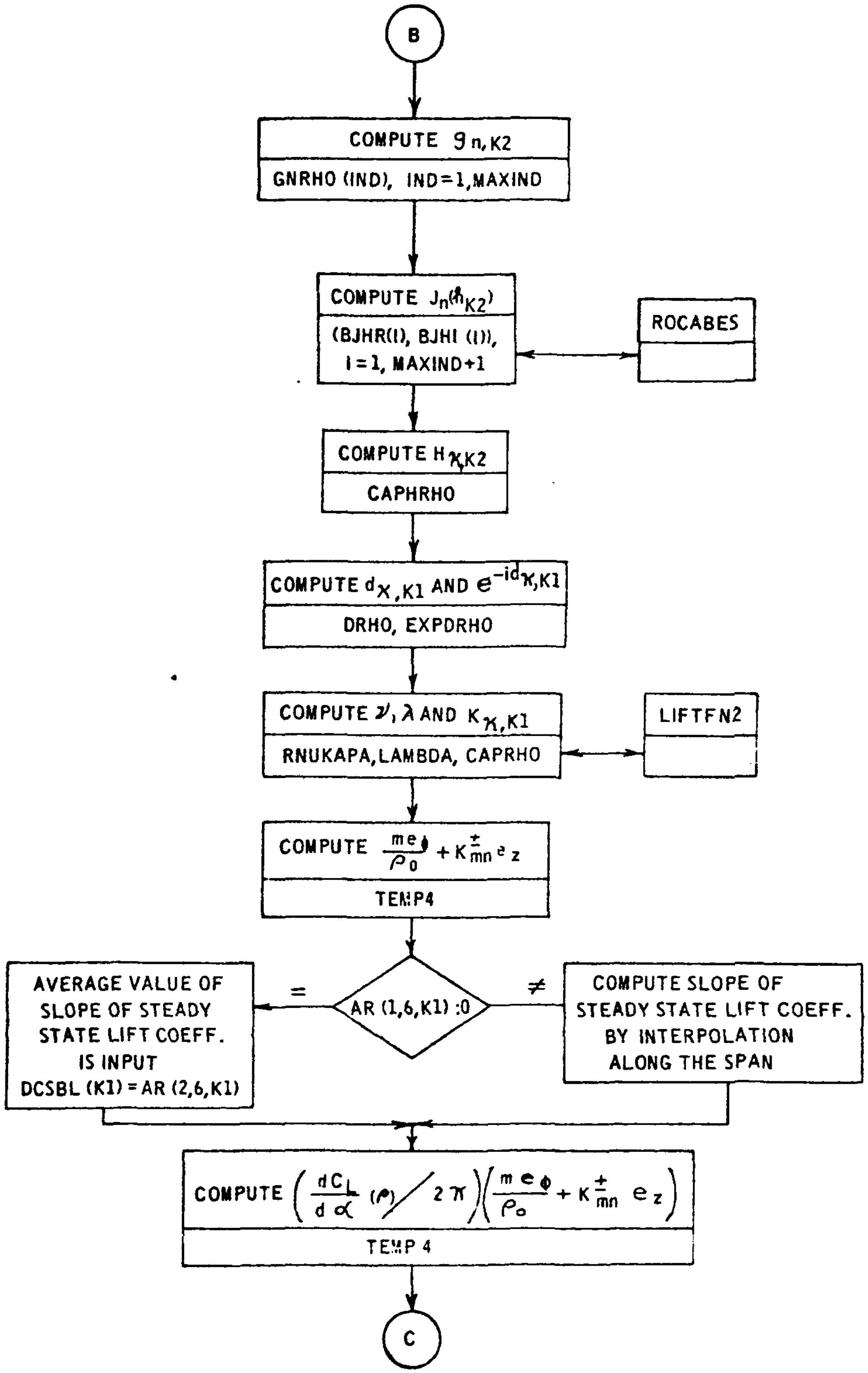
3 $\leq$ ARMISC (18 + K2) $\leq$ 15

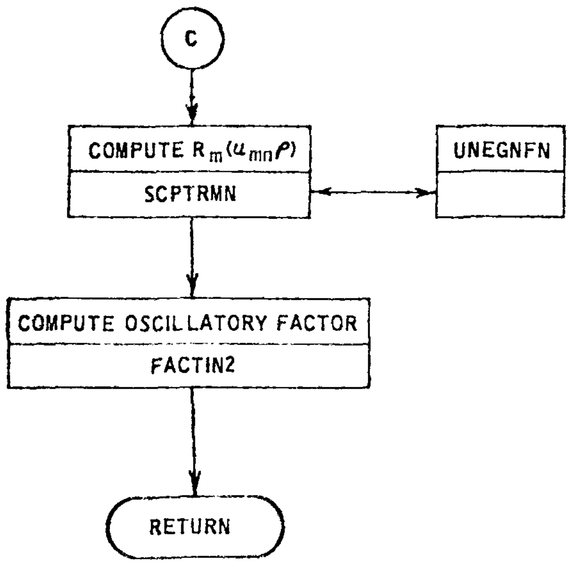
Timing: The timing is dominated by the interpolation, Bessel function evaluation, and lift function evaluation and is approximately equal to five unit calls to subroutine MTLUP (see ref. 42) plus one unit call to subroutine ROCABES (see sec. 3.3.11) and one unit call to subroutine LIFTFN2 (see sec. 3.2.10).

Accuracy: The accuracy is of the algorithmic type and, in particular, it is dominated by ROCABES.


CALLING LIST
RHO, ARMISC, MAXDIM, MAXJ, AR
START
INITIALIZE K INDEX
KI = K1
10
INITIALIZE TABLE LOOK UP INDEX
IP = -1
MTLUP
AVERAGE VALUE OF INLET MACH NO. IS INPUT
MSBI (KI)=AR (2,7,KI)
=
AR (1,7,KI):0
≠
COMPUTE INLET MACH NO. BY INTERPOLATION ALONG THE SPAN
40
AVERAGE VALUE OF EXIT MACH NO. IS INPUT
MSBE (KI) = AR(2,8,KI)
=
AR(1,8,KI):0
≠
COMPUTE EXIT MACH NO. BY INTERPOLATION ALONG THE SPAN
70
AVERAGE VALUE OF AXIAL MACH NO. IS INPUT
MSBZ (KI) = AR (2,9,KI)
=
AR (1,9,KI):0
≠
COMPUTE AXIAL MACH NO. BY INTERPOLATION ALONG THE SPAN
100
AVERAGE VALUE OF CHORD IS INPUT
CHORD (KI)= AR (2,2,KI)
=
AR (1,2,KI):0
≠
COMPUTE CHORD BY INTERPOLATION ALONG THE SPAN
A


A
130
COMPUTE $M_{M,KI}$ AND θ_{KI}
MSBM (KI), THETA (KI)
KI : K2
≠
KI = K2
10
=
LOOP ON GLAUERT COEFFICIENT INDEX
DO 170 IND=1, MAXIND
MTLUP
AVERAGE VALUE OF GLAUERT COEFF. IS INPUT
CAPA (IND) = AR (2,IND+9,K2)
=
AR (1,IND+9,K2):0
≠
COMPUTE GLAUERT COEFF. BY INTERPOLATION ALONG THE SPAN
170 CONTINUE
COMPUTE Γ^0_{K2} AND $a_{\varkappa,K1}$
GAMMA, ARHO
COMPUTE h_{K2}
HRHO
B


B
COMPUTE $g_{n,K2}$
GNRHO (IND), IND=1,MAXIND
COMPUTE $J_n(\rho_{K2})$
(BJHR(I), BJHI (I)), I=1, MAXIND+1
ROCABES
COMPUTE $H_{\kappa,K2}$
CAPHRHO
COMPUTE $d_{\kappa,K1}$ AND $e^{-id_{\kappa,K1}}$
DRHO, EXPDRHO
COMPUTE ν, λ AND $K_{\kappa,K1}$
RNUKAPA, LAMBDA, CAPRHO
LIFTFN2
COMPUTE $\frac{me_\phi}{\rho_0} + K^{\pm}_{mn} e_z$
TEMP4
AR (1,6,K1) :0
=
≠
AVERAGE VALUE OF SLOPE OF STEADY STATE LIFT COEFF. IS INPUT
DCSBL (K1) = AR (2,6,K1)
COMPUTE SLOPE OF STEADY STATE LIFT COEFF. BY INTERPOLATION ALONG THE SPAN
COMPUTE $\left(\frac{dC_L}{d\alpha}(\rho)\Big/2\pi\right)\left(\frac{me_\phi}{\rho_0} + K^{\pm}_{mn} e_z\right)$
TEMP 4
C


C
COMPUTE $R_m(u_{mn}\rho)$
SCPTRMN
UNEGNFN
COMPUTE OSCILLATORY FACTOR
FACTIN2
RETURN

```
      COMPLEX FUNCTION FACTIN2(RHO,ARMISC,MAXDIM,MAXJ,AR)
      REAL MSBE(3),MSBI(3),MSBM(3),MSBZ(3)
      COMPLEX CAPHRHO,CONLIFT,CTEMP1,CTEMP2,HRHO,LAMDA,LIFT,TERM(15)
      COMPLEX CAPKRHO,EXPORHO
      DIMENSION AR(MAXDIM,MAXJ,3),ARMISC(1),BJHR(250),BJHI(250),CAPA(15)
      DIMENSION CHORD(3),DCSBL(3),GNRHO(15),YRE(20),YIM(20),THETA(3)
      COMMON/SCRATCH/BES(1000)
      COMMON/CFACT2/B,CAPKMN,CAPNMN,C3,C6,C7,C8,C9,C11,C12,C13,C14,K1,
     1              K2,L,M,N,NK2,RMUMN,SIGOL
      EQUIVALENCE(BJHR(1),BES(1)),(BJHI(1),BES(251))
      KI = K1
10    IP = -1
      IF(AR(1,7,KI) .GT. 0.)  GO TO 20
      MSBI(KI) = AR(2,7,KI)
      GO TO 40
20    NPTS = AR(1,7,KI)
      CALL MTLUP(RHO,MSBI(KI),1,NPTS,NPTS,1,IP,AR(3,1,KI),AR(3,7,KI))
40    IF(AR(1,8,KI) .GT. 0.)  GO TO 50
      MSBE(KI) = AR(2,8,KI)
      GO TO 70
50    NPTS = AR(1,8,KI)
      CALL MTLUP(RHO,MSBE(KI),1,NPTS,NPTS,1,IP,AR(3,1,KI),AR(3,8,KI))
70    IF(AR(1,9,KI) .GT. 0.)  GO TO 80
      MSBZ(KI) = AR(2,9,KI)
      GO TO 100
80    NPTS = AR(1,9,KI)
      CALL MTLUP(RHO,MSBZ(KI),1,NPTS,NPTS,1,IP,AR(3,1,KI),AR(3,9,KI))
100   IF(AR(1,2,KI) .GT. 0.)  GO TO 110
      CHORD(KI) = AR(2,2,KI)
      GO TO 130
110   NPTS = AR(1,2,KI)
      CALL MTLUP(RHO,CHORD(KI),1,NPTS,NPTS,1,IP,AR(3,1,KI),AR(3,2,KI))
130   IF(MSBI(KI) .LT. MSBZ(KI))   TEMP1 = 0.
      IF(MSBI(KI) .GE. MSBZ(KI))   TEMP1 = SQRT(MSBI(KI)**2
     1                                    - MSBZ(KI)**2)
      IF(MSBE(KI) .LT. MSBZ(KI))   TEMP2 = 0.
      IF(MSBE(KI) .GE. MSBZ(KI))   TEMP2 = SQRT(MSBE(KI)**2
     1                                    - MSBZ(KI)**2)
      TEMP3 = TEMP1 + TEMP2
      TEMP4 = (TEMP3**2)/4. + MSBZ(KI)**2
      MSBM(KI) = SQRT(TEMP4)
      TEMP1 = ACOS( MSBZ(KI)/MSBM(KI) )
      THETA(KI) = ABS(TEMP1)
      IF(KI .EQ. K2)     GO TO 140
      KI = K2
      GO TO 10
140   MAXIND = ARMISC(K2+18)
      DO 170 IND=1,MAXIND
      IF(AR(1,IND+9,K2) .GT. 0.)    GO TO 150
      CAPA(IND) = AR(2,IND+9,K2)
      GO TO 170
150   NPTS = AR(1,IND+9,K2)
```

```
      CALL MTLUP(RHO,CAPA(IND),1,NPTS,NPTS,1,IP,AR(3,1,K2),
     1           AR(3,IND+9,K2))
170   CONTINUE
      GAMMA = 3.14159265358 93*CHORD(K2)*MSBM(K2)*(CAPA(1) + CAPA(2))
      TEMP1 = C6*( CHORD(K1)/(2.*RHO) )*NK2
      TEMP2 = -C7*SIGOL*( B/RHO )*NK2
      TEMP3 = EXP(TEMP2)
      ARHO  = TEMP1*TEMP3
      TEMP1 = (C12*SIGOL*CHORD(K2)*NK2) / (2.*RHO)
      TEMP2 = C13*(1.570796326795 - C14*THETA(K2))
      CTEMP1 = CMPLX(0.,TEMP2)
      HRHO  = TEMP1*CEXP(CTEMP1)
      TEMP1 = CAPA(1) + CAPA(2)
      CAPA(MAXIND+1) = 0.
      CAPA(MAXIND+2) = 0.
      DO 180 IND=1,MAXIND
180   GNRHO(IND) = (CAPA(IND+2) - CAPA(IND)) / TEMP1
      HRHOR = REAL(HRHO)
      HRHOI = AIMAG(HRHO)
      CALL ROCABES(HRHOR,HRHOI,0.,MAXIND,BJHR,BJHI,YRE,YIM)
      CAPHRHO = CMPLX(BJHR(1),BJHI(1))
      DO 190 I=1,MAXIND
190   TERM(I) = (CMPLX(0.,C11)**I)*GNRHO(I)*CMPLX(BJHR(I+1),BJHI(I+1))
      DO 200 I=1,MAXIND
200   CAPHRHO = CAPHRHO + TERM(I)
      TEMP1 = C8*THETA(K1)
      TEMP2 = C9*SIGOL*NK2
      TEMP3 = ( B/RHO )*TAN(THETA(K1))
      TEMP4 = (ARMISC(7)*CHORD(K2)) / (2.*MSBM(K2))
      DRHO  = TEMP1 + TEMP2*(TEMP3 + TEMP4)
      CTEMP1 = CMPLX(0.,-DRHO)
      EXPDRHO = CEXP(CTEMP1)
      RNUKAPA = (ARMISC(7)*NK2*CHORD(K1)*SIGOL) / (MSBM(K1)*2.)
      RNUKAPA = C6*RNUKAPA
      TEMP1 = C3*(1.570796326795 - THETA(K1))
      CTEMP1 = CMPLX(0.,TEMP1)
      CTEMP2 = CEXP(CTEMP1)
      TEMP2 = C6*SIGOL*( NK2/(2.*RHO) )*CHORD(K1)
      LAMDA = TEMP2*CTEMP2
      CALL LIFTFN2(RNUKAPA,LAMDA,LIFT,CONLIFT)
      CAPKRHO = LIFT
      SINTHS = SIN(THETA(K1))
      COSTHS = COS(THETA(K1))
      INDX2 = ARMISC(5) + ARMISC(13)
      IF(INDX2 .EQ. 1  .OR.  INDX2 .EQ. 2)      221,222
221   TEMP3 = (M*COSTHS) / RHO
      GO TO 223
222   TEMP3 = -(M*COSTHS) / RHO
223   TEMP4 = TEMP3  + CAPKMN*SINTHS
      IF(AR(1,6,K1) .GT. 0.)  GO TO 224
      DCSBL(K1) = AR(2,6,K1)
      GO TO 225
```

```
 224  NPTS = AR(1,6,K1)
      IP = -1
      CALL MTLUP(RHO,DCSBL(K1),1,NPTS,NPTS,1,IP,AR(3,1,K1),AR(3,6,K1))
 225  TEMP4 = ( DCSBL(K1)/6.2831853071796 ) * TEMP4
C
      SCPTRMN = UNEGNFN(M,RMUMN,ARMISC(3),RHO) / CAPMMN
      FACTIN2 =MSBM(K1)*GAMMA*ARHO*CAPHRHO*EXPDRHO*CAPKRHO*TEMP4*SCPTRMN
      IF(ARMISC(6) .EQ. 3.)  WRITE(6,230) RHO,MSBM(K1),GAMMA,ARHO,
     1                       CAPHRHO,EXPDRHO,CAPKRHO,TEMP4,SCPTRMN,FACTIN2
 230  FORMAT(1H ,F7.4,F6.3,2(1X,E9.2),3(1X,2E9.2),2(1X,E9.2),1X,2E10.3)
      RETURN
      END
```

3.2.8 Function FACTIN3

Purpose: This function evaluates the interval of the oscillatory factor called by subroutine GAUSS2 in the primary subroutine BBCAA.

Method: The procedure is as follows:

1) Evaluate the normalized eigenfunction.

2) Initialize the distortion coefficient to zero.

3) If the distortion coefficient index ℓ is zero, proceed to step 16.

4) If the cone model is not being used, proceed to step 8.

5) Calculate $D = (1 - V_A/V_1)/(A^2 - 1)$, $\delta_{\ell,0}$ and $\delta_{\ell,1} + \delta_{\ell-1}$.

6) Evaluate the integral part by dividing the interval into two equispaced subintervals, integrating on each subinterval with an eight-point Gaussian formula, and summing the integrals.

7) Compute $V_\ell(p)$ and $W_\ell(p)$ in the cone model and proceed to step 16.

8) If the power model is not being used, proceed to step 12.

9) If the average value of a_1 is input, use it and proceed to step 11.

10) Compute an average value of a_1.

11) Compute $W_\ell(p)$ in the power model and proceed to step 16.

12) If the distortion coefficients are not input, proceed to step 16.

13) Determine if the distortion coefficient index $|\ell|$ corresponds to an input value. If it does not, proceed to step 16; if it does, determine which index.

14) Compute the distortion sine and cosine coefficients for the present index, depending upon whether average or spanwise data is input.

15) For $\ell < 0$, conjugate the coefficient.

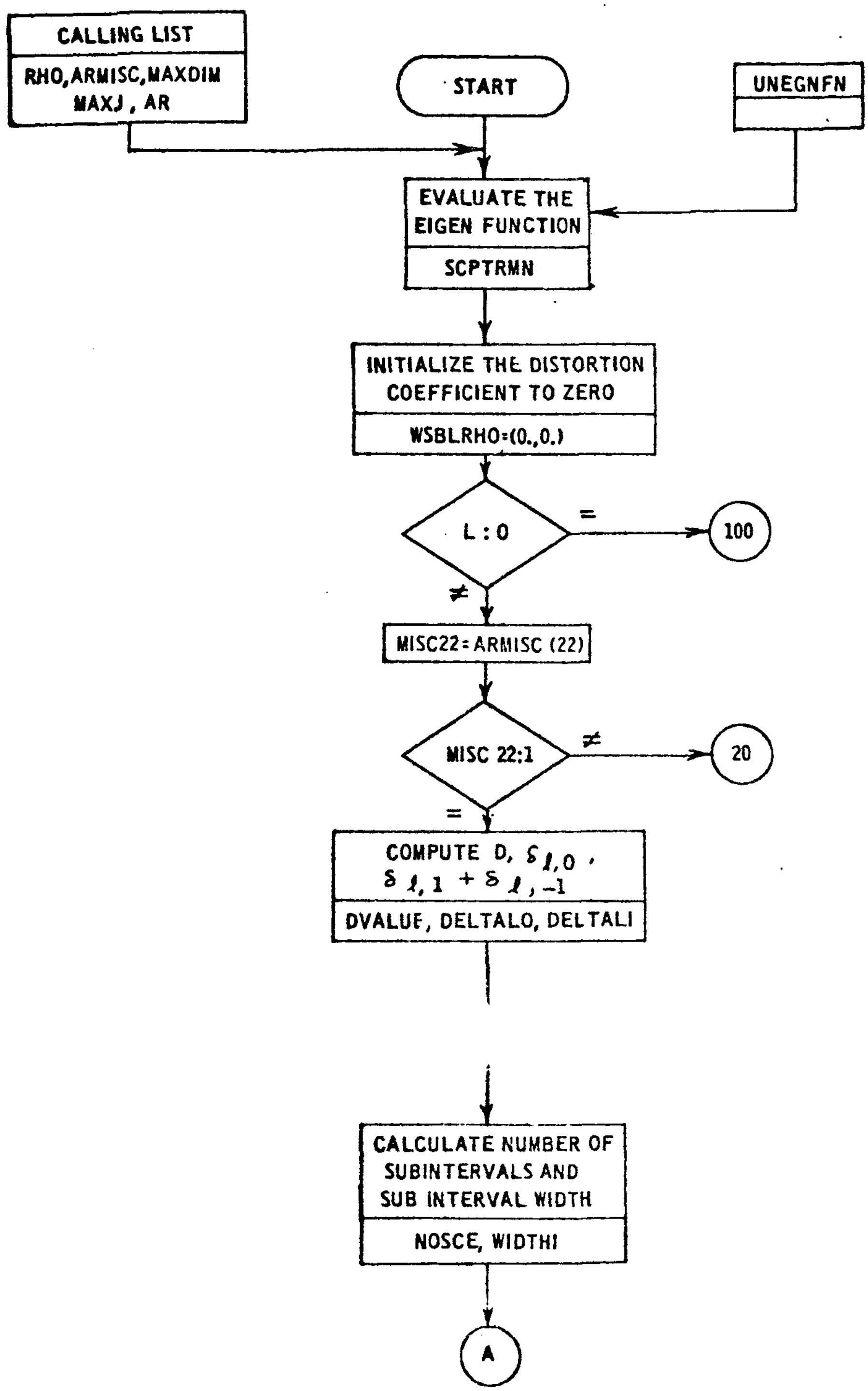
16) Compute the integrand as the product of the eigenfunction and the computed distortion coefficient.

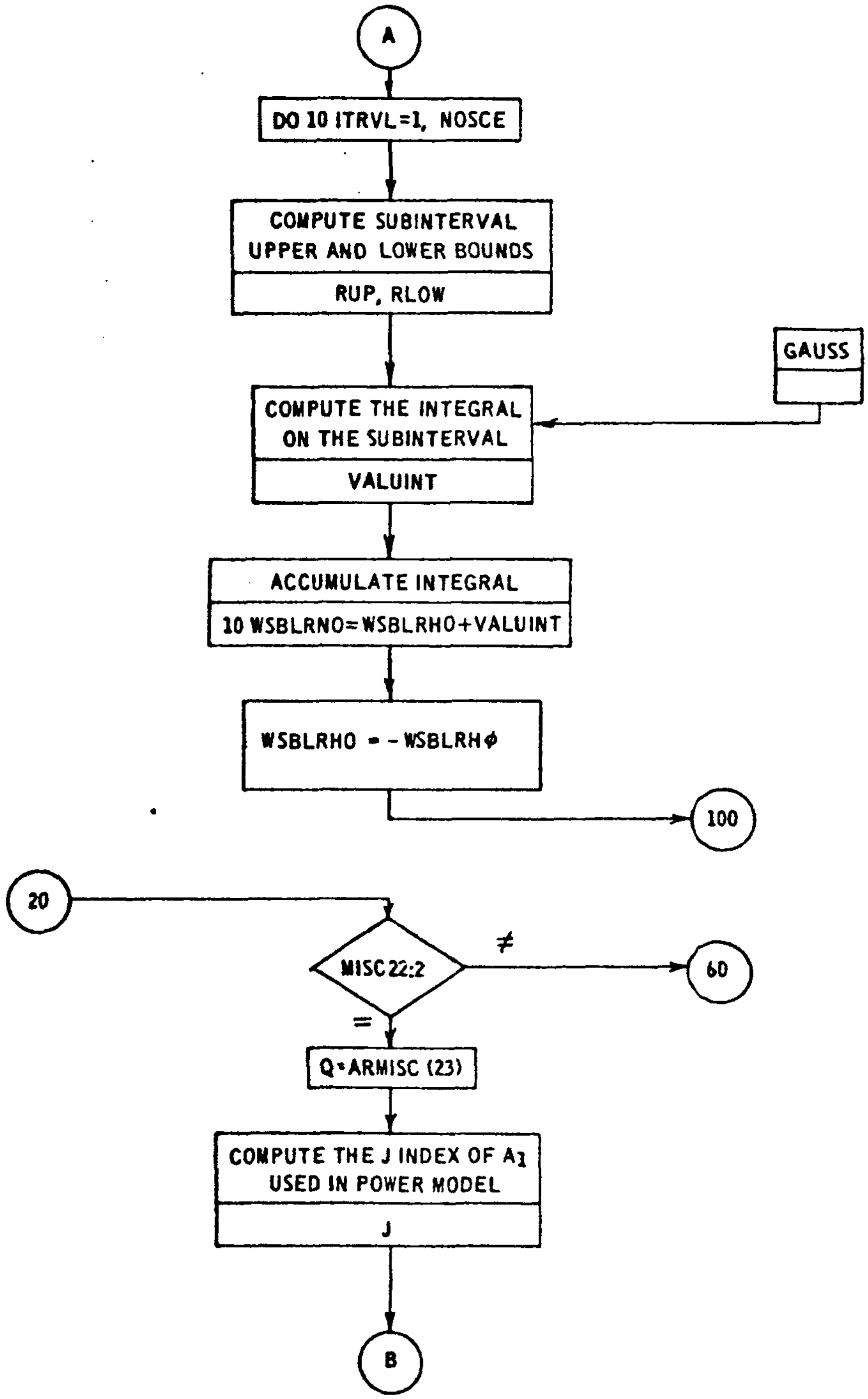
Usage: CALLING SEQUENCE

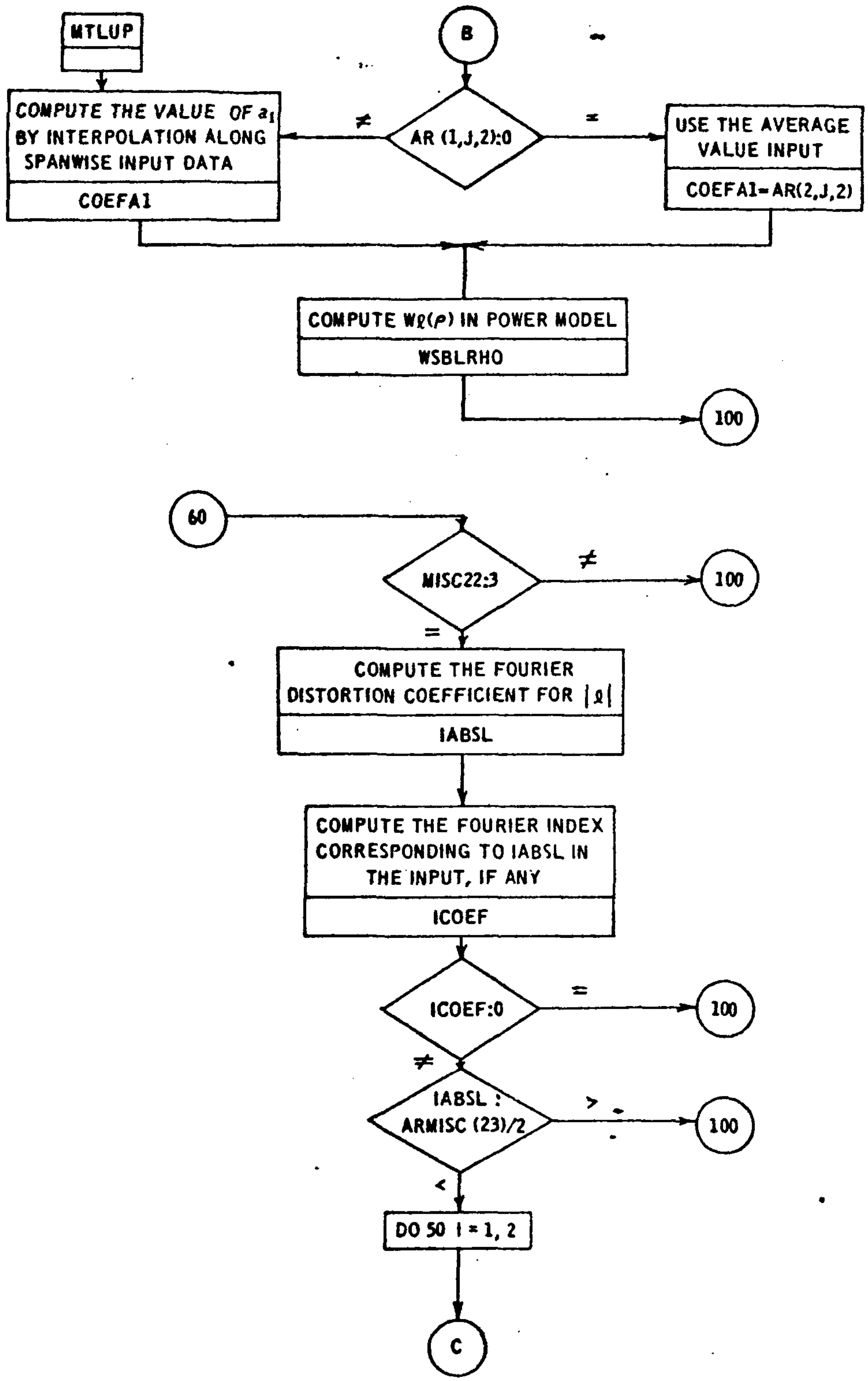
```
COMPLEX FACTIN3,VFACTIN
COMMON/CFACT/M,N,RMUMN,CAPNMN,ETA,SIGN,L,CAPKMN
DIMENSION ARMISC(NARMISC),AR(MAXDIM,MAXJ,3)
.
.
.
VFACTIN = FACTIN3(RHO,ARMISC,MAXDIM,MAXJ,AR)
```

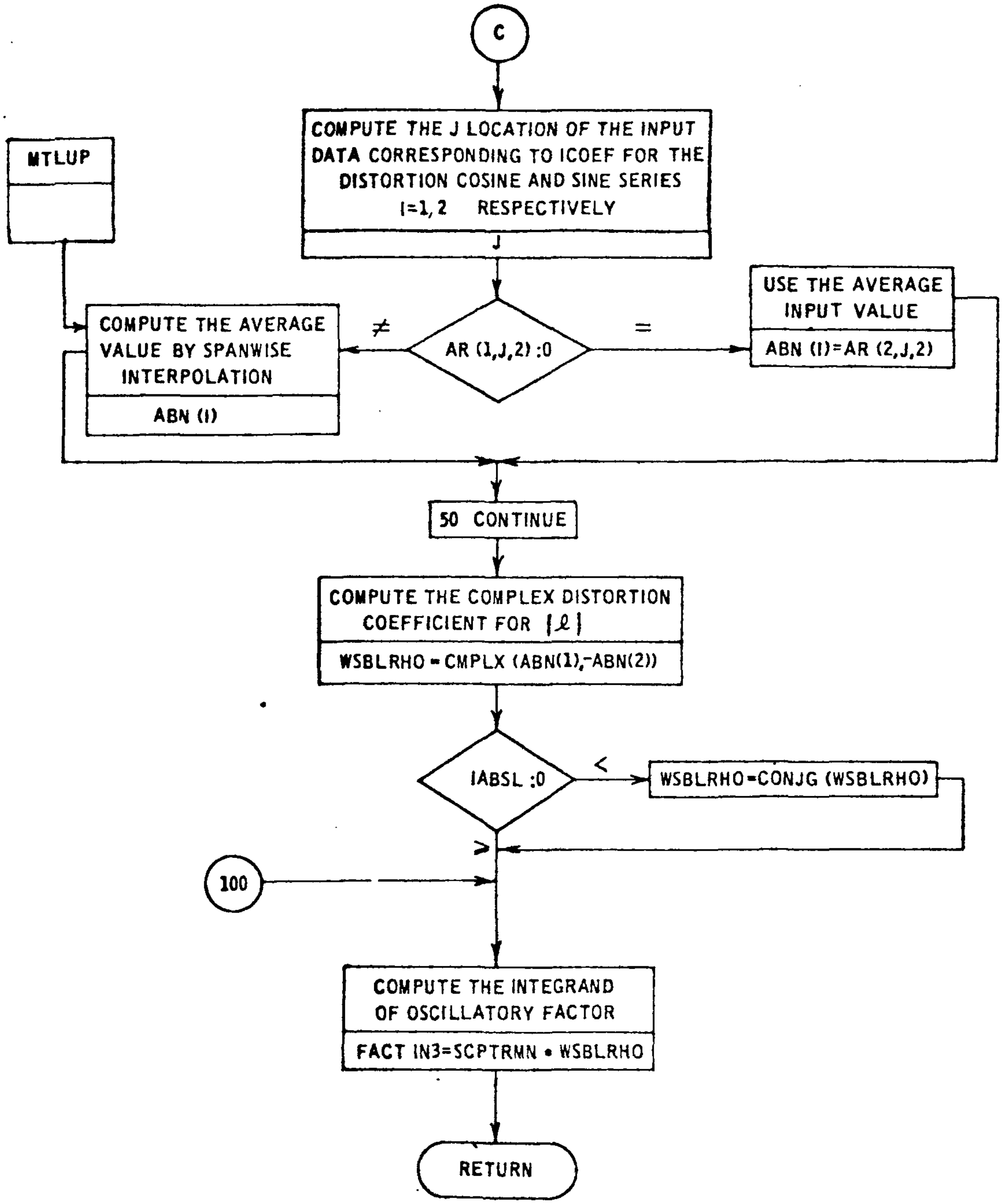
Timing: The timing is approximately equal to the time for a unit call to UNEGNFN plus, for the cone model, 2 x $|\ell|$ unit calls to GAUSS. For the power model, the timing is equal to one unit call to MTLUP and, for input values, two unit calls to MTLUP.

Accuracy: The accuracy is of the algorithmic type and is dominated by UNEGNFN and, for the cone model, GAUSS, and for the power model or direct Fourier coefficient input, MTLUP.


CALLING LIST
RHO,ARMISC,MAXDIM MAXJ , AR
START
UNEGNFN
EVALUATE THE EIGEN FUNCTION
SCPTRMN
INITIALIZE THE DISTORTION COEFFICIENT TO ZERO
WSBLRHO=(0.,0.)
L : 0
=
100
≠
MISC22=ARMISC (22)
MISC 22:1
≠
20
=
COMPUTE D, δ l,0 , δ l,1 + δ l,-1
DVALUE, DELTALO, DELTALI
CALCULATE NUMBER OF SUBINTERVALS AND SUB INTERVAL WIDTH
NOSCE, WIDTHI
A


A
DO 10 ITRVL=1, NOSCE
COMPUTE SUBINTERVAL UPPER AND LOWER BOUNDS
RUP, RLOW
GAUSS
COMPUTE THE INTEGRAL ON THE SUBINTERVAL
VALUINT
ACCUMULATE INTEGRAL
10 WSBLRNO=WSBLRHO+VALUINT
WSBLRHO = - WSBLRHϕ
100
20
MISC 22:2
≠
60
=
Q=ARMISC (23)
COMPUTE THE J INDEX OF A1 USED IN POWER MODEL
J
B


MTLUP
COMPUTE THE VALUE OF a_1 BY INTERPOLATION ALONG SPANWISE INPUT DATA
COEFA1
B
AR (1,J,2):0
≠
=
USE THE AVERAGE VALUE INPUT
COEFA1=AR(2,J,2)
COMPUTE Wℓ(ρ) IN POWER MODEL
WSBLRHO
100
60
MISC22:3
≠
100
=
COMPUTE THE FOURIER DISTORTION COEFFICIENT FOR |ℓ|
IABSL
COMPUTE THE FOURIER INDEX CORRESPONDING TO IABSL IN THE INPUT, IF ANY
ICOEF
ICOEF:0
=
100
≠
IABSL : ARMISC (23)/2
>
100
<
DO 50 I = 1, 2
C


C
COMPUTE THE J LOCATION OF THE INPUT DATA CORRESPONDING TO ICOEF FOR THE DISTORTION COSINE AND SINE SERIES I=1, 2 RESPECTIVELY
J
MTLUP
AR (1,J,2) :0
≠
=
COMPUTE THE AVERAGE VALUE BY SPANWISE INTERPOLATION
ABN (I)
USE THE AVERAGE INPUT VALUE
ABN (I)=AR (2,J,2)
50 CONTINUE
COMPUTE THE COMPLEX DISTORTION COEFFICIENT FOR |ℓ|
WSBLRHO = CMPLX (ABN(1),-ABN(2))
IABSL :0
<
>
WSBLRHO=CONJG (WSBLRHO)
100
COMPUTE THE INTEGRAND OF OSCILLATORY FACTOR
FACT IN3=SCPTRMN • WSBLRHO
RETURN

```
      COMPLEX FUNCTION FACTIN3(RHO,ARMISC,MAXDIM,MAXJ,AR)
C
C PURPOSE      EVALUATE THE INTEGRAND FACTIN3 TO BE CALLED BY THE
C              INTEGRATOR GAUSS2
C
C
C

      DIMENSION AR(MAXDIM,MAXJ,3),ARMISC(1),ABN(2)
      COMMON/CFACT/  M,N,RMUMN,CAPNMN,ETA,SIGN,L,CAPKMN
      COMPLEX WSBLRHO,VSBLRHO,DISINT,VALUINT
      COMMON/CDISINT/CAPADIS,RHOINC
      EXTERNAL DISINT
      DATA PI/3.14159265358979/
C
C              EVALUATE NORMALIZED EIGENFUNCTION
C
      SCPTRMN = UNEGNFN(M,RMUMN,ETA,RHO)/CAPNMN
C
C              COMPUTE THE DISTORTION FACTOR
C
      WSBLRHO = (0.,0.)
      IF(L.EQ.0) GO TO 100
      IF( ARMISC(22) .EQ. 0. )  GO TO 100
C
C              COMPUTE THE FOURIER COEFFICIENTS FROM CONE MODE
C
      IF( ARMISC(22) .NE. 1. )  GO TO 20
      RHOINC = RHO
      VADBV1 = ARMISC(23)
      CAPADIS= ARMISC(24)
      DVALUE = (1.-VADBV1)/(CAPADIS**2-1.)
      DELTALO= 0.
      IF(L.EQ.0)DELTALO=1.
      DELTAL1= 0.
      IABSL  = IABS(L)
      IF(IABSL.EQ.1)DELTAL1=1.
      IORDGS = 2
      NOSCE  = MAX0(2,2*IABSL)
      WIDTHI = 2.*PI/NOSCE
      DO 10 ITRVL=1,NOSCE
      RLOW  = (ITRVL-1)*WIDTHI
      RUP = RLOW + WIDTHI
      CALL GAUSS(RLOW,RUP,VALUINT,DISINT,IORDGS)
   10 WSBLRHO = WSBLRHO + VALUINT
      WSBLRHO = (1.+DVALUE)*DELTALO - .5*CAPADIS*RHO*DELTAL1*DVALUE
     1          - .5*DVALUE*WSBLRHO/PI
      WSBLRHO =     -2.*PI*WSBLRHO
      GO TO 100
C
```

```
C               COMPUTE THE FOURIER COEFFICIENT FROM POWER MODEL
C
 20    IF( ARMISC(22) .NE. 2. )  GO TO 60
        Q = ARMISC(23)
        IABSL = IABS(L)
       J = 9 + ARMISC(20) *2 +1
       NSPN = AR(1,J,2)
       IF( NSPN ) 40,30,40
   30   COEFA1 = AR(2,J,2)
       GO TO 50
   40  IPA = -1
       CALL MTLUP(RHO,COEFA1,1,NSPN,NSPN,1,IPA,AR(3,1,2),AR(3,J,2) )
   50  WSBLRHO = .5*COEFA1/FLOAT(IABSL)**Q
       GO TO 100
C
C               COMPUTE THE FOURIER COEFFICIENT FROM INPUT VALUES
C
 60    IF( ARMISC(22) .NE. 3. )  GO TO 100
       IABSL = IABS(L)
       MAXCOEF = ARMISC(23)/2
       MULTFCT  = ARMISC(24)
       DO 65 NCOEF=1,MAXCOEF
       IF(IABSL.NE.NCOEF*MULTFCT) GO TO 65
       ICOEF = NCOEF
       GO TO 67
   65  CONTINUE
       GO TO 100
   67  DO 90 IAB=1,2
       J = 9 + ARMISC(20) + 2      + 2*(ICOEF-1) + IAB
       NSPN = AR(1,J,2)
       IF( NSPN ) 80,70,80
   70  ABN(IAB)=AR(2,J,2)
       GO TO 90
   80  IPA = -1
       CALL MTLUP(RHO,ABN(IAB),1,NSPN,NSPN,1,IPA,AR(3,1,2),AR(3,J,2))
   90  CONTINUE
       WSBLRHO = .5*CMPLX( ABN(1),-ABN(2) )
       IF( L.LT.0 ) WSBLRHO = CONJG( WSBLRHO)
C
C               COMBINE TO FORM FACTIN3
C
  100  FACTIN3 = SCPTRMN*WSBLRHO
C
       RETURN
       END
```

3.2.9 Function FACTIN4

Purpose: This function evaluates the oscillatory factor of subroutine BBCAA.

If ARMISC(25) = 3 (i.e., LIFTFN3 or NONCPT is specified):

$$\text{FACTIN4} = \mathcal{R}_m\left(\mu_{mn}\rho\right)\left(g_1(\rho)F_1(\rho)-g_2(\rho)F_2(\rho)\right),$$

where:

$$g_j(\rho) = 2\pi P_j \cdot e^{-\frac{(\rho-R)^2}{2a_j^2}} \cdot I_\ell\left(\frac{\rho R}{a_j^2}\right) e^{-\frac{\rho R}{a_j^2}} e^{-i\ell\phi}$$

$$P_j = U_j \begin{cases} \dfrac{2\ BT_j\ \text{SIN}\ (B\tau)}{\sqrt{2\pi}\ B\tau} & \text{if} \quad BT_j < .1 \\ E_j & \text{if} \quad .1 \le BT_j \le 10 \\ e^{-\frac{1}{2}\left(\frac{\tau}{T_j}\right)^2} & \text{if} \quad 10 < BT_j \end{cases}$$

$$T_j = \frac{L_j}{M_{z,2}} \quad , \quad E_j = \frac{T_j}{\sqrt{2\pi}} \int_{-B}^{B} e^{-\frac{(\omega T_j)^2}{2}} \text{COS}\ (\omega\ \tau)\ d\ \omega$$

If ARMISC(38) = 0, then:

$$F_1(\rho) = M_{M,2}(\rho)\ \text{SIN}\left(\Theta_2(\rho)\right)S\left(\nu_\ell(\rho)\right) - \alpha(\rho)M_{Z,2}(\rho)F_\alpha\left(\nu_\ell(\rho)\right)$$
$$-f(\rho)M_{Z,2}(\rho)F_f\left(\nu_\ell(\rho)\right)$$

$$F_2(\rho) = M_{Z,2}(\rho)S\left(\nu_\ell\ (\rho)\right) + \alpha(\rho)M_{M,2}(\rho)\text{SIN}\left(\Theta_2(\rho)\right)F_\alpha\left(\nu_\ell(\rho)\right)$$
$$+f(\rho)M_{M,2}(\rho)\text{SIN}\left(\Theta_2(\rho)\right)F_f\left(\nu_\ell(\rho)\right)$$

If ARMISC(38) $\neq$ 0, then:

$$F_1(\rho) = M_{M,2}(\rho)\,SIN\left(\Theta_2(\rho)\right)S\left(\nu_\ell(\rho)\right)J\left(\kappa^{\pm}_{mn\sigma}\right) - \alpha(\rho)M_{Z,2}(\rho)\;J\left(\nu_\ell + \kappa^{\pm}_{mn\sigma}\right)$$

$$-f(\rho)M_{Z,2}(\rho)\left\{J\left(\kappa^{\pm}_{mn\sigma}\right)F\left(\nu_\ell\right) + \frac{2J_1\left(\nu_\ell + \kappa^{\pm}_{mn\sigma}\right)}{\nu_\ell + \kappa^{\pm}_{mn\sigma}}\right.$$

$$\left. - \frac{2}{\nu_\ell}\sum_{j=1}^{\infty}(-1)^j\;J_1\left(\nu_\ell\right)\left[J_{j+1}\left(\kappa^{\pm}_{mn\sigma}\right) + J_{j-1}\left(\kappa^{\pm}_{mn\sigma}\right)\right]\right\}$$

$$F_2(\rho) = M_{Z,2}(\rho)S\left(\nu_\ell(\rho)\right) + \alpha(\rho)M_{M,2}(\rho)\,SIN\left(\Theta_2(\rho)\right)J\left(\nu_\ell + \kappa^{\pm}_{mn\sigma}\right)$$

$$+f(\rho)M_{M,2}(\rho)\,SIN\left(\Theta_2(\rho)\right)\left\{J_\ell\left(\kappa^{\pm}_{mn\sigma}\right)F\left(\nu_\ell\right) + \frac{2J_1\left(\nu_\ell + \kappa^{\pm}_{mn\sigma}\right)}{\nu_\ell + \kappa^{\pm}_{mn\sigma}}\right.$$

$$\left. - \frac{2}{\nu_\ell}\sum_{j=1}^{\infty}(-1)^j\;J_1\left(\nu_\ell\right)\left[J_{j+1}\left(\kappa^{\pm}_{mn\sigma}\right) + J_{j-1}\left(\kappa^{\pm}_{mn\sigma}\right)\right]\right\}$$

$$M_{M,2}(\rho) = \sqrt{\frac{1}{4}\left(\sqrt{M^2_{I,2}(\rho) - M^2_{Z,2}(\rho)} + \sqrt{M^2_{E,2}(\rho) - M^2_{Z,2}(\rho)}\right)^2 + M^2_{Z,2}(\rho)}$$

$$SIN\left(\Theta_2(\rho)\right) = \frac{\sqrt{M^2_{I,2}(\rho) - M^2_{Z,2}(\rho)} + \sqrt{M^2_{E,2}(\rho) - M^2_{Z,2}(\rho)}}{2 \cdot M_{M,2}(\rho)}$$

$$\nu_\ell = \frac{C_2(\rho)}{2\,\rho}\;\ell \cdot SIN\left(\Theta_2(\rho)\right)$$

If ARMISC(25) = 4 (i.e., LIFTFN4 is specified):

$$FACTIN4 = 2\sqrt{2\pi}\ \mathcal{R}_m\left(\mu_{mn}\rho\right) e^{-i\ell\phi}$$

$$\left\{M_{M,2}(\rho)\ \mathrm{SIN}\left(\Theta_2(\rho)\right) P_1\ I_\ell\left(\frac{\rho R}{a_1^2}\right) e^{-\rho R/a_1^2}\ I_1\right.$$

$$\left. -M_{Z,2}(\rho)\ P_2\ I_\ell\left(\frac{\rho R}{a_2^2}\right) e^{-\rho R/a_2^2}\ I_2\right\}$$

where:

$$I_j = a_j\ \mathrm{Re}\left\{\int_0^\infty e^{-i(\rho-R)K}\ T^*\left(h(\rho,K),\ \psi(\rho,K)\right) e^{-K^2a_j^2/2}\ dK\right\}$$

LIFTFN4 calculates $T(h,\psi)$

$$h(\rho,K) = \frac{C_2(\rho)}{2}\sqrt{K^2 + \left[\frac{\ell}{\rho}\ \mathrm{SIN}\ \Theta_2(\rho)\right]^2}$$

$$\psi(\rho,K) = \mathrm{COS}^{-1}\left(\frac{C_2(\rho)K}{2h(\rho,K)}\right) = \mathrm{COS}^{-1}\left(\frac{1}{\sqrt{1+\left(\frac{\ell}{K\rho}\ \mathrm{SIN}\ \Theta_2(\rho)\right)^2}}\right)$$

$$= \mathrm{TAN}^{-1}\left(\frac{\ell}{K\rho}\ \mathrm{SIN}\ \Theta_2(\rho)\right)$$

Method: The procedure is as follows:

1) Evaluate normalized eignfunction, $\mathcal{R}_m(\mu_{mn})$.

2) Compute BT_j.

3) If $BT_j < .1$, calculate P_j as in equation above and go to step 6.

4) If $.1 \leq BT_j \leq 10$, then calculate E_j by using subroutine GAUSS with $[2\tau B/\pi] + 1$ subintervals of the interval $[0,B]$ (the integrand is an even function so the interval $[0,B]$ was used). Calculate P_j and go to step 6.

5) If $10 < BT_j$, calculate P_j.

6) Calculate $I_\ell\ (\rho R/a_j^2)\ e^{-\rho R/a_j^2}$.

7) Calculate $e^{-i\ell\Phi}$.

8) Obtain $M_{I,2}(\rho)$, $M_{E,2}(\rho)$, $M_{z,2}(\rho)$, $C_2(\rho)$ from array AR using linear interpolation, if necessary.

9) If ARMISC(25) = 3 (i.e., LIFTFN3 or NONCPT is specified), obtain $f\ (\rho)$, $\alpha(\rho)$ from array AR.

10) Calculate $M_{M,2}(\rho)$ and $\sin\theta_2(\rho)$.

11) If ARMISC(25) = 3 (i.e., LIFTFN3 or NONCPT is specified), calculate g_j; use subroutine LIFTFN3 or NONCPT F_1 and F_2. Calculate FACTIN4 and return.

12) If ARMISC(25) = 4 (i.e., LIFTFN4 is specified), calculate I_j using "trapezoidal rule open at upper limit" as given on the following page:

$$I_j = a_j \, \mathrm{Re}\left\{ \int_0^\infty e^{-i(\rho-R)K} \; T^*(h,\psi) \; e^{-K^2 a_j^2/2} \, dK \right\}$$

$$= \mathrm{Re}\left\{ a_j \int_0^\infty e^{-i(\rho-R)\, x/a} \, T^*(h,\psi) \; e^{-x^2/2} \, \frac{1}{a_j} \, dx \right\}$$

(Let x = Ka)

$$= \mathrm{Re}\left\{ \int_0^\infty e^{-i(\rho-R)\, x/a_j} \, T^*\Big(h(\rho, x/a_j),\ \psi(\rho, x/a_j)\Big) \, e^{-x^2/2} \, dx \right\}$$

$$\approx \Delta \sum_{K=0}^{K_{MAX}}{}' \; \mathrm{Re}\left\{ e^{-i(\rho-R)} \left(K\Delta/a_j\right) e^{(k\Delta)^2/2} \; T^*\Big(h\left(\rho, K\Delta/a_j\right), \psi\left(\rho, K\Delta/a_j\right)\Big) \right\}$$

where:

$$\Delta = \frac{a_j}{R}, \quad K_{MAX} = \frac{20}{\Delta} + 1, \text{ and } \sum_{K=0}^{N}{}' \; a_K = \frac{1}{2} a_0 + a_1 + \cdots + a_N;$$

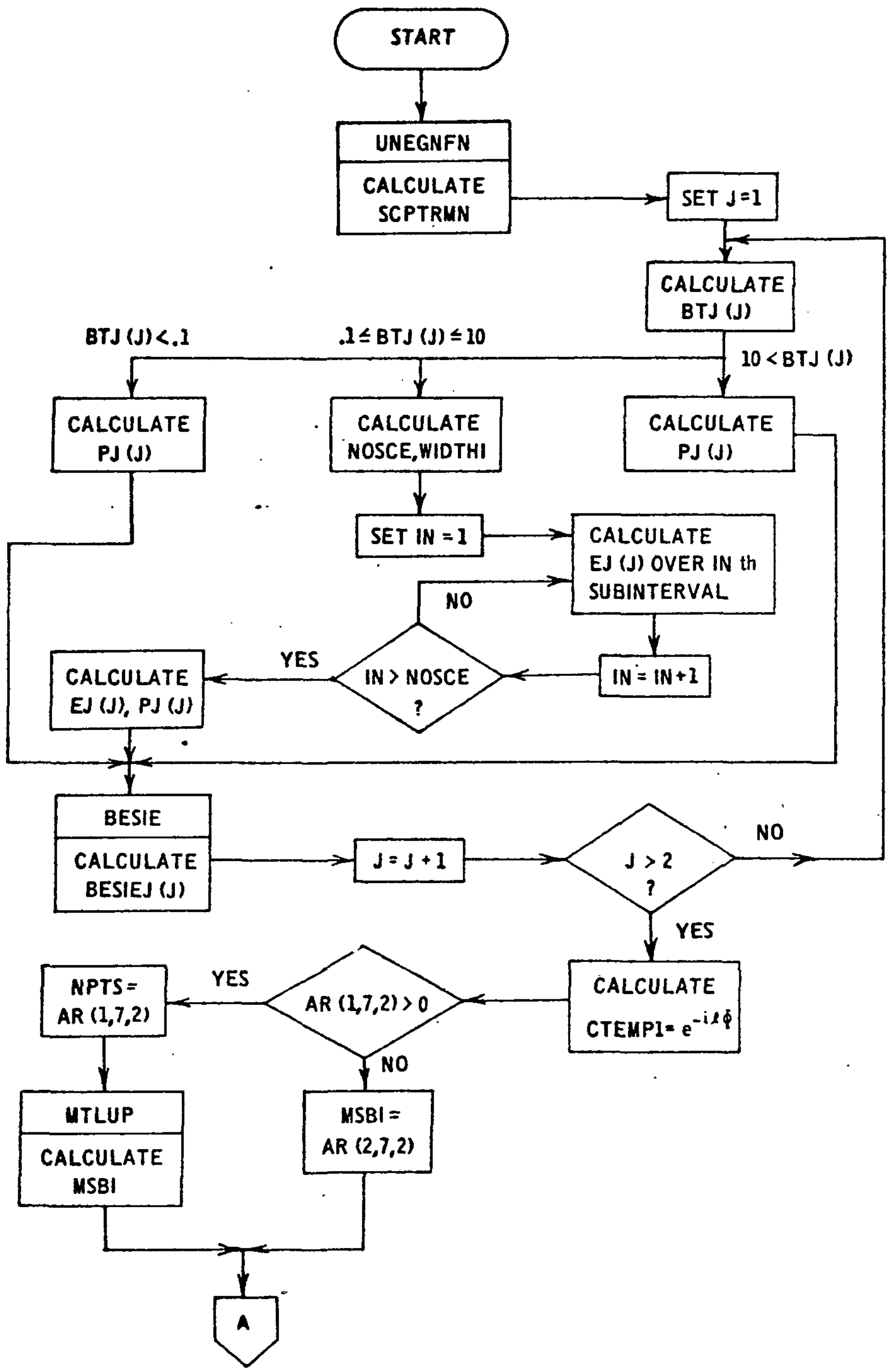
calculate FACTIN4 and return.

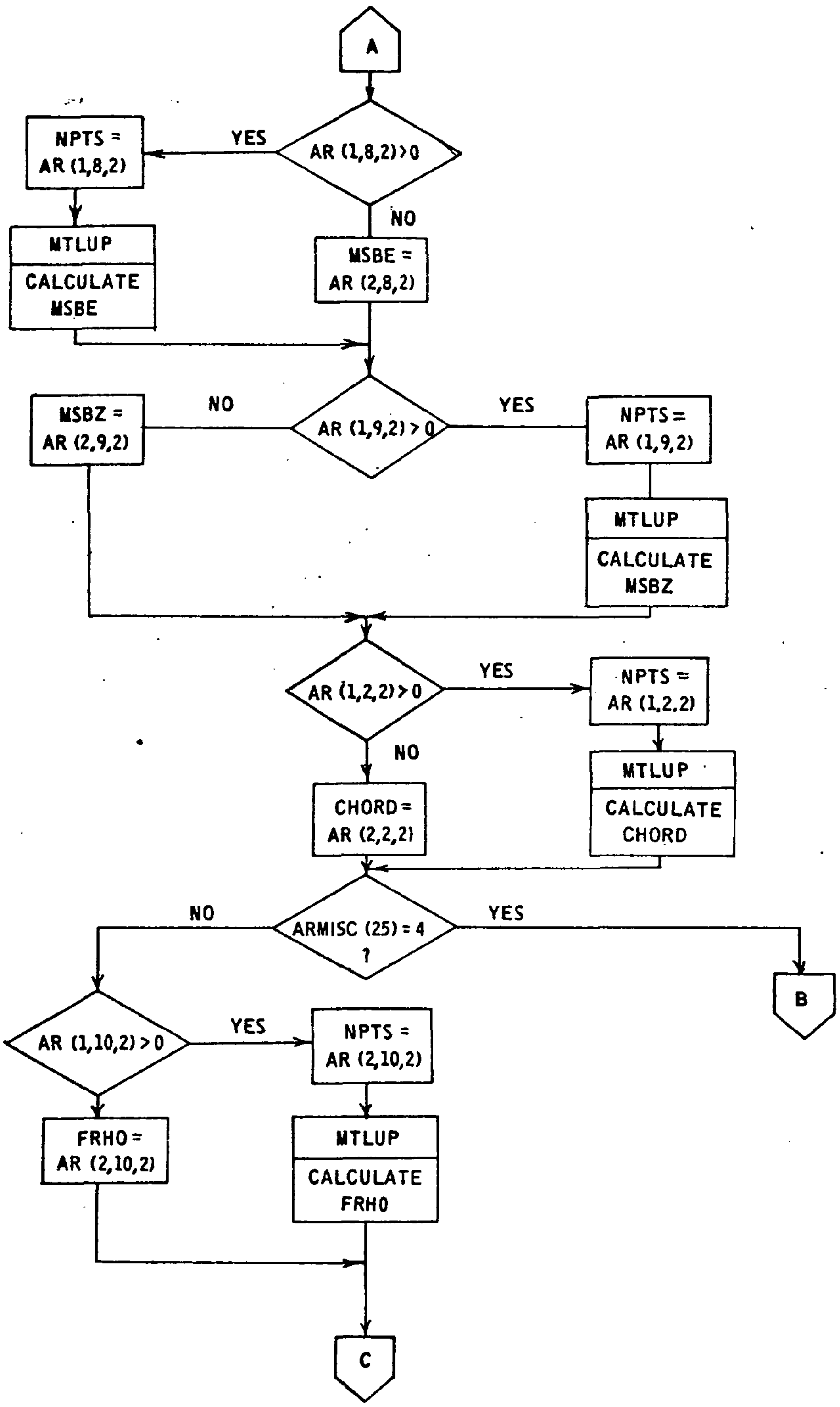
Usage: CALLING SEQUENCE

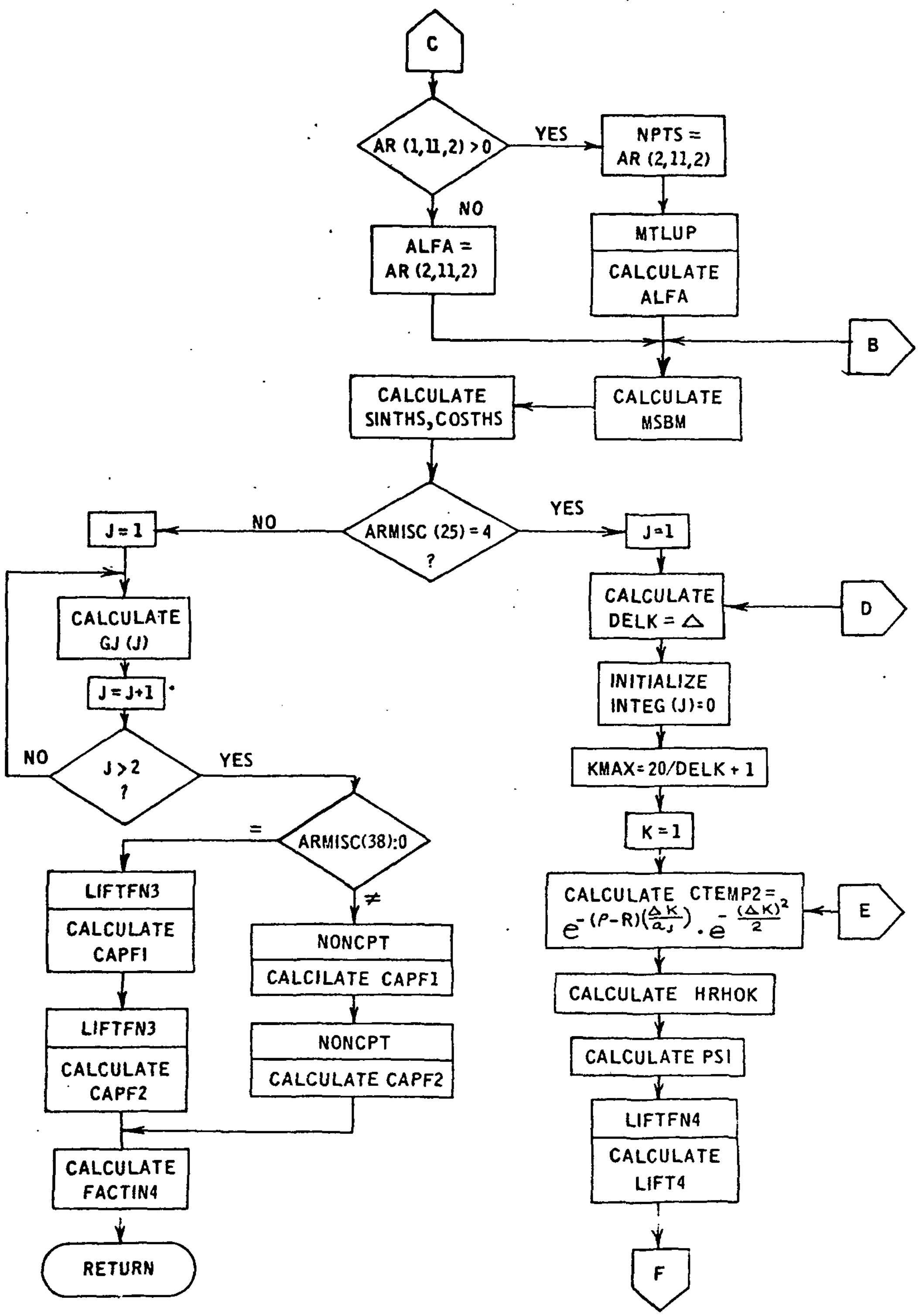
```
COMPLEX FACTIN4,Z
DIMENSION ARMISC(40),AR(MAXDIM,MAXJ,3)
COMMON/CFACT/M,N,RMUMN,CAPNMN,ETA,SIGN,L,CAPKMN
COMMON/SCRATCH/BES(1000)
.
.
.
Z = FACTIN4(RHO,ARMISC,MAXDIM,MAXJ,AR)
```

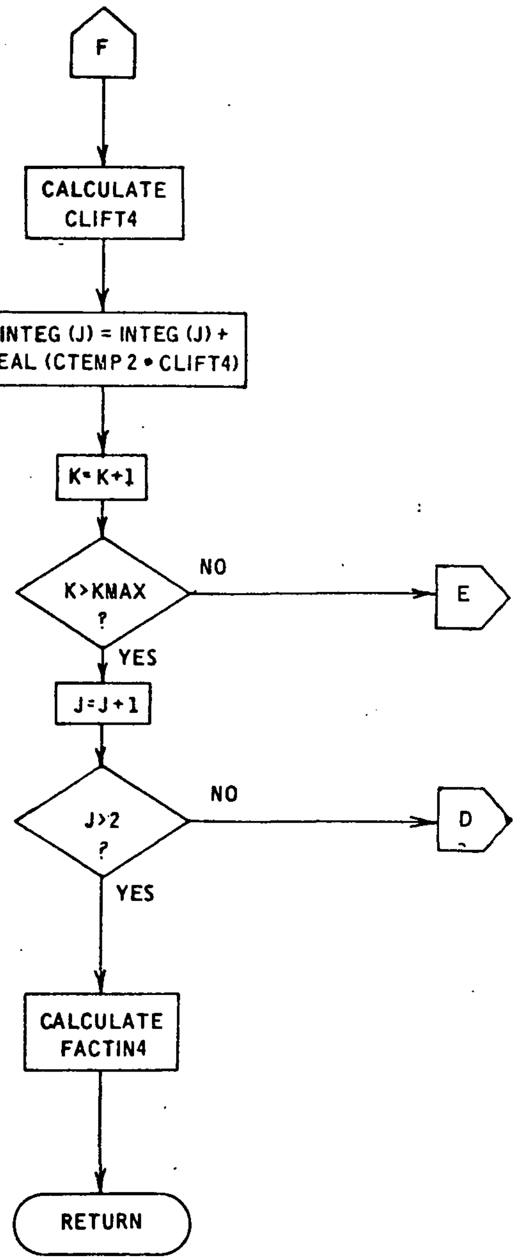
Accuracy: If ARMISC(25) = 3, the accuracy is dominated by the accuracy of subroutines UNEGNFN and LIFTFN3 or NONCPT.

If ARMISC(25) = 4, the accuracy is dominated by the accuracy of subroutine LIFTFN4 and the truncated trapezoidal rule used to calculate I_j.


START
UNEGNFN
CALCULATE SCPTRMN
SET J=1
CALCULATE BTJ (J)
BTJ (J)<.1
.1≤ BTJ (J) ≤10
10 < BTJ (J)
CALCULATE PJ (J)
CALCULATE NOSCE,WIDTHI
CALCULATE PJ (J)
SET IN = 1
CALCULATE EJ (J) OVER IN th SUBINTERVAL
NO
YES
CALCULATE EJ (J), PJ (J)
IN > NOSCE ?
IN = IN+1
BESIE
CALCULATE BESIEJ (J)
J = J + 1
J > 2 ?
NO
YES
NPTS= AR (1,7,2)
YES
AR (1,7,2) > 0
CALCULATE CTEMP1= e^-iℓφ
NO
MTLUP
CALCULATE MSBI
MSBI = AR (2,7,2)
A


A
AR (1,8,2)>0
YES
NO
NPTS =
AR (1,8,2)
MTLUP
CALCULATE
MSBE
MSBE =
AR (2,8,2)
AR (1,9,2) > 0
NO
YES
MSBZ =
AR (2,9,2)
NPTS =
AR (1,9,2)
MTLUP
CALCULATE
MSBZ
AR (1,2,2) > 0
YES
NO
NPTS =
AR (1,2,2)
MTLUP
CALCULATE
CHORD
CHORD =
AR (2,2,2)
ARMISC (25) = 4
?
NO
YES
B
AR (1,10,2) > 0
YES
NPTS =
AR (2,10,2)
FRHO =
AR (2,10,2)
MTLUP
CALCULATE
FRHO
C


C
AR (1,11,2) > 0
YES
NPTS =
AR (2,11,2)
NO
ALFA =
AR (2,11,2)
MTLUP
CALCULATE
ALFA
B
CALCULATE
SINTHS,COSTHS
CALCULATE
MSBM
J = 1
NO
ARMISC (25) = 4
?
YES
J = 1
CALCULATE
DELK = △
D
CALCULATE
GJ (J)
J = J+1
INITIALIZE
INTEG (J) = 0
NO
J > 2
?
YES
KMAX = 20/DELK + 1
K = 1
=
ARMISC(38):0
≠
LIFTFN3
CALCULATE
CAPFI
CALCULATE CTEMP2 =
E
NONCPT
CALCILATE CAPF1
CALCULATE HRHOK
LIFTFN3
CALCULATE
CAPF2
NONCPT
CALCULATE CAPF2
CALCULATE PSI
LIFTFN4
CALCULATE
LIFT4
CALCULATE
FACTIN4
RETURN
F


F
CALCULATE CLIFT4
INTEG (J) = INTEG (J) + REAL (CTEMP 2 • CLIFT4)
K= K+1
K>KMAX ?
NO
E
YES
J=J+1
J>2 ?
NO
D
YES
CALCULATE FACTIN4
RETURN

```
      COMPLEX FUNCTION FACTIN4(RHO,ARMISC,MAXDIM,MAXJ,AR)
C
C PURPOSE       EVALUATE THE INTEGRAND FACTIN4 TO BE CALLED BY THE
C               INTEGRATOR GAUSS2
C
C
      REAL  INTEGJ,MSBE,MSBI,MSBM,MSBT,MSBZ
      COMPLEX CAPF1,CAPF2,CLIFT4,CTEMP1,CTEMP2,FUNIN4,GJ(2),LIFT4,
     1        VALUINT
      DIMENSION AR(MAXDIM,MAXJ,3),ARMISC(1),ABN(2),EJ(2),INTEGJ(2)
      DIMENSION BESIEJ(2),BTJ(2),PJ(2),TJ(2)
      COMMON/CFACT/  M,N,RMUMN,CAPNMN,ETA,SIGN,L,CAPKMN
      COMMON/CFUNIN4/CTJ,TAU
      COMMON/SCRATCH/BES(1000)
      EXTERNAL FUNIN4
      DATA PI,TWOPI/ 3.14159265358979, 6.28318530717959/
      DATA SQRT2PI/2.506628274631/

               EVALUATE NORMALIZED EIGENFUNCTION

      SCPTRMN = UNEGNFN(M,RMUMN,ETA,RHO)/CAPNMN

 10   DO 50 J=1,2
      TJ(J) = ARMISC(33+J) / AR(2,9,2)
      BTJ(J) = ARMISC(36)*TJ(J)
      IF( BTJ(J)..LE. 10.  )  GO TO 20
      TEMP1 = -ARMISC(37)*ARMISC(37) / (2.*TJ(J)*TJ(J))
      PJ(J) = ARMISC(29+J)*EXP(TEMP1)
      GO TO 40
      IF( BTJ(J) .LT. .1    )  GO TO 30
      IORDGS = 2
      TAU = ARMISC(37)
      CTJ = TJ(J)
      EJ(J) = 0.
      NOSCE = 2*IFIX( TAU*ARMISC(36) / PI ) + 1
      WIDTHI = ARMISC(36) / NOSCE
      DO 25 IN=1,NOSCE
      RLOW = FLOAT(IN-1)*WIDTHI
      RUP = RLOW + WIDTHI
      CALL GAUSS(RLOW,RUP,VALUINT,FUNIN4,IORDGS)
      EJ(J) = EJ(J) + REAL(VALUINT)
 25   CONTINUE
      EJ(J) = (2./SQRT2PI)*TJ(J)*EJ(J)
      PJ(J) = ARMISC(29+J)*EJ(J)
      GO TO 40
 30   BTAJ = ARMISC(36)*ARMISC(37)
      IF(BTAU .EQ. 0.)   TEMP1 = 1.
      IF(BTAU .NE. 0.)   TEMP1 = SIN(BTAU)/BTAU
      PJ(J) = ARMISC(29+J)*2.*BTJ(J)*TEMP1 / SQRT2PI
 40   TEMP1 = RHO*ARMISC(23) / (ARMISC(31+J)*ARMISC(31+J))
      CALL BESIE(L,TEMP1,BESIEJ(J))
 50   CONTINUE
```

```
      CTEMP1 = CMPLX(0.,-L*ARMISC(29))
      CTEMP1 = CEXP(CTEMP1)
      IP = -1
      IF( AR(1,7,2) .GT. 0. )  GO TO 60
      MSBI = AR(2,7,2)
      GO TO 70
   60 NPTS = AR(1,7,2)
      CALL MTLUP(RHO,MSBI,1,NPTS,NPTS,1,IP,AR(3,1,2),AR(3,7,2))
   70 IF( AR(1,8,2) .GT. 0. )  GO TO 80
      MSBE = AR(2,8,2)
      GO TO 90
   80 NPTS = AR(1,8,2)
      CALL MTLUP(RHO,MSBE,1,NPTS,NPTS,1,IP,AR(3,1,2),AR(3,8,2))
      IF( AR(1,9,2) .GT. 0. ) GO TO 100
      MSBZ = AR(2,9,2)
      GO TO 110
      NPTS = AR(1,9,2)
      CALL MTLUP(RHO,MSBZ,1,NPTS,NPTS,1,IP,AR(3,1,2),AR(3,9,2))
      IF(AR(1,2,2) .GT. 0.)   GO TO 120
      CHORD = AR(2,2,2)
      GO TO 130
  120 NPTS = AR(1,2,2)
      CALL MTLUP(RHO,CHORD,1,NPTS,NPTS,1,IP,AR(3,1,2),AR(3,2,2))
      IF( ARMISC(25) .EQ. 4. )  GO TO 170
      IF(AR(1,10,2) .GT. 0.)  GO TO 140
      FRHO = AR(2,10,2)
      GO TO 150
      NPTS = AR(1,10,2)
      CALL MTLUP(RHO,FRHO,1,NPTS,NPTS,1,IP,AR(3,1,2),AR(3,10,2))
      IF(AR(1,11,2) .GT. 0.)  GO TO 160
      ALFA = AR(2,11,2)
      GO TO 170
  160 NPTS = AR(1,11,2)
      CALL MTLUP(RHO,ALFA,1,NPTS,NPTS,1,IP,AR(3,1,2),AR(3,11,2))
  170 IF(MSBI .LE. MSBZ) TEMP1 = 0.
      IF(MSBI .GT. MSBZ) TEMP1 = SQRT(MSBI*MSBI - MSBZ*MSBZ)
      IF(MSBE .LE. MSBZ) TEMP2 = 0.
      IF(MSBE .GT. MSBZ) TEMP2 = SQRT(MSBE*MSBE - MSBZ*MSBZ)
      TEMP3 = .25*(TEMP1 + TEMP2)*(TEMP1 + TEMP2) + MSBZ*MSBZ
      MSBM  = SQRT(TEMP3)
      SINTHS = (TEMP1 + TEMP2) / (2.*MSBM)
      COSTHS = SQRT( 1.-SINTHS**2 )
      COTTHS = COSTHS / SINTHS
      IF( ARMISC(25) .EQ. 4. )   GO TO 190
      DO 180 J=1,2
      TEMP1 = -(RHO-ARMISC(28))*(RHO-ARMISC(28)) /
     1         (2.*ARMISC(31+J)*ARMISC(31+J))
  180 GJ(J) = TWOPI*PJ(J)*EXP(TEMP1)*BESIEJ(J)*CTEMP1
      MSBT = ARMISC(7)
      RNU = CHORD*L*MSBT / (2.*MSBM)
      B1 = 1.
      B2 = -ALFA*COTTHS
```

```
      B3 =  -FRHO*COTTHS
      IF(ARMISC(38).EQ.0.) CALL LIFTFN3(RNU,B1,B2,B3,CAPF1)
      IF(ARMISC(38).NE.0.) CALL NONCPT(B1,B2,B3,CHORD,CAPKMN,COSTHS,M,
     1                                RHO,RNU,SINTHS,CAPF1)
      B1 = COTTHS
      B2 = ALFA
      B3 = FRHO
      IF(ARMISC(38).EQ.0.) CALL LIFTFN3(RNU,B1,B2,B3,CAPF2)
      IF(ARMISC(38).NE.0.) CALL NONCPT(B1,B2,B3,CHORD,CAPKMN,COSTHS,M,
     1                                RHO,RNU,SINTHS,CAPF2)
      FACTIN4 = SCPTRMN*(GJ(1)*CAPF1 - GJ(2)*CAPF2)*MSBM*SINTHS
      RETURN
  190 IFORM = 2
      FKMAX = 20.
      DO 250 J=1,2
      DELK = ARMISC(31+J) / ARMISC(28)
      INTEGJ(J) = .5
      KMAX = FKMAX/DELK + 1.
      DO 240 K=1,KMAX
      TEMP1 = DELK*K / ARMISC(31+J)
      TEMP2 = -(RHO - ARMISC(28))*TEMP1
      CTEMP2 = CMPLX(0.,TEMP2)
      CTEMP2 = CEXP(CTEMP2)*EXP(-.5*(DELK*K)**2)

              COMPUTE FILOTAS L. R. F.

      TEMP2 = TEMP1**2 + ( L*SINTHS / RHO )**2
      HRHOK = .5*CHORD*SQRT(TEMP2)
      PSI = ATAN(L*SINTHS / (RHO*TEMP1) )
      CALL LIFTFN4(HRHOK,PSI,IFORM,LIFT4,IERLFT4)

              ACCUMULATE INTEGRAL

      CLIFT4 = CONJG(LIFT4)
      INTEGJ(J) = INTEGJ(J) + REAL(CTEMP2*CLIFT4)
  240 CONTINUE
      INTEGJ(J) = DELK*INTEGJ(J)
  250 CONTINUE
      FACTIN4 = 2.*SQRT2PI*SCPTRMN*CTEMP1*
     1          ( MSBM*SINTHS*PJ(1)*BESIEJ(1)*INTEGJ(1)
     2           -MSBZ*PJ(2)*BESIEJ(2)*INTEGJ(2)        )
      RETURN
      END
```

3.2.10 Subroutine LIFTFN2

Purpose: This subroutine computes a generalized airfoil lift response function (see ref. 33). Subroutine LIFTFN3 computes the airfoil lift response to a simple harmonic gust "frozen" in the fluid (the Sears function), while this subroutine computes the corresponding response when the gust is simple harmonic but not frozen in the fluid. Both response functions arise in thin airfoil theory with two-dimensional, uniform, inviscid, incompressible flow.

The response function, which is the return variable LIFT, depends on ν and λ, where:

ν = reduced temporal frequency

λ = complex reduced spatial frequency

$$\text{LIFT} = \begin{cases} \overline{K_L(\nu,\lambda)} & \nu \geq 0 \\ K_L(-\nu, -\bar{\lambda}) & \nu < 0 \end{cases}$$

CONLIFT = $\overline{\text{LIFT}}$, where an overbar indicates complex conjugation, and

$$K_L(\nu,\lambda) = \begin{cases} \left[J_o(\lambda)-iJ_1(\lambda)\right]\dfrac{H_1^{(2)}(\nu)}{H_1^{(2)}(\nu)+i\,H_o^{(2)}(\nu)} + i\,\nu/\lambda\quad J_1\,(\lambda) & \text{if}\quad \nu\lambda \neq 0 \\ \dfrac{H_1^{(2)}(\nu)}{H_1^{(2)}(\nu) + iH_o^{(2)}(\nu)} + i\,\nu/2 & \text{if}\quad \nu \neq 0, \lambda = 0 \\ J_o(\lambda) - i\,J_1\,(\lambda) & \text{if}\quad \nu = 0, \lambda \neq 0 \\ 1 & \text{if}\quad \nu = 0, \lambda = 0 \end{cases}$$

$-100 \leq \nu \leq 100$ and λ is complex with $|\lambda| \leq 100$.

Method: The procedure is as follows:

1) If $\nu \geq 0$, go to step 3 to calculate $K_L(\nu,\lambda)$.

2) If $\nu < 0$, go to step 3 to calculate $K_L(-\nu,-\bar{\lambda})$.

3) If $\nu \neq 0$, calculate HANKEL =

$$\frac{H_1^{(2)}(\nu)}{H_1^{(2)}(\nu)} + i\, H_0^{(2)}(\nu)$$

4) If $\lambda \neq 0$, calculate BESJLAM = $J_0(\lambda) - i\, J_1(\lambda)$.

5) If $\nu \neq 0$ and $\lambda \neq 0$, calculate CAPKL = BESJLAM * HANKEL + $i\, \nu\, J_1(\lambda)/\lambda$ and go to step 9.

6) If $\nu \neq 0$ and $\lambda = 0$, calculate CAPKL = HANKEL + $i\, \nu/2$ and go to step 9.

7) If $\nu = 0$ and $\lambda \neq 0$, calculate CAPKL = BESJLAM and go to step 9.

8) If $\nu = 0$ and $\lambda = 0$, let CAPKL = 1 and go to step 9.

9) If step 3 reached from step 1, go to step 10.
If step 3 reached from step 2, go to step 11.

10) Calculate CONLIFT = CAPKL = $K_L(\nu, \lambda)$, LIFT = $\overline{\text{CONLIFT}}$ = $\overline{K_L(\nu, \lambda)}$, and return.

11) Calculate LIFT = CAPKL = $K_L(-\nu, -\bar{\lambda})$ and calculate CONLIFT = $\overline{\text{LIFT}}$ = $\overline{K_L(-\nu, -\bar{\lambda})}$.

Usage: CALLING SEQUENCE

```
COMPLEX CONLIFT,LAMDA,LIFT
•
•
•
CALL LIFTFN2(RNU,LAMDA,LIFT,CONLIFT)
```

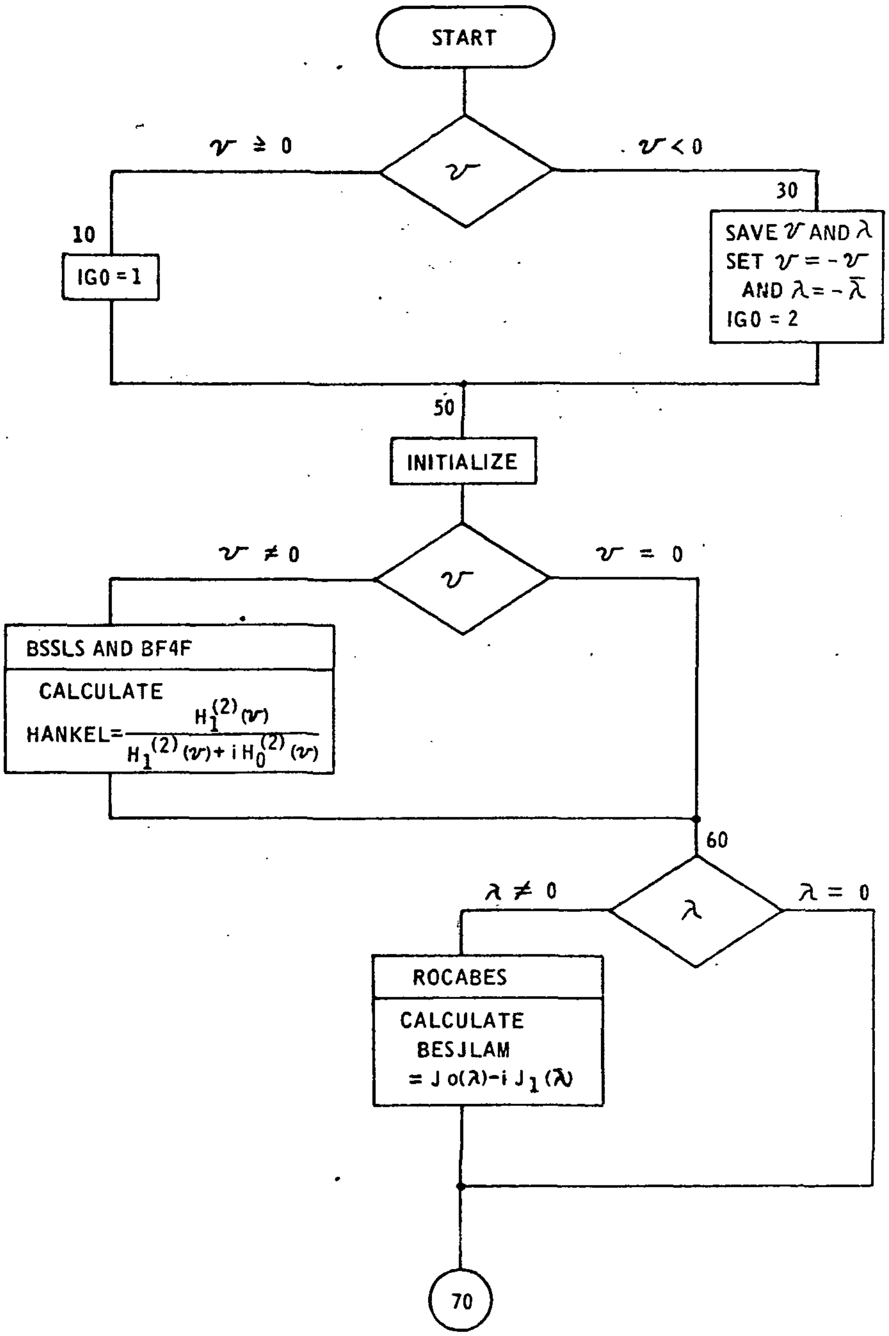
Common Blocks: SCRATCH

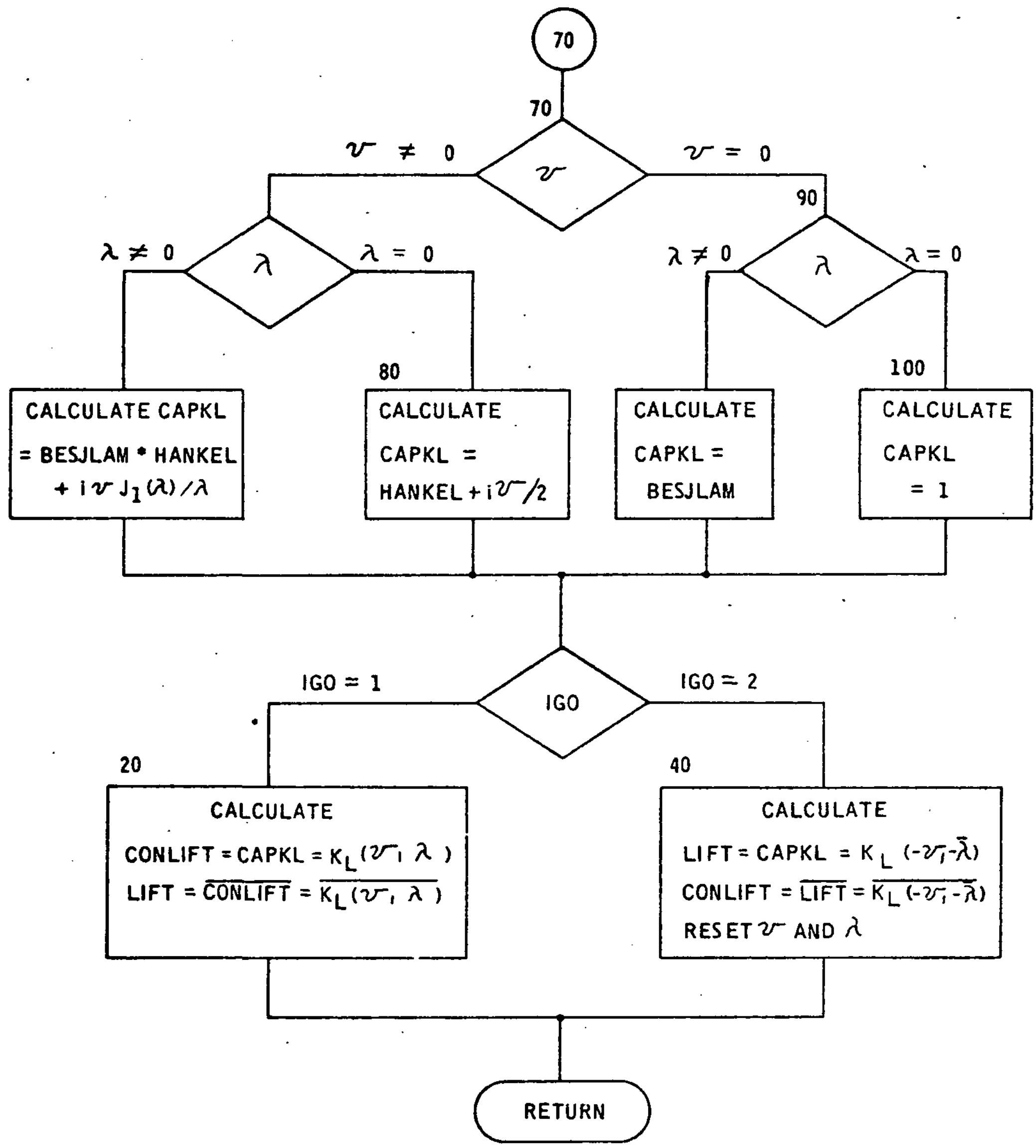
Restrictions: $-100 \le RNU \le 100$ and $|LAMDA| \le 100$

Timing: The average time over 1400 calls to LIFTFN2 is .015 second per call.

Accuracy: Each value checked had six significant digits. The table below shows which values were checked.

ν	λ
0	0; $e^{i\theta}$, $\theta = 10°, 20°, \ldots, 90°$; $5e^{i\theta}$, $\theta = 10°, 20°, \ldots, 90°$; $10e^{i\theta}$, $\theta = 10°, 30°, 60°, 90°$
1	0; $e^{i\theta}$, $\theta = 10°, 30°, 60°, 90°$; $5e^{i\theta}$, $\theta = 10°, 30°, 60°, 90°$
2	0
5	0; $e^{i\theta}$, $\theta = 10°, 30°, 60°, 90°$; $5e^{i\theta}$, $\theta = 10°, 30°, 60°, 90°$
10	0
100	0


START
ν ≥ 0
ν
ν < 0
10
IGO = 1
30
SAVE ν AND λ
SET ν = −ν
AND λ = −λ̄
IGO = 2
50
INITIALIZE
ν ≠ 0
ν
ν = 0
BSSLS AND BF4F
CALCULATE
HANKEL = H1(2)(ν) / (H1(2)(ν) + i H0(2)(ν))
60
λ ≠ 0
λ
λ = 0
ROCABES
CALCULATE
BESJLAM
= J o(λ) − i J1(λ)
70


70
70
ν ≠ 0
ν
ν = 0
90
λ ≠ 0
λ
λ = 0
λ ≠ 0
λ
λ = 0
80
100
CALCULATE CAPKL = BESJLAM * HANKEL + i ν J1(λ)/λ
CALCULATE CAPKL = HANKEL + iν/2
CALCULATE CAPKL = BESJLAM
CALCULATE CAPKL = 1
IGO = 1
IGO
IGO = 2
20
40
CALCULATE
CONLIFT = CAPKL = KL(ν, λ)
LIFT = CONLIFT = KL(ν, λ)
CALCULATE
LIFT = CAPKL = KL(-ν, -λ)
CONLIFT = LIFT = KL(-ν, -λ)
RESET ν AND λ
RETURN

```
      SUBROUTINE LIFTFN2(RNU,LAMDA,LIFT,CONLIFT)
      REAL LAMDAR,LAMDAI
      COMPLEX BESJLAM,BJ1LAM,BJ2LAM,CAPKL,CONLIFT,HANKEL,H1RNU,H2RNU
      COMPLEX LAMDA,LIFT,SAVELAM
      DIMENSION BJLAMR(250),BJLAMI(250),BYLAMR(50),BYLAMI(50)
      COMMON/SCRATCH/BES(1000)
      EQUIVALENCE (BES(1),BJLAMR(1)),(BES(251),BJLAMI(1))
      EQUIVALENCE (BES(501),BYLAMR(1)),(BES(551),BYLAMI(1))
      IF(RNU .GE. 0.)    10,30
10    IGO = 1
      GO TO 50
      CONLIFT = CAPKL
      LIFT = CONJG(CONLIFT)
      RETURN
      SAVERNU = RNU
      SAVELAM = LAMDA
      RNU = -RNU
      LAMDA = -CONJG(LAMDA)
      IGO = 2
      GO TO 50
      LIFT = CAPKL
      CONLIFT = CONJG(LIFT)
      RNU = SAVERNU
      LAMDA = SAVELAM
      RETURN
      ISIGN = -1
      NB = 1
      ABSLAM = CABS(LAMDA)
      IF(RNU .EQ. 0.)    GO TO 60
      CALL BSSLS(RNU,BES,NB,IERR)
      BJ1RNU = BES(1)
      BJ2RNU = BES(2)
      CALL BF4F(RNU,BES,NB,IERR,ISIGN)
      BY1RNU = -BES(1)
      BY2RNU = -BES(2)
      H1RNU = CMPLX(BJ1RNU,BY1RNU)
      H2RNU = CMPLX(BJ2RNU,BY2RNU)
      HANKEL = H2RNU / (H2RNU + (0.,1.)*H1RNU)
      IF(ABSLAM .EQ. 0.) GO TO 70
      LAMDAR = REAL(LAMDA)
      LAMDAI = AIMAG(LAMDA)
      CALL FOCABES(LAMDAR,LAMDAI,0.,NB,BJLAMR,BJLAMI,BYLAMR,BYLAMI)
      BJ1LAM = CMPLX(BJLAMR(1),BJLAMI(1))
      BJ2LAM = CMPLX(BJLAMR(2),BJLAMI(2))
      BESJLAM = BJ1LAM - (0.,1.)*BJ2LAM
70    IF(RNU .EQ. 0.)    GO TO 90
      IF(ABSLAM .EQ. 0.) GO TO 80
      CAPKL = BESJLAM*HANKEL + (0.,1.)*RNU*BJ2LAM/LAMDA
      GO TO (20,40) IGO
80    CAPKL = HANKEL + (0.,1.)*RNU/2.
      GO TO (20,40) IGO
90    IF(ABSLAM .EQ. 0.) GO TO 100
      CAPKL = BESJLAM
      GO TO (20,40) IGO
100   CAPKL = (1.,0.)
      GO TO (20,40) IGO
      END
```

3.2.11 Subroutine LIFTFN3

Purpose: LIFTFN3 computes the complex frequency response of aerodynamic lift of a thin, two-dimensional airfoil with parabolic mean camber line and angle of attack in a uniform, inviscid, incompressible subsonic mean flow. The current procedure is to use a linear combination of the responses to the transverse and longitudinal components of the incident velocity perturbation. The response to the transverse component corresponds to the Sears function (see ref. 32), while the response to the longitudinal component is the sum of two terms, one proportional to the angle of attack and the other proportional to the ratio of maximum camber to half-chord (see ref. 35):

$$L(\nu) = S(\nu) - \mathrm{COT}\beta\left[f\, F_f(\nu) + \alpha F_\alpha(\nu)\right]$$

with ν the reduced frequency, β the angle made by the velocity perturbation and the mean flow through the cascade, S the Sears function, f and α the ratio of maximum camber to the half-chord and the angle of attack, respectively, and F_f, F_α the camber and angle of attack responses to the longitudinal component of the velocity perturbation.

$$F_f(\nu) = \frac{H_o^{(2)}(\nu) + i\, H_1^{(2)}(\nu)}{-H_o^{(2)}(\nu) + i\, H_1^{(2)}(\nu)}\left[J_o(\nu) - \frac{J_1(\nu)}{\nu} - i\, J_1(\nu)\right]$$

$$- \left[J_o(\nu) - \frac{J_1(\nu)}{\nu} + i\, J_1(\nu)\right] + \frac{4}{\nu} J_1(\nu)$$

$$F_\alpha(\nu) = J_o(\nu) + i\, J_1(\nu)\ , \quad S(\nu) = \frac{-i}{\frac{2\nu}{\pi}\left(H_o^{(2)}(\nu) + i\, H_1^{(2)}(\nu)\right)}$$

It should be noted that these response functions, including the Sears function, are spectral functions under the convention:

$$\bar{g}(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{g}(\omega)\, e^{i\omega t}\, d\omega.$$

The subroutine computes:

$$L(\nu) = b_1 S(\nu) + b_2\, F_\alpha(\nu) + b_3\, F_f(\nu)$$

where $b_1 = 1.$, $b_2 = -\alpha \cot\beta$, and $b_3 = -f \cot\beta$ are calculated outside the subroutine.

Method: The procedure is as follows:

1) Input ν, b_1, b_2, b_3 through calling sequence.

2) Set $S(\nu) = 1$, $F_\alpha(\nu) = 1$, $F_f(\nu) = 2$, and go to step 8 if $\nu = 0$.

3) Compute $J_o(|\nu|)$, $J_1(|\nu|)$, $Y_o(|\nu|)$, $Y_1(|\nu|)$.

4) Compute $H_o^{(2)}(|\nu|)$, $H_1^{(2)}(|\nu|)$.

5) Compute $S(\nu) = \begin{cases} S(|\nu|) & \text{if} \quad \nu > 0 \\ \overline{S(|\nu|)} & \text{if} \quad \nu < 0 \end{cases}$

6) Compute $F_\alpha(\nu) = \begin{cases} F_\alpha(|\nu|) & \text{if} \quad \nu > 0 \\ \overline{F_\alpha(|\nu|)} & \text{if} \quad \nu < 0 \end{cases}$

7) Compute $F_f(\nu) = \begin{cases} F_f(|\nu|) & \text{if} \quad \nu > 0 \\ \overline{F_f(|\nu|)} & \text{if} \quad \nu < 0 \end{cases}$

8) Compute $L(\nu) = b_1 S(\nu) + b_2 F_\alpha(\nu) + b_3 F_f(\nu)$.

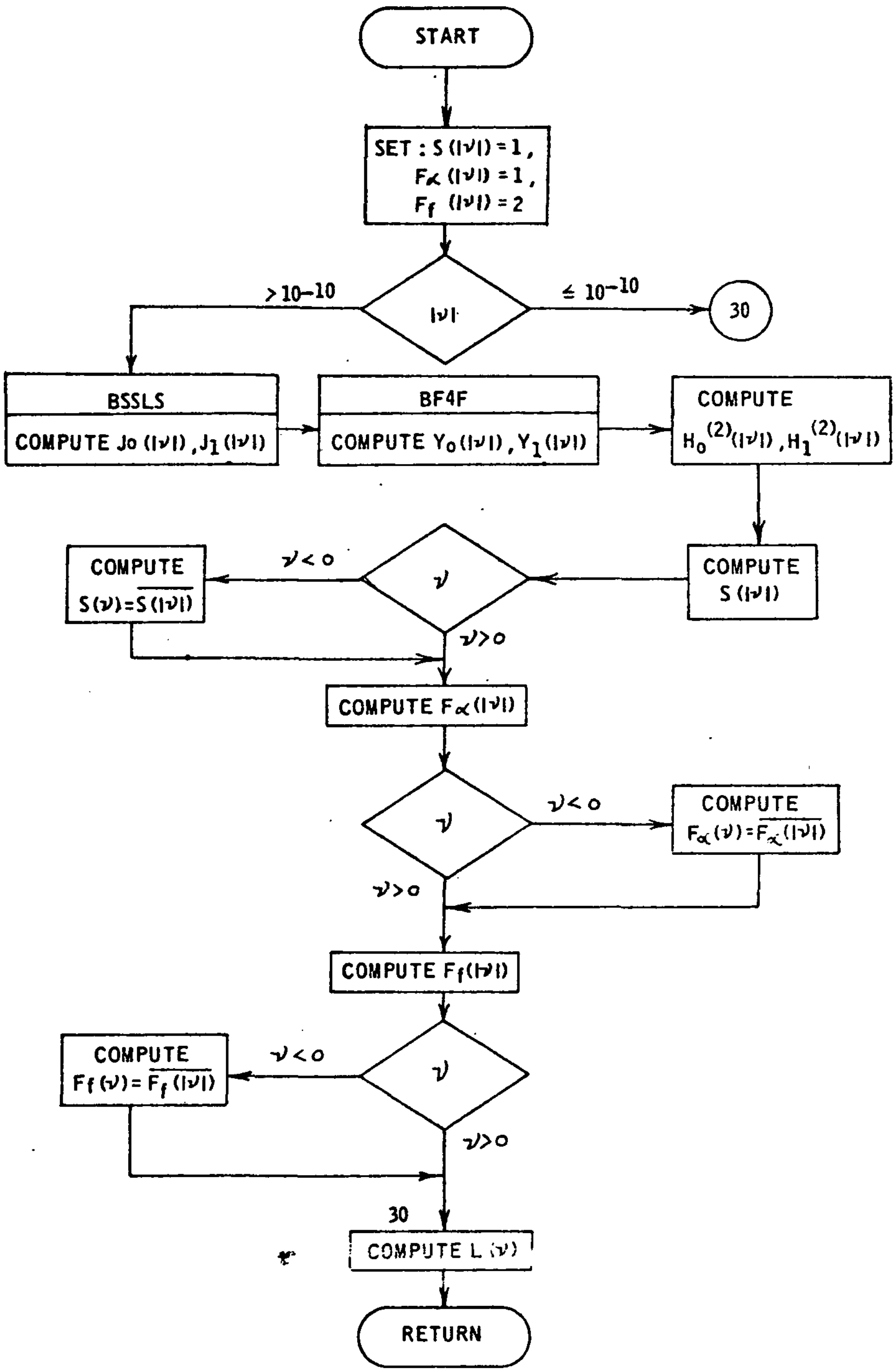
9) Return.

Usage: CALLING SEQUENCE

```
COMPLEX CAPLT
COMMON/SCRATCH/BES(1000)
.
.
.
CALL LIFTFN3(RNU,B1,B2,B3,CAPLT)
```

Restrictions: None

Accuracy: The accuracy is of the algorithmic type and, in particular, is dominated by subroutines BSSLS and BF4F.


START
SET : S (|ν|) = 1 , Fα (|ν|) = 1 , Ff (|ν|) = 2
|ν|
> 10−10
≤ 10−10
30
BSSLS
COMPUTE Jo (|ν|) , J1 (|ν|)
BF4F
COMPUTE Y0 (|ν|) , Y1 (|ν|)
COMPUTE H0(2)(|ν|) , H1(2)(|ν|)
COMPUTE S (|ν|)
ν
ν < o
ν > o
COMPUTE S(ν) = S(|ν|)
COMPUTE Fα (|ν|)
ν
ν < o
ν > o
COMPUTE Fα (ν) = Fα (|ν|)
COMPUTE Ff (|ν|)
ν
ν < o
ν > o
COMPUTE Ff (ν) = Ff (|ν|)
30
COMPUTE L (ν)
RETURN

```
      SUBROUTINE LIFTFN3(RNU,B1,B2,B3,CAPLT)
C
C
C REFERENCE     H. NAUMANN AND H. YEN, LIFT AND PRESSURE FLUCTUATIONS OF
C               A CAMBERED AIRFOIL UNDER PERIODIC GUSTS AND APPLICATIONS
C               IN TURBOMACHINERY, JOURNAL OF ENGINEERING FOR POWER,
C               TRANSACTIONS OF THE ASME, JANUARY 1973.
C
C
      COMPLEX CAPLT,CTEMP1,FALPHNU,FFNU,H1RNU,H2RNU,SNU
      COMMON/SCRATCH/BES(1000)
      DATA PI,ISIGN/3.14159265358979,-1/
C               FOR RNU IN ABSOLUTE VALUE LESS THAN 1.E-10, SET THE
C               FUNCTIONS TO 1 AND RETURN
C
      SNU = (1.,0.)
      FALPHNU =(1.,0.)
      FFNU = (2.,0.)
      ABSNU = ABS(RNU)
      IF( ABSNU .LE. 1.E-10 ) GO TO 30
C
C               COMPUTE FUNCTIONS FOR ABSNU AND THEN RNU
C
C
C               COMPUTE REQUIRED BESSEL FUNCTIONS
C
   20 CALL BSSLS(ABSNU,BES,1,IERR)
      BJ1 = BES(1)
      BJ2 = BES(2)
      CALL BF4F(ABSNU,BES,1,IERR,ISIGN)
      BY1 = BES(1)
      BY2 = BES(2)
C
C               COMPUTE HANKEL FUNCTIONS AN SNU
C
      H1RNU = CMPLX(BJ1,-BY1)
      H2RNU = CMPLX(BJ2,-BY2)
C
      CTEMP1 = CMPLX(BJ1-BY2,-BJ2-BY1)
      SNU = 1./(.5*PI*ABSNU*CTEMP1)
      IF( RNU.LT.0.) SNU = CONJG( SNU )
C
C               COMPUTE FALPHNU
C
      FALPHNU = CMPLX(BJ1,BJ2)
      IF( RNU.LT.0. ) FALPHNU = CONJG( FALPHNU )
C
C               COMPUTE T, F, AND FALPHNU
C
      CTEMP1 =  (H1RNU + (0.,1.)*H2RNU) / (-H1RNU + (0.,1.)*H2RNU)
      TEMP1  =  BJ1 - BJ2/ABSNU
      FFNU = CTEMP1*CMPLX(TEMP1,-BJ2) - CMPLX(TEMP1,BJ2) + 4.*BJ2/ABSNU
C
      IF( RNU.LT. 0. ) FFNU = CONJG(FFNU)
C
C
   30 CAPLT = B1*SNU + B2*FALPHNU + B3*FFNU
C
      RETURN
      END
```

3.2.12 Subroutine LIFTFN4

Purpose: This subroutine computes the lift response function of a flat-plate, thin, two-dimensional airfoil in incompressible flow to an oblique, frozen-convected gust (see fig. 12).

The response function $T(\nu,\beta)$ depends on ν and β, where:

ν = reduced frequency

β = gust yaw angle

and

$$T(\nu,\beta) = \frac{1}{\frac{\pi}{2}\nu \quad F(\nu,\beta)} \cdot \frac{I_o(\nu_2) + I_1(\nu_2)}{J_o(\nu_1 - i\ \nu_2) + i\ J_1(\nu_1 - i\ \nu_2)}$$

$$F(\nu,\beta) = \frac{\pi}{2}\nu + e^{-i\nu_1}\left\{\nu_2\ K_1(\nu_2) - iK_o(\nu_2)\right\} - \nu\ \mathrm{SEC}\beta\ G(\nu_2, \mathrm{TAN}\beta)$$

$$G(x,\alpha) = \int_x^\infty e^{-i\alpha z}\ K_o(z)\ dz$$

where $\nu_1 = \nu \sin\beta$, $\nu_2 = \nu \cos\beta$, I_0, I_1 and K_0, K_1 are the modified Bessel functions, and J_0 and J_1 are the Bessel functions.

An approximation formula (ref. 36) is:

$$T(\nu,\beta) \sim \frac{e^{-i\nu\left[\mathrm{SIN}\,\beta \quad \frac{\pi\beta\ (1+\frac{1}{2}\ \mathrm{COS}\beta)}{1\ +\ 2\pi\nu(1\ +\ \frac{1}{2}\ \mathrm{COS}\beta)}\right]}}{\left[1 + \pi\nu(1 - \mathrm{SIN}^2\beta + \pi\ \nu\ \mathrm{COS}\beta)\right]^{\frac{1}{2}}}$$

The subroutine evaluates either the response function or the approximation in the return variable LIFT4.

Method: The procedure is as follows:

1) Initialize the error return to zero.

2) Set $\nu_1 = |\nu| \sin\beta$ and $\nu_2 = |\nu| \cos\beta$.

3) When $|\nu| \leq EPS = 10^{-10}$, $T(|\nu|,\beta) = (1.,0.)$ and return.

4) When the approximation is to be evaluated, proceed to step 16; otherwise continue with step 5.

5) When $|\beta - \pi/2| \leq EPS = 10^{-10}$, proceed to step 13.

6) When $|\beta - \pi/2| > EPS = 10^{-10}$, compute $G(\nu_2, \tan\beta)$ using a Gaussian formula according to steps 7 to 9.

7) Divide the range of integration into subintervals of width $\Delta = \pi/\max(1,\tan\beta)$ starting at ν_2.

8) Compute the integral on each subinterval by a 12-point Gaussian formula, an 8-point Gaussian formula when $1 \leq \nu_2 \leq 10$, and a 4-point Gaussian formula when $\nu_2 > 10$, with the 12-point being used whenever interval bound $/\Delta < 4$ and summing the integrals.

9) When the integral on a subinterval in absolute value is less than $EPS = 10^{-10}$ times the sum in absolute value, accept the value and proceed to step 10.

10) When an interval lower bound is more than 1000., set an error return and proceed to step 10; otherwise go to step 7.

11) Compute the modified Bessel functions I_0, I_1, K_0, and K_1 at ν_2.

12) Evaluate $F(|\nu|, \beta)$ and proceed to step 13.

13) Compute $F(|\nu|, \beta)$ directly as:

$$F(|\nu|,\tfrac{\pi}{2}) = \mathrm{COS}\nu + \nu\ \mathrm{Si}(\nu) + i\left[\nu\ \mathrm{Ci}(\nu) - \mathrm{SIN}\nu\right]$$

where Si and Ci are the sine and cosine integrals.

14) Compute the complex Bessel functions:

$$J_o(\nu_1 - i\nu_2) \text{ and } J_1(\nu_1 - i\nu_2).$$

15) Compute $T(\nu, \beta)$ using the appropriate formula, depending on whether ν is positive or negative, and return.

16) Evaluate the approximation formula and return.

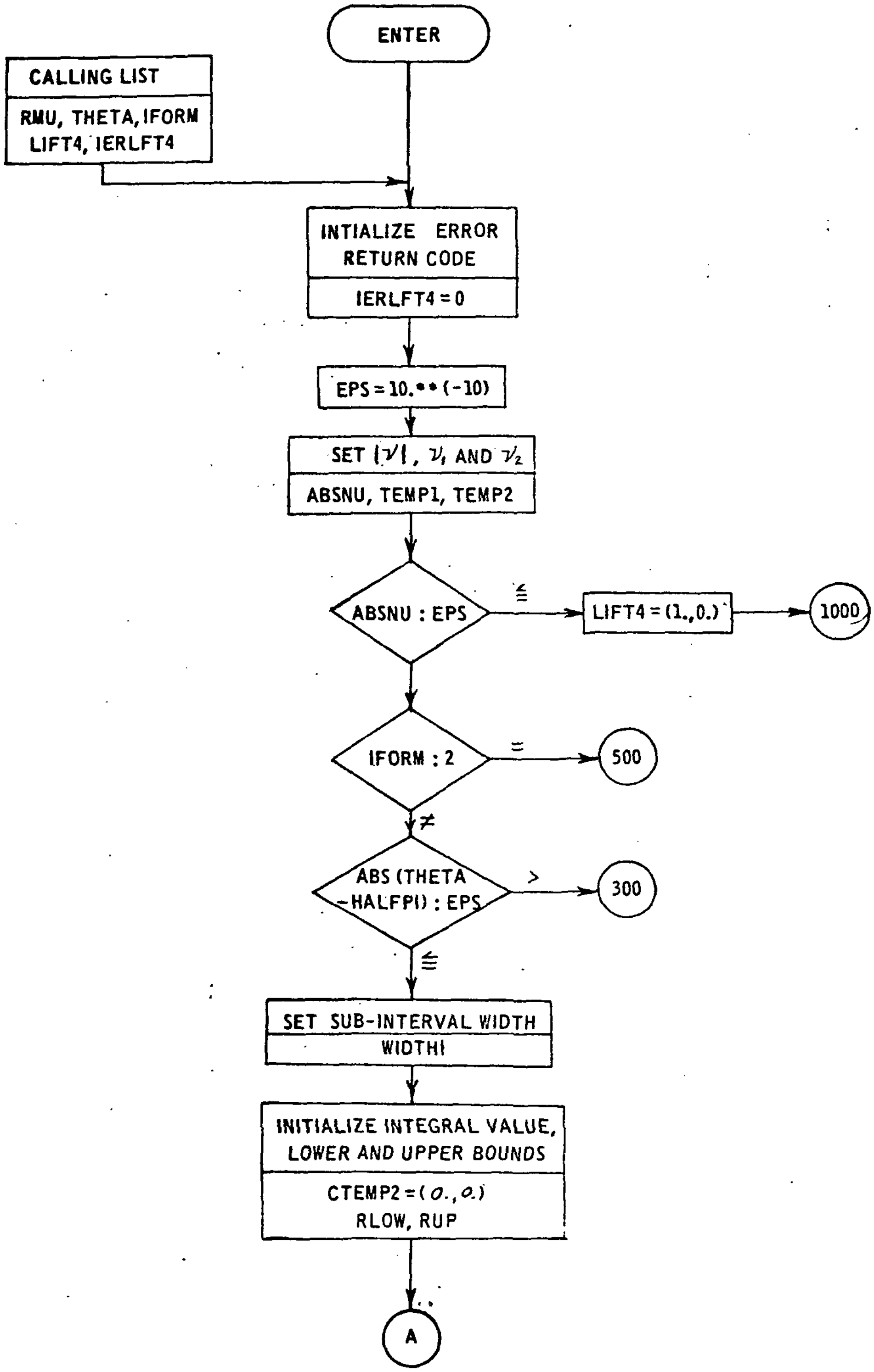
Usage: CALLING SEQUENCE

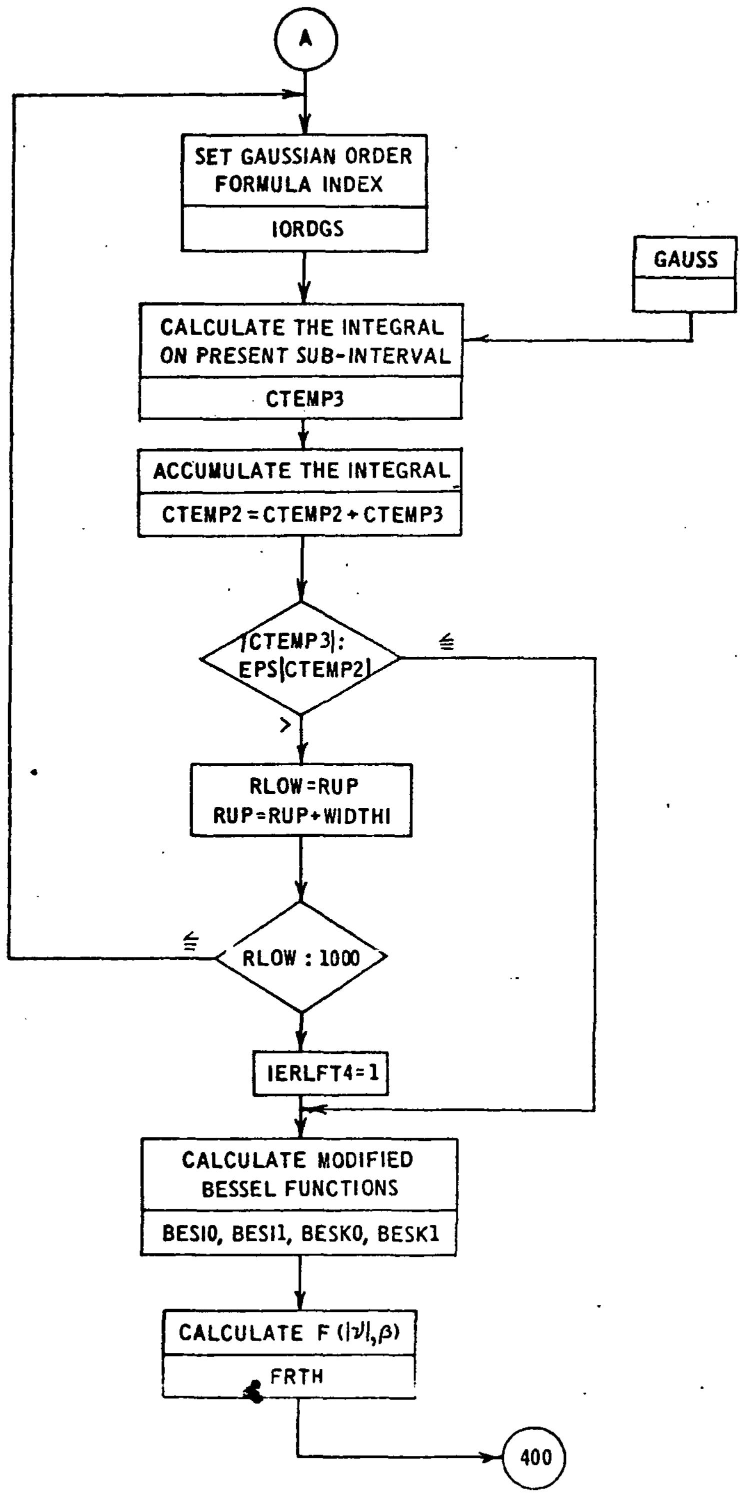
```
COMPLEX LIFT4
•
•
•
CALL LIFTFN4(RNU,THETA,IFORM,LIFT4,IERLFT4)
```

Restrictions: RNU ≥ 0

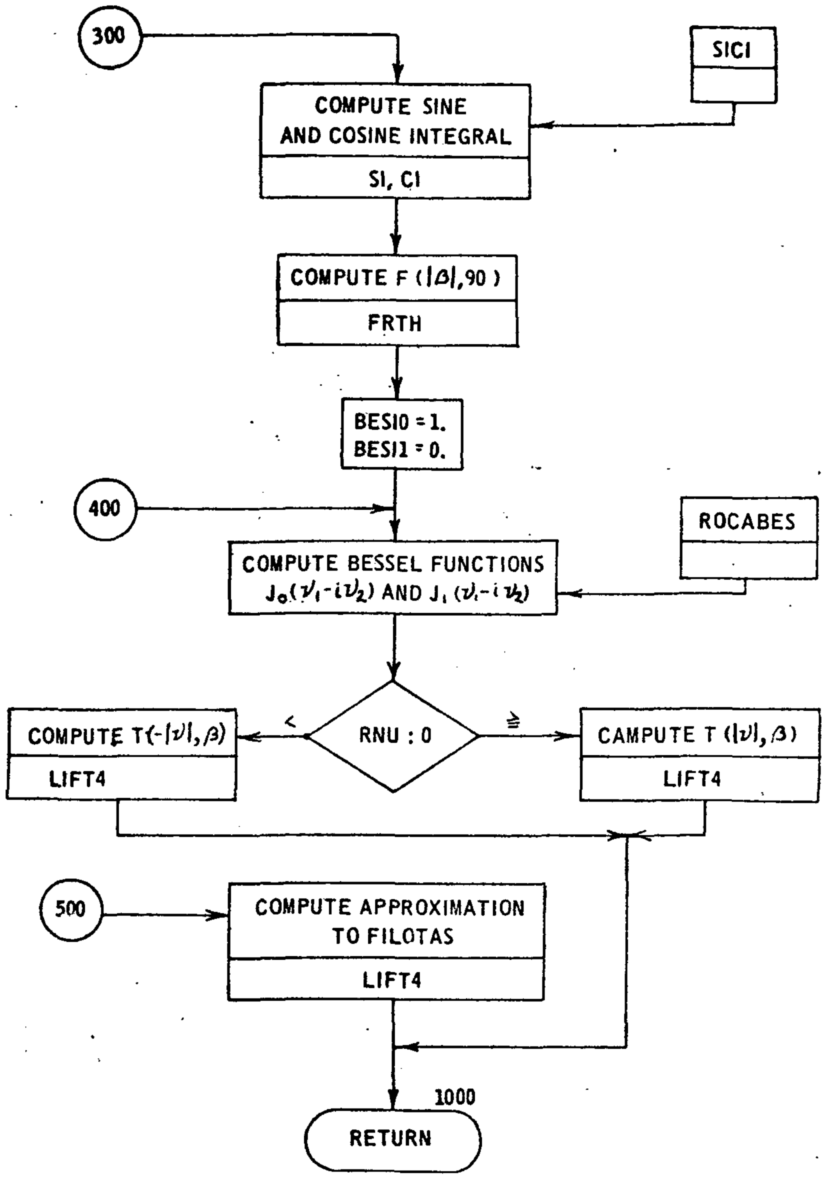
Timing: The timing is dependent upon the input variables ν and β, especially in the computation of $G(x,\alpha)$.

Accuracy: The accuracy is of the algorithmic type and, in particular, is dominated by the computation of $G(x,\alpha)$ and the Bessel functions of complex argument.


ENTER
CALLING LIST
RMU, THETA, IFORM
LIFT4, IERLFT4
INTIALIZE ERROR
RETURN CODE
IERLFT4 = 0
EPS = 10.** (-10)
SET |ν|, ν₁ AND ν₂
ABSNU, TEMP1, TEMP2
ABSNU : EPS
≦
LIFT4 = (1.,0.)
1000
IFORM : 2
=
500
≠
ABS (THETA
-HALFPI) : EPS
>
300
≦
SET SUB-INTERVAL WIDTH
WIDTHI
INITIALIZE INTEGRAL VALUE,
LOWER AND UPPER BOUNDS
CTEMP2 = (0.,0.)
RLOW, RUP
A



A
SET GAUSSIAN ORDER FORMULA INDEX
IORDGS
GAUSS
CALCULATE THE INTEGRAL ON PRESENT SUB-INTERVAL
CTEMP3
ACCUMULATE THE INTEGRAL
CTEMP2 = CTEMP2 + CTEMP3
|CTEMP3| : EPS|CTEMP2|
≦
>
RLOW=RUP
RUP=RUP+WIDTHI
RLOW : 1000
≦
IERLFT4=1
CALCULATE MODIFIED BESSEL FUNCTIONS
BESI0, BESI1, BESK0, BESK1
CALCULATE F (|ν|,β)
FRTH
400


300
SICI
COMPUTE SINE AND COSINE INTEGRAL
SI, CI
COMPUTE F (|β|,90)
FRTH
BESI0 = 1.
BESI1 = 0.
400
ROCABES
COMPUTE BESSEL FUNCTIONS
J₀(ν₁-iν₂) AND J₁ (ν₁-iν₂)
COMPUTE T(-|ν|, β)
LIFT4
RNU : 0
CAMPUTE T (|ν|, β)
LIFT4
500
COMPUTE APPROXIMATION TO FILOTAS
LIFT4
1000
RETURN

```
      SUBROUTINE LIFTFN4(RNU,THETA,IFORM,LIFT4,IERLFT4)


      COMMON/SCRATCH/BES(1000)
      COMMON/ALPHA/ALPHA
      COMPLEX CTEMP1,CTEMP2,CTEMP3,FRTH,GRTHFCN,LIFT4
      EXTERNAL GRTHFCN
      DATA PI,HALFPI/3.14159265358975,1.57079632679490/

                SET RESULT FOR SMALL RNU

      IERLFT4 = 0
      EPS = 1.E-10
      ABSNU = ABS(RNU)
      TEMP3 = SIN(THETA)
      TEMP4 = COS(THETA)
      TEMP1 = ABSNU*TEMP3
      TEMP2 = ABSNU*TEMP4
      IF(ABSNU-EPS) 10,10,20
      LIFT4 = (1.,0.)
      GO TO 1000
      IF(IFORM.EQ.2) GO TO 500

                COMPUTE EXACT FILOTAS NUMERICALLY


                COMPUTE F(ABSNU,THETA)

      IF(ABS(THETA-HALFPI).LE.EPS) GO TO 300

                COMPUTE G(ABSNU,THETA)  USING GAUSS FORMULA BETWEEN HALF
                                        CYCLES OF THE TRIGNOMETRIC FACTOR
                                        UNTIL A CONTRIBUTION IS SMALL

      ALPHA = TEMP3/TEMP4
      WIDTHI = PI/AMAX1(1.,ALPHA)
      CTEMP2 = (0.,0.)
      RLOW = TEMP2
      IF( RLOW .GE. 1000.)  GO TO 130
      N1 = TEMP2/WIDTHI + 1.
      RUP = N1*WIDTHI
      IF(RUP .LE. RLOW)  RUP = RUP + WIDTHI
  110 IORDGS = 3
      IF(TEMP2 .GT. 1.)         IORDGS = 2
      IF(TEMP2 .GT. 10.)        IORDGS = 1
      IF(RUP/WIDTHI .LT. 4.)  IORDGS = 3
      CALL GAUSS(RLOW,RUP,CTEMP3,GRTHFCN,IORDGS)
      CTEMP2 = CTEMP2 + CTEMP3
      IF( CABS(CTEMP3) .LE. EPS*CABS(CTEMP2) )     GO TO 130
      RLOW = RUP
      RUP = RUP + WIDTHI
```

```
      IF( RLOW - 1000. )  110,110,120
  110 IERLFT4 = 1

  120 CALL BESIK(TEMP2,6,BESI0,BESI1,BESK0,BESK1,IERBES)
      CTEMP1 = CMPLX( COS(TEMP1), -SIN(TEMP1) )
      FRTH = HALFPI*ABSNU*CTEMP1*CMPLX( TEMP2*BESK1, -TEMP1*BESK0)
     1        - ABSNU*CTEMP2/TEMP4
      GO TO 400

                COMPUTE F(ABSNU,HALFPI)

  300 CONTINUE

                COMPUTE T(RNU,THETA)  USING F(ABSNU,THETA)

      CALL SICI(SI,CI,ABSNU)
      SI=SI+HALFPI
      FRTH = CMPLX( COS(ABSNU) + ABSNU*SI, ABSNU*CI - SIN(ABSNU) )
      BESI0=1.
      BESI1=0.

      CALL ROCABES(TEMP1,-TEMP2,0.,1,BES(1),BES(450),BES(900),BES(950))

      IF(2NU)  410,1000,420
      LIFT4 = (1./(-HALFPI*ABSNU + CONJG(FRTH))) *
     1 ((BESI0-BESF1)/(CMPLX(BES(1),BES(450))+CMPLX(BES(451),-BES(2))))
      GO TO 1000
      LIFT4 = (1./(HALFPI*ABSNU + FRTH)) *
     1 ((BESI0+BESI1)/(CMPLX(BES(1),BES(450))-CMPLX(BES(451),-BES(2))))
      GO TO 1000

                COMPUTE APPROXIMATION TO T(R,THETA) (EQUATION 32)

  500 TEMP1= PI*RNU*( 1. + .5*TEMP4 )
      TEMP1=RNU*TEMP3 - THETA*TEMP1/( 1. + 2.*TEMP1 )
      TEMP2 = 1. + PI*RNU*( 1. + TEMP3**2 + PI*RNU*TEMP4 )
      TEMP2 = 1. / SQRT(TEMP2)
      LIFT4 = CMPLX( COS(TEMP1), -SIN(TEMP1) )*TEMP2

 1000 CONTINUE

      RETJRN
      END
```

3.2.13 Function DISINT

Purpose: This routine evaluates the function:

$$\left[(A\rho\ \cos\phi\ -1)^2\ -\ (A^2-1)(\rho^2-1)\right]^{\frac{1}{2}}\ e^{-i\ell\phi}$$

which is called by GAUSS in the computation of the Fourier coefficients in the cone distortion model.

Method: The procedure is as follows:

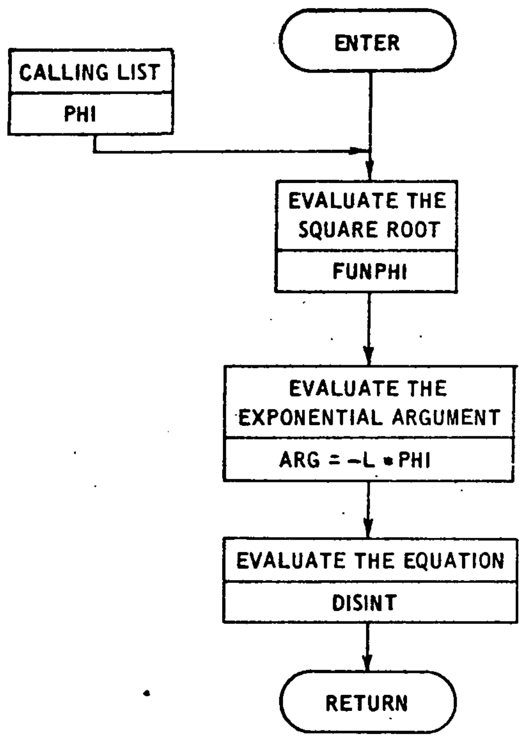
1) Compute the square root.

2) Multiply the square root with the trigonometric (exponential), evaluating the equation.

Usage: CALLING SEQUENCE

```
COMPLEX DISINT,VDISINT
COMMON/CDISINT/CAPADIS,RHOINC
COMMON/CFACT/M,N,RMUMN,CAPNMN,ETA,SIGN,L,CAPKMN
.
.
.
VDISINT=DISINT(PHI)
```

Accuracy: The accuracy is of the algorithmic type and is dominated by the system routines SQRT, SIN, and COS.


ENTER
CALLING LIST
PHI
EVALUATE THE SQUARE ROOT
FUNPHI
EVALUATE THE EXPONENTIAL ARGUMENT
ARG = −L * PHI
EVALUATE THE EQUATION
DISINT
RETURN

```
      COMPLEX FUNCTION DISINT(PHI)
C
C PURPOSE        EVALUATE THE INTEGRAND OF THE INTEGRAL IN THE
C                DISTORTION COEFFICIENT
C
      COMMON/CDISINT/CAPADIS,RHOINC
      COMMON/CFACT/  M,N,RMUMN,CAPNMN,ETA,SIGN,L,CAPKMN
C
      FUNPHI = CAPADIS*RHOINC*COS(PHI)-1.
      FUNPHI = FUNPHI**2 - (CAPADIS**2-1.)*(RHOINC**2-1.)
      FUNPHI = SQRT( FUNPHI )
      ARG = -L*PHI
      DISINT = FUNPHI*CMPLX( COS(ARG),SIN(ARG) )
C
      RETURN
      END
```

3.2.14 Function FUNIN4

Purpose: Complex Function FUNIN4 computes:

$$e^{-\frac{(\omega T_j)^2}{2}} \cos(\omega T)$$

An integral of this function is needed in function FACTIN4. Subroutine GAUSS, which requires a complex function subprogram to evaluate the function, is used to compute this integral.

Method: The procedure is to calculate the function and return.

Usage: CALLING SEQUENCE

```
COMPLEX Y,FUNIN4
COMMON/CFUNIN4/CTJ,TAU
•
•
•
Y = FUNIN4 (OMEGA)
```

Accuracy: The accuracy is of the computer type.

```
COMPLEX FUNCTION FUNIN4(OMEGA)
COMMON/CFUNIN4/CTJ,TAU
TEMP1 = -.5*( (OMEGA*CTJ)**2 )
TEMP2 = EXP(TEMP1) * COS(OMEGA*TAU)
FUNIN4 = CMPLX(TEMP2,0.)
RETURN
END
```

3.2.15 Subroutine NONCPT

Purpose: This subroutine computes the acoustic response function for a noncompact airfoil from an integration of the pressure difference function of reference 35.

Subroutine NONCPT computes:

$$L'_n(\nu) = b_1 \cdot S(\nu)\, J\left(\kappa^{\pm}_{mn\sigma}\right) + b_2\, J\left(\nu + \kappa^{\pm}_{mn\sigma}\right)$$

$$+ b_3 \left\{ J\left(\kappa^{\pm}_{mn\sigma}\right) F(\nu) + \frac{2J_1\left(\nu + \kappa^{\pm}_{mn\sigma}\right)}{\nu + \kappa^{\pm}_{mn\sigma}} \right.$$

$$\left. - \frac{2}{\nu} \sum_{j=1}^{JMAX} (-1)^j\, J_j(\nu) \left[J_{j+1}\left(\kappa^{\pm}_{mn\sigma}\right) + J_{j-1}\left(\kappa^{\pm}_{mn\sigma}\right) \right] \right\}$$

where $J(X) = J_o(X) + i\, J_1(X)$

$$S(X) = \frac{-1}{\frac{\pi X}{2}\left(-H_o^{(2)}(X) + i\, H_1^{(2)}(X)\right)}$$

$$T(X) = \frac{H_o^{(2)}(X) + iH_1^{(2)}(X)}{-H_o^{(2)}(X) + iH_1^{(2)}(X)}$$

$$F(X) = T(X) \left[\overline{J\ (X)} - \frac{J_1(X)}{X} \right] - \left[J(X) - \frac{J_1(X)}{X} \right]$$

$$\kappa^{\pm}_{mn\sigma} = \frac{C_2}{2} \left[K^{\pm}_{mn}\ e_\phi - \frac{m}{\rho}\ e_z \right], \text{ and}$$

$b_1, b_2, b_3, C_2, K^{\pm}_{mn}, e_\phi, m, \rho, \nu, e_z$ are input.

Method:

1) Set $F(|\nu|) = 0$, $S(|\nu|) = 1$ and if $\nu = 0$, go to step 5.

2) Compute $H_o^{(2)}(|\nu|)$, $H_1^{(2)}(|\nu|)$ and $S(|\nu|)$.

3) Compute $S(\nu) = \begin{cases} S(|\nu|) & \text{if } \nu > 0 \\ \overline{S(|\nu|)} & \text{if } \nu < 0 \end{cases}$

4) Compute $F(|\nu|)$ and

$$F(\nu) = \begin{cases} F(|\nu|) & \text{if } \nu > 0 \\ \overline{F(|\nu|)} & \text{if } \nu < 0 \end{cases}$$

5) Compute $\kappa^{\pm}_{mn\sigma}$.

6) Compute $J_o\left(\nu + \kappa^{\pm}_{mn\sigma}\right)$, $J_1\left(\nu + \kappa^{\pm}_{mn\sigma}\right)$,

and $\quad \text{TEMP3} = \dfrac{2J_1\left(\nu + \kappa^{\pm}_{mn\sigma}\right)}{\nu + \kappa^{\pm}_{mn\sigma}}$ where

if $\nu + \kappa^{\pm}_{mn\sigma} = 0$, then $J_o(0) = 1$, $J_1(0) = 0$, and TEMP3 = 1.

7) If $\nu \cdot \kappa^{\pm}_{mn\sigma} = 0$, go to step 12.

8) Set JMAX $= \max\left(|\kappa^{\pm}_{mn\sigma}|, |\nu|\right) + 1$.

9) Compute $J_j(|\nu|)$, $j = 1, 2, \text{---}, \text{JMAX}$ and

$J_i(|\kappa^{\pm}_{mn\sigma}|)$, $i = 0, 1, \text{---}, \text{JMAX}+1$ using

subroutine $\begin{cases} \text{BSSLS} & \text{if JMAX} + 1 \le 100 \\ \text{BESNX} & \text{if } 100 < \text{JMAX} + 1 \end{cases}$

10) Compute $J_j(X) = (-1)^j J_j(|X|)$ where $X = \nu, \kappa^{\pm}_{mn\sigma}$.

11) Compute $\text{SUM} = \frac{2}{\nu} \sum_{j=1}^{\text{JMAX}} (-1)^j J_j(\nu) \left[J_{j+1}\left(\kappa^{\pm}_{mn\sigma}\right) + J_{j-1}\left(\kappa^{\pm}_{mn\sigma}\right) \right]$

and go to step 14.

12) Compute necessary Bessel functions for steps 13 and 14, i.e., compute $J_0\left(\kappa^{\pm}_{mn\sigma}\right)$, $J_1\left(\kappa^{\pm}_{mn\sigma}\right)$ for step 14,

compute $J_2\left(\kappa^{\pm}_{mn\sigma}\right)$ if $\kappa^{\pm}_{mn\sigma} \neq 0$ and

compute $J_1(\nu)$ if $\nu \neq 0$.

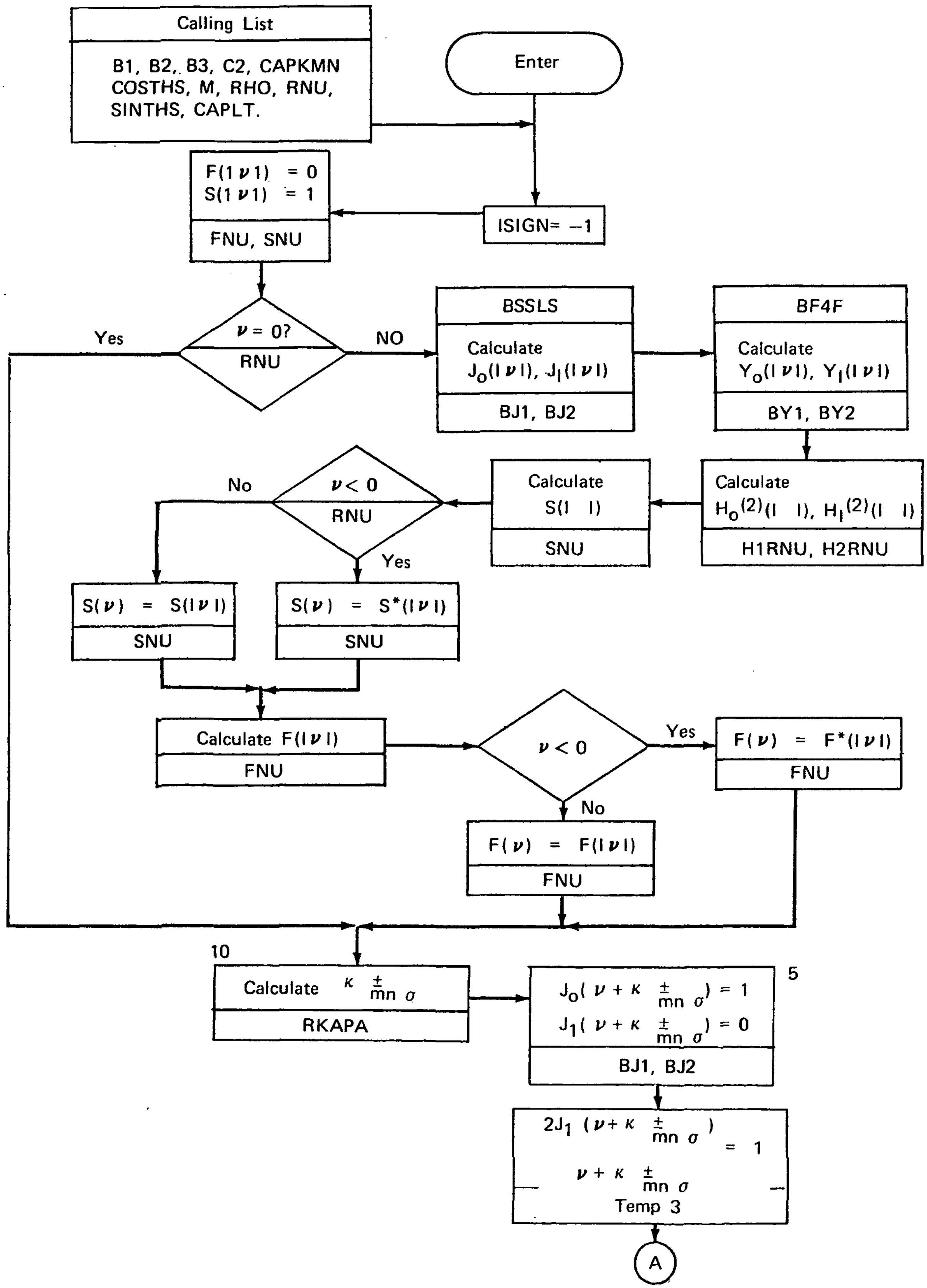
13) Compute $\text{SUM} = \begin{cases} -1 & \text{if } \nu = 0 \text{ and } \kappa^{\pm}_{mn\sigma} = 0 \\ -\left[J_2\left(\kappa^{\pm}_{mn\sigma}\right) + J_0\left(\kappa^{\pm}_{mn\sigma}\right)\right] & \text{if } \nu=0 \text{ and } \kappa^{\pm}_{mn\sigma} \neq 0 \\ -\left(\frac{2}{\nu}\right) J_1(\nu) & \text{if } \nu \neq 0 \text{ and } \kappa^{\pm}_{mn\sigma} = 0 \end{cases}$

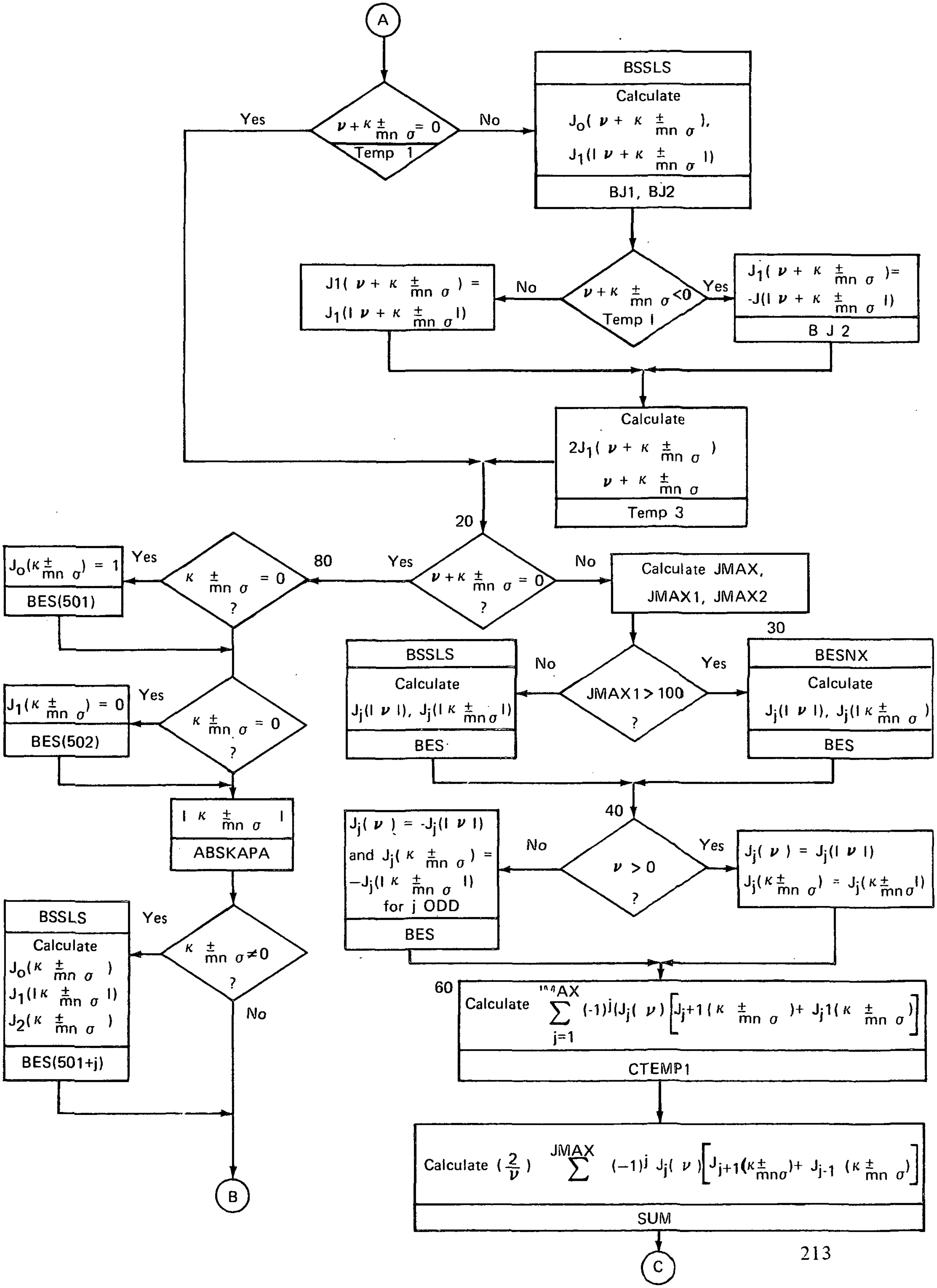
14) Compute $L'_{\Omega}(\nu)$ and return.

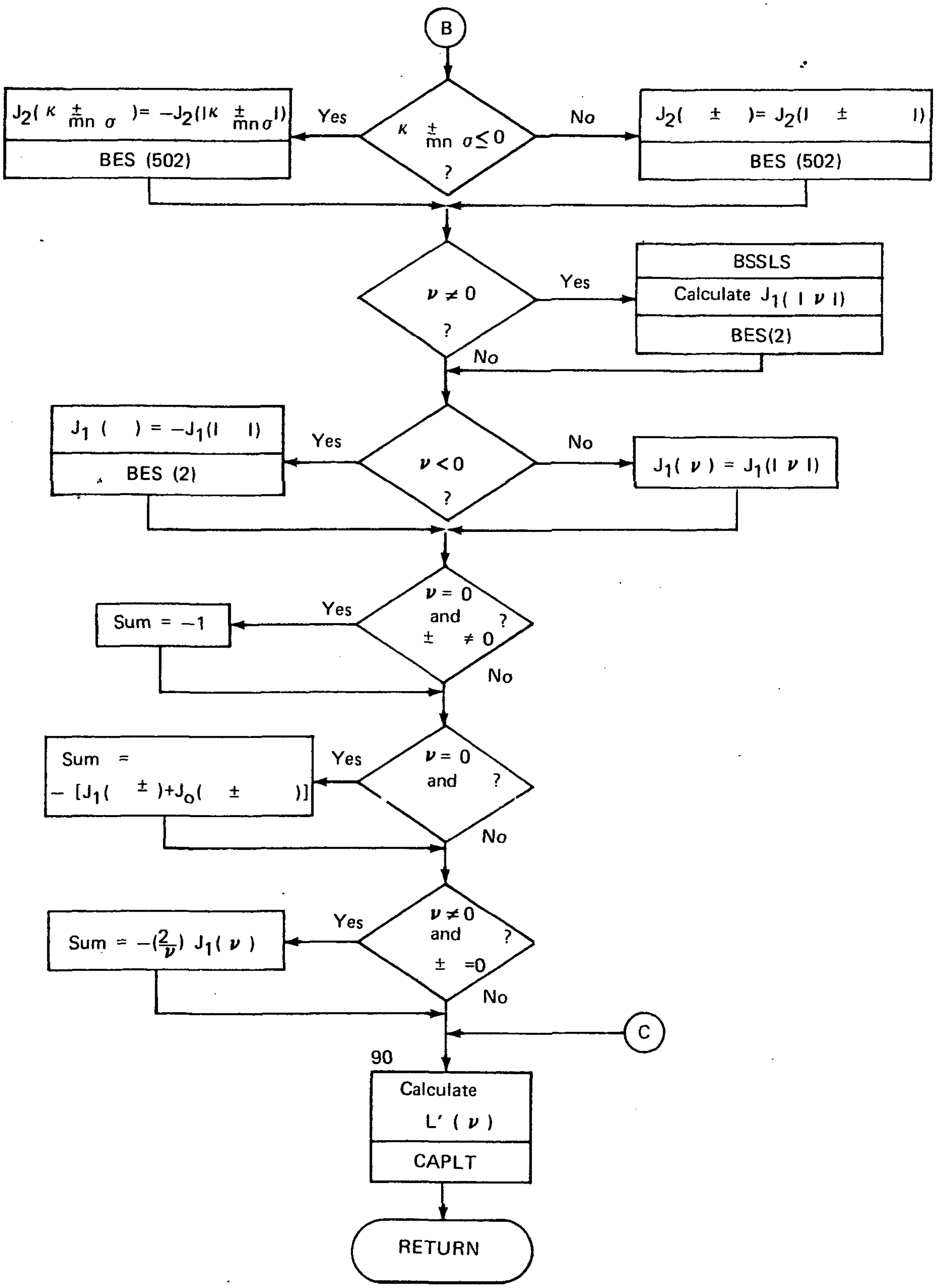
Usage: CALLING SEQUENCE

```
COMPLEX CAPLN
COMMON/SCRATCH/BES(1000)
 .
 .
 .
CALL NONCPT(B1,B2,B3,C2,CAPKMN,COSTHS,M,RHO,RNU,SINTHS,
*    CAPLN)
```

Accuracy: The accuracy is of the computer type.


Calling List
B1, B2, B3, C2, CAPKMN
COSTHS, M, RHO, RNU,
SINTHS, CAPLT.
Enter
ISIGN= −1
F(1 ν 1) = 0
S(1 ν 1) = 1
FNU, SNU
ν = 0?
RNU
Yes
NO
BSSLS
Calculate
J0(I ν I), J1(I ν I)
BJ1, BJ2
BF4F
Calculate
Y0(I ν I), Y1(I ν I)
BY1, BY2
Calculate
H0(2)(I I), H1(2)(I I)
H1RNU, H2RNU
Calculate
S(I I)
SNU
ν < 0
RNU
No
Yes
S(ν) = S(I ν I)
SNU
S(ν) = S*(I ν I)
SNU
Calculate F(I ν I)
FNU
ν < 0
Yes
No
F(ν) = F*(I ν I)
FNU
F(ν) = F(I ν I)
FNU
10
Calculate κ ± mn σ
RKAPA
5
J0(ν + κ ± mn σ) = 1
J1(ν + κ ± mn σ) = 0
BJ1, BJ2
2J1 (ν + κ ± mn σ) / ν + κ ± mn σ = 1
Temp 3
A


A
ν + κ ± mn σ = 0
Temp 1
Yes
No
BSSLS
Calculate
J0(ν + κ ± mn σ),
J1(| ν + κ ± mn σ |)
BJ1, BJ2
ν + κ ± mn σ < 0
Temp I
J1(ν + κ ± mn σ) = J1(| ν + κ ± mn σ |)
J1(ν + κ ± mn σ) = -J(| ν + κ ± mn σ |)
B J 2
Calculate
2J1(ν + κ ± mn σ) / ν + κ ± mn σ
Temp 3
20
ν + κ ± mn σ = 0 ?
80
κ ± mn σ = 0 ?
J0(κ ± mn σ) = 1
BES(501)
Calculate JMAX, JMAX1, JMAX2
30
JMAX1 > 100 ?
BSSLS
Calculate
Jj(| ν |), Jj(| κ ± mn σ |)
BES
BESNX
Calculate
Jj(| ν |), Jj(| κ ± mn σ |)
BES
κ ± mn σ = 0 ?
J1(κ ± mn σ) = 0
BES(502)
| κ ± mn σ |
ABSKAPA
40
ν > 0 ?
Jj(ν) = -Jj(| ν |) and Jj(κ ± mn σ) = −Jj(| κ ± mn σ |) for j ODD
BES
Jj(ν) = Jj(| ν |)
Jj(κ ± mn σ) = Jj(| κ ± mn σ |)
κ ± mn σ ≠ 0 ?
BSSLS
Calculate
J0(κ ± mn σ)
J1(| κ ± mn σ |)
J2(κ ± mn σ)
BES(501+j)
60
Calculate Σ j=1 JMAX (-1)^j (Jj(ν) [Jj+1 (κ ± mn σ) + Jj1(κ ± mn σ)]
CTEMP1
Calculate (2/ν) Σ JMAX (−1)^j Jj(ν) [Jj+1(κ ± mn σ) + Jj-1 (κ ± mn σ)]
SUM
B
C


B
Yes
No
BES (502)
BES (502)
BSSLS
BES(2)
Yes
No
Yes
No
BES (2)
Yes
Sum = −1
No
Yes
No
Yes
No
C
90
Calculate
CAPLT
RETURN

```
      SUBROUTINE NONCPT(B1,B2,B3,C2,CAPKMN,COSTHS,M,RHO,RNU,SINTHS,
     1                     CAPLT)
      COMPLEX CAPLT,CTEMP1,FNU,H1RNU,H2RNU,SNU
      COMMON/SCRATCH/BES(1000)
      DATA ISIGN,PI/-1,3.14159265358979/
      ABSNU = ABS(RNU)
      FNU = (0.,0.)
      SNU = (1.,0.)
      IF(RNU .EQ. 0.)  GO TO 10
      CALL  BSSLS(ABSNU,BES,1,IERR)
      BJ1 = BES(1)
      BJ2 = BES(2)
      CALL  BF4F(ABSNU,BES,1,IERR,ISIGN)
      BY1 = BES(1)
      BY2 = BES(2)
      H1RNU = CMPLX(BJ1,-BY1)
      H2RNU = CMPLX(BJ2,-BY2)
      CTEMP1 = CMPLX(BJ1-BY2,-BJ2-BY1)
      SNU = 1./(.5*PI*ABSNU*CTEMP1)
      IF(RNU .LT. 0.)  SNU = CONJG(SNU)
      CTEMP1 = (H1RNU + (0.,1.)*H2RNU) / (-H1RNU + (0.,1.)*H2RNU)
      TEMP1 = BJ1 - BJ2/ABSNU
      FNU = CTEMP1*CMPLX(TEMP1,-BJ2) - CMPLX(TEMP1,BJ2)
      IF(RNU .LT. 0.)  FNU = CONJG(FNU)
      RKAPA = (C2/2.)*(CAPKMN*COSTHS - (M*SINTHS)/RHO )
      TEMP1 = RNU + RKAPA
      BJ1 = 1.
      BJ2 = 0.
      TEMP3 = 1.
      IF(TEMP1 .EQ. 0.)  GO TO 20
      TEMP2 = ABS(TEMP1)
      CALL BSSLS(TEMP2,BES,1,IERR)
      BJ1 = BES(1)
      BJ2 = BES(2)
      IF(TEMP1 .LT. 0.)  BJ2 = -BJ2
      TEMP3 = (2.*BJ2) / TEMP1
20    IF(RNU*RKAPA .EQ. 0.)  GO TO 80
      ABSKAPA = ABS(RKAPA)
      JMAX = IFIX( AMAX1(ABSKAPA,ABSNU) ) + 1
      JMAX1 = JMAX + 1
      JMAX2 = JMAX + 2
      IF(JMAX1 .GT. 100)  GO TO 30
      CALL  BSSLS(ABSNU,BES,JMAX1,IERR)
      CALL  BSSLS(ABSKAPA,BES(501),JMAX2,IERR)
      GO TO 40
30    CALL  BESNX(JMAX1,ABSNU,BES)
      CALL  BESNX(JMAX2,ABSKAPA,BES(501))
40    IF(RNU .GT. 0.)  GO TO 60
      DO 50 I=2,JMAX2,2
      BES(I) = -BES(I)
50    BES(500+I) = -BES(500+I)
60    CTEMP1 = (0.,0.)
```

```
      DO 70 J=1,JMAX
      SIGN = 1.
      IF( MOD(J,2) .NE. 0)  SIGN = -1.
70    CTEMP1 = CTEMP1 + SIGN*BES(J+1)*( BES(502+J) + BES(500+J) )
      SUM = (2./RNU)*CTEMP1
      GO TO 90
80    IF(RKAPA .EQ. 0. )  BES(501) = 1.
      IF(RKAPA .EQ. 0. )  BES(502) = 0.
      ABSKAPA = ABS(RKAPA)
      IF(RKAPA .NE. 0.)  CALL BSSLS(ABSKAPA,BES(501),2,IERR)
      IF(RKAPA .LT. 0.)  BES(502) = -BES(502)
      IF(RNU .NE. 0.)  CALL BSSLS(ABSNU,BES,1,IERR)
      IF(RNU .LT. 0.)  BES(2) = -BES(2)
      IF(RNU.EQ.0. .AND. RKAPA.EQ.0.)  SUM= -1.
      IF(RNU.EQ.0. .AND. RKAPA.NE.0.)  SUM = -(BES(503) + BES(501))
      IF(RNU.NE.0. .AND. RKAPA.EQ.0.)  SUM = -(2./RNU)*BES(2)
90    CTEMP1 = CMPLX(BES(501),BES(502))
      CAPLT = B1*SNU*CTEMP1 + B2*CMPLX(BJ1,BJ2)
     1        +B3*(CTEMP1*FNU + TEMP3 - SUM)
      RETURN
      END
```

3.3 Secondary General-Purpose Subprogram Descriptions

3.3.1 Subroutine APROX1

Purpose: This subroutine evaluates the asymptotic expression, formulas (9.5.28) and (9.5.31) of reference 30, for the zeros of the function:

$$J'_\nu(Z)\, Y'_\nu(\lambda Z) - J'_\nu(\lambda Z)\, Y'_\nu(Z)$$

for $\lambda \leqq 5$, where $Z \sim \beta + \frac{p}{\beta} + \frac{q - p^2}{\beta^3} + \frac{r - 4pq + 2p^3}{\beta^5}$

$$\mu = 4\nu^2, \quad \beta = \frac{S\pi}{\lambda - 1}$$

with S equal to the ordinal number of zero when $\nu = 0$

$$p = \frac{\mu + 3}{8\lambda}, \quad q = (\mu^2 + 46\mu - 63)(\lambda^3 - 1)$$

$$r = \frac{(\mu^3 + 185\mu^2 - 2053\mu + 1899)(\lambda^5 - 1)}{5(4\lambda)^5\,(\lambda - 1)}$$

Method: The procedure is as follows:

1) Given η, $.2 \leqq \eta < 1$, calculate $\lambda = \frac{1}{\eta}$, $\lambda - 1$, μ, μ^2, and β.

2) Calculate p, q, r.

3) Calculate Z.

4) Multiply Z by λ and output $Z\lambda$.

Usage: CALLING SEQUENCE

```
CALL APROX1 (RM,NS,ETA,RZ)
```

INPUT

RM the value of ν

NS the value of S, a positive integer

ETA the value of η, where $\lambda = 1/\eta$

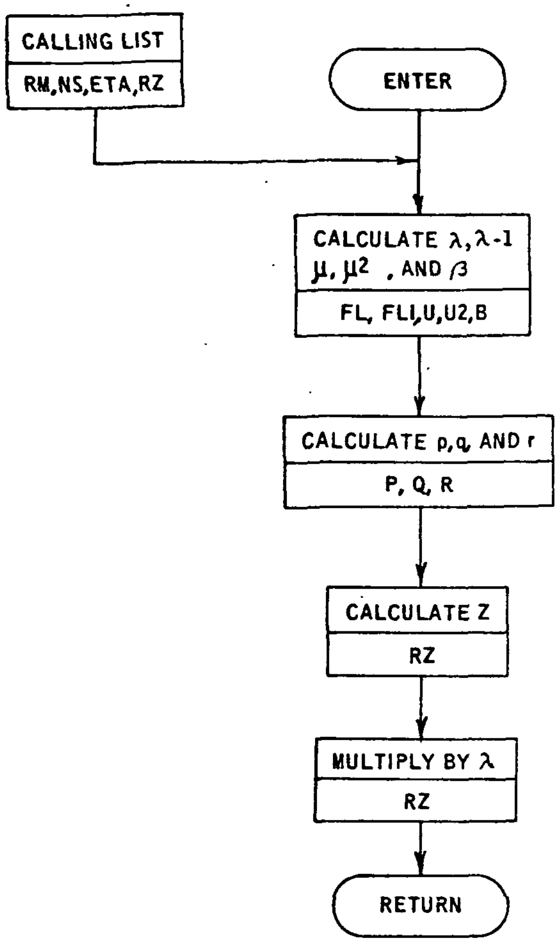
OUTPUT

RZ the computed value of $Z\lambda$

Restrictions: $.2 \leq \eta < 1$

Timing: The time is proportional to the number of arithmetic operations, which is 48 multiplications, 8 divisions, 7 additions, and 7 subtractions.

Accuracy: The accuracy is of the computer type.



CALLING LIST
RM,NS,ETA,RZ
ENTER
CALCULATE λ,λ-1 μ, μ2 , AND β
FL, FLI,U,U2,B
CALCULATE p,q, AND r
P, Q, R
CALCULATE Z
RZ
MULTIPLY BY λ
RZ
RETURN

```
      SUBROUTINE APROX1(RM,NS,ETA,RZ)
C
C
C PURPOSE        APPLY APPROXIMATION FORMULA TO THE ZEROS OF THE EQUATION
C
C                 JP(M,X)*YP(M,ETA*X)-YP(M,X)*JP(M,ETA*X) = R(M,X)
C
C                WHERE JP AND YP ARE THE DERIVATIVES OF THE BESSEL
C                FUNCTIONS OF THE FIRST AND SECOND KIND, RESPECTIVELY,
C                OF ORDER M AND ARGUMENT X OR ETA*X. THE APPROXIMATION
C                EXPANSION IS FORMULA 9.5.31 OF THE REFERENCE. IN
C                APPLYING THE EXPANSION, THE VALUE OF ETA IS RESTRICTED
C                TO BE AT LEAST .2 BUT LESS THAN 1 ( ETA = 1/LAMBDA IN
C                REFERENCE ). THE FORMULA IS CODED FOR REAL NON-NEGATIVE
C                ORDER RM BUT SHOULD BE USED FOR VERY SMALL ORDER,
C                AS ZERO HERE
C
C REFERENCE      HANDBOOK OF MATHEMATICAL FUNCTIONS EDITED BY
C                 M. ABRAMOWITZ AND I. STEGUM, NATIONAL BUREAU OF
C                 STANDARDS APPLIED MATHEMATICS SERIES 55 ISSUED JUNE 1964
C
C INPUT          VARIABLE  DEFINITION
C                      RM    REAL ORDER M, SHOULD BE SMALL
C                      NS    FIND THE NS-TH ZERO APPROXIMATION, THE LARGER
C                             NS W.R.T. THE ORDER, THE BETTER THE APPROX.
C                      ETA    AT LEAST .2 ( TO FIND NS-TH ZERO, ESP. FOR
C                             NS SMALL ) AND LESS THAN 1 ( THIS IS RATIO
C                             OF INNER TO OUTER RADII IN ANNULAR DUCT ).
C
C OUTPUT               RZ    THE APPROXIMATION TO THE ZERO
C
C RESTRICTIONS         THE RESTRICTION PLACED ON THE INPUT ABOVE
C                       GUARENTEE AN VALID APPROX. TO NS-TH ZERO IS FOUND
C                       (PLACING ETA=.15, RM=0., NS=1 WILL GIVE ESTIMATE
C                        NOT OF FIRST BUT MUCH HIGHER ZERO ).
C
      DATA PI/017216220773250420551/
C
      FL=1./ETA
      FL1=FL-1.
      U=4.*RM*RM
      U2 = U*U
      B =  NS*PI/FL1
C
      P= (U+3.)/(8.*FL)
      Q = (U2+46.*U-63.)*(FL**3-1.)/(6.*((4.*FL)**3)*FL1)
      R=  (U**3+185.*U2-2053.*U+1899.)*(FL**5 -1.)/(5.*((4.*FL)**5)*FL1)
C
      RZ = B + P/B + (Q-P** 2)/B**3 + (R-4.*P*Q+2.*P**3)/B**5
      RZ = RZ*FL
C
      RETURN
      END
```

3.3.2 Subroutine APROX2

Purpose: This subroutine computes an approximate value for one zero of the ordered set of zeros of the cross-product function:

$$J_o'(X)\ Y_o'(\eta X) - Y_o'(X)\ J_o'(\eta X),$$

when $\eta < 0.2$, where J_o and Y_o are, respectively, the Bessel and Neumann functions of the zeroth order, and primes denote differentiation with respect to the argument. For $\eta = 0$, formula (9.5.13) of reference 30 is used. For $0 < \eta < .2$, quadratic interpolation is used with the values of $\eta = 0$, obtained from this routine, and for $\eta = .2$ and $.3$, obtained from subroutine APROX1 (see preceding description of APROX1).

Method: The procedure is as follows:

1) If the ordinal number of the zero is one, equate the zeros to stored values.

2) If the ordinal number of the zero is not one and the input $\eta > 0$, obtain approximate values for the zeros for the table values $\eta = .2$ and $.3$ from subroutine APROX1.

3) If the ordinal number of the zero is not one, compute the approximate value for the zero for the table value $\eta = 0$ using formula (9.5.13) of reference 30.

4) If the input $\eta = 0$, return the computed value from step 3.

5) Compute the approximate value for the zero by quadratic interpolation.

Usage: CALLING SEQUENCE

```
CALL APROX2(RM,NS,ETA,RZ)
```

INPUT

RM 0.

NS the n^{th} positive zero is to be approximated

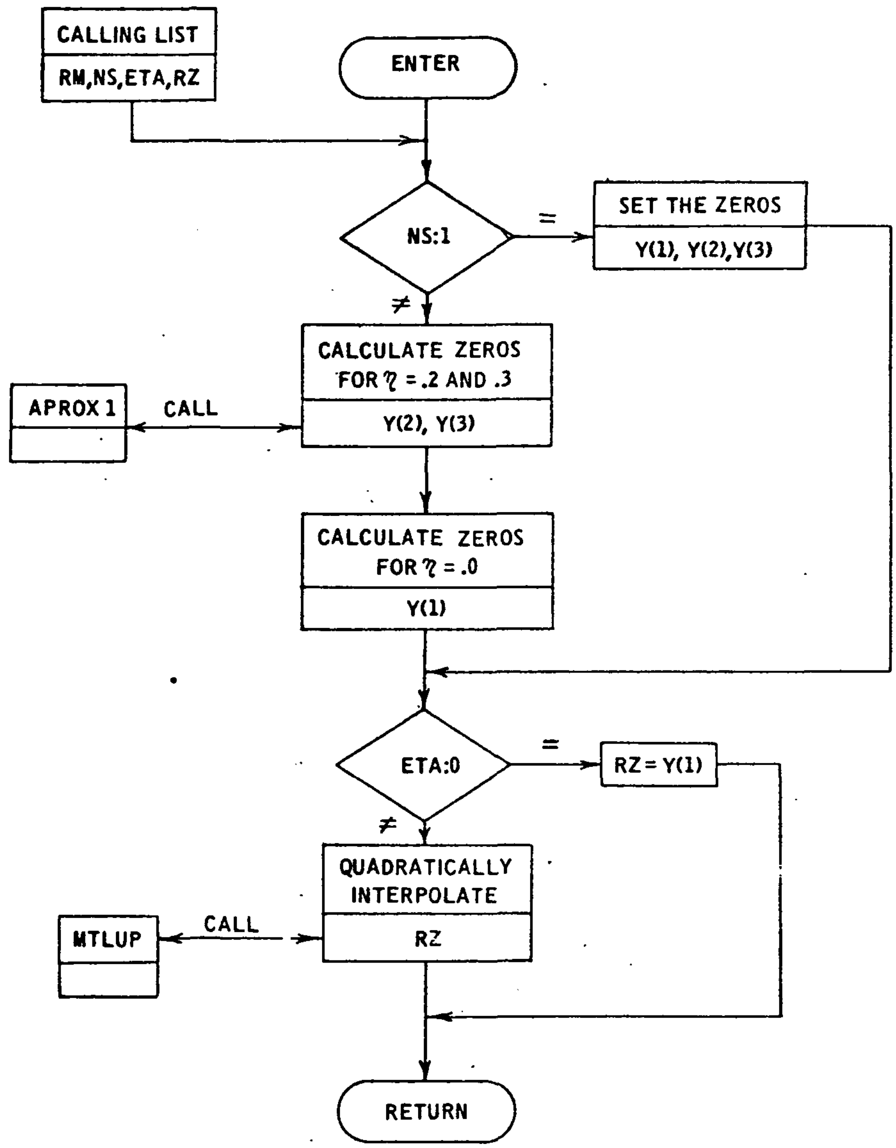
ETA the hub-to-tip ratio, $.0 \leq \eta < .2$

OUTPUT

RZ the corresponding approximation

Timing: For $\eta = 0$, the time is equal to that of APROX1; for $\eta > 0$, the time is three unit times for APROX1 plus the unit time for MTLUP.

Accuracy: The accuracy is of the statistical type for the interpolation.


CALLING LIST
RM,NS,ETA,RZ
ENTER
NS:1
=
SET THE ZEROS
Y(1), Y(2),Y(3)
≠
CALCULATE ZEROS
FOR η = .2 AND .3
Y(2), Y(3)
APROX 1
CALL
CALCULATE ZEROS
FOR η = .0
Y(1)
ETA:0
=
RZ = Y(1)
≠
QUADRATICALLY
INTERPOLATE
RZ
MTLUP
CALL
RETURN

```
      SUBROUTINE APROX2(RM,NS,ETA,RZ)
C
C
CPURPOSE          FIND AN APPROXIMATION TO THE ZEROS OF
C
C                  JP(M,X)*YP(M,ETA*X)-YP(M,X)*JP(M,ETA*X)
C                 IN THE REGION WHERE APROX1 FAILS
C                 WHERE JP AND YP ARE THE DERIVATIVES OF THE BESSEL
C                 FUNCTIONS OF THE FIRST AND SECOND KIND, RESPECTIVELY,
C                 OF ARGUMENT X AND ETA*X, ETA BETWEEN 0 AND 1,
C
CREFERENCE        HANDBOOK OF MATHEMATICAL FUNCTIONS EDITED BY
C                 M. ABRAMOWITZ AND I. STEGUM, NATIONAL BUREAU OF
C                 STANDARDS APPLIED MATHEMATICS SERIES 55 ISSUED JUNE 1964
CMETHOD           SUBROUTINE APROX1 USES FORMULA 9.5.31 FROM THE REFERENCE
C                 TO APPROXIMATE THE NS-TH ZERO OF THE ABOVE EQUATION
C                 BUT THIS FAILS FOR ETA BELOW .2 . TO CORRECT THIS PROBLEM
C                 THIS ROUTINE IS PROVIDED. THE APPROXIMATION TO THE ZERO
C                 IS FOUND BY QUADRATIC INTERPOLATION USING THE
C                 APPROXIMATIONS FOR ETA .2 AND .3 AND THE APPROXIMATION
C                 FORMULA 9.5.13 OF THE REFERENCE FOR JP(M,X) = 0 WHICH
C                 CORRESPONDS TO ETA=0. FOR THE FIRST POSITIVE ZERO, THE
C                 CORRESPONDING ZEROS ARE TABULATED BECAUSE THE
C                 APPROXIMATION FORMULA ARE POOR FOR THE FIRST ZERO.
C
CINPUT            VARIABLE  DEFINITION
C                      RM    BESSEL ORDER M AND IS 0 HERE
C                      NS    THE NS-TH POSITIVE ZERO IS TO BE APPROXIMATED
C                      ETA   HUB TO TIP RATIO, 0 .LT. ETA .LT. .2
C
COUTPUT                RZ    APPROXIMATION TO ZERO
C
CSUBPROGRAMS      APROX1    APPROXIMATION FOR ETA.GE. .2
C
C                 MTLUP    LRC LIBRARY INTERPOLATOR
C
C
      DIMENSION X(3),Y(3)
      DATA X/0.,.2,.3/
      NOETAT=NS
      IF(NOETAT-1)10,10,20
C
C                USE EXACT ZEROS FOR NS=1, THE APPROXIMATION IS TOO POOR
C
   10 Y(1) =3.8317 $ Y(2)= 4.2357 $ Y(3) =4.7058
      GO TO 30
C
C                USE APROX1 FOR ZERO APPROXIMATION AT ETA .2 AND .3
C
   20 IF(ETA.EQ.0.) GO TO 25
      CALL APROX1(RM,NOETAT,X(2),Y(2) )
      CALL APROX1(RM,NOETAT,X(3),Y(3) )
```

```
C
C               APPLY REFERENCE FORMULA 9.5.13 FOR ETA O AND M=0
C
   25  BETAP = (NOETAT        +.25)*3.14159265
       BETAP8 = 8.*BETAP
       TERM1 =-3./BETAP8
       TERM2 = 36./(3.*BETAP 8**3)
       TERM3 =-113184./(15.*BETAP8**5)
       TERM4 = 374632128./(105.*BETAP8**7)
       Y(1)  = BETAP+TERM1+TERM2+TERM3+TERM4
C
   30 IF(ETA) 40,40,50
   40 RZ = Y(1)
      GO TO 60
   50 IPA = -1
      CALL MTLUP(ETA,RZ,2,3,3,1,IPA,X,Y)
C
   60 RETURN
      END
```

3.3.3 Subroutine JARRATT

Purpose: This subroutine computes a single, real zero of a real valued, nonlinear function, i.e., it computes X such that $f(X) < \varepsilon$, with ε a controllably small number. The method is that of reference 47. This is an iterative method in which a rational function is fitted through previously computed values, giving the iteration formula:

$$X_{n+1} = X_n + \frac{(X_n - X_{n-1})(X_n - X_{n-2})f_n(f_{n-1} - f_{n-2})}{(X_n - X_{n-1})(f_{n-2} - f_n)\, f_{n-1} + (X_n - X_{n-2})(f_n - f_{n-1})f_{n-2}}$$

where f_n is $f\left(X_n\right)$.

Method: The procedure is as follows:

1) Set the perturbation value used in step 4.

2) Initialize the error return (see ERROR subsection of this routine description), the iteration counter, and the counter used in the subloop of step 4.

3) Generate the iteration values and corresponding function values required in the initial evaluation of the iteration formula by equating the first three iterates to the ordered triplet of input starting values and computing the function values.

4) Test for equal function values, changing one of them when this happens by reevaluation with the argument equal to the corresponding iterate plus the perturbation constant from step 1. This procedure should be repeated at most three times (see ERROR subsection).

5) Compute the current iteration value for the zero with the iteration formula.

6) Compute the function value for the argument equal to the current iterate.

7) Test for convergence of iteration by comparing percentage difference between new and old iterates with an input tolerance or by comparing the function value from step 6 with an input tolerance, and exit from the algorithm with the current iterate when the test is successful.

8) Compare iteration counter to limit and exit from the algorithm when the limit is exceeded, accompanied by an error message.

9) Generate new iteration values and corresponding function values, add one to the iteration counter, and start over with step 4.

Usage: CALLING SEQUENCE

```
      DIMENSION START(3)
      .
      .
      .
      CALL JARRATT(START,MAXITER,TOLITER,TOLFUN,FUN,ROOT,FUNROOT,
     *   IERJAR)
```

INPUT

START(3) an array of three nonequal starting values for the iterates X_1, X_2, X_3

MAXITER maximum number of iterations allowed

TOLITER maximum relative difference between consecutive iterates

TOLFUN	tolerance on the absolute function value
FUN	the function generator; this must be EXTERNAL and of the form FUNCTION FUN (X)

OUTPUT

ROOT	the value of the zero when TOLITER or TOLFUN tolerance is satisfied
FUNROOT	the function value corresponding to ROOT

ERROR

IERJAR

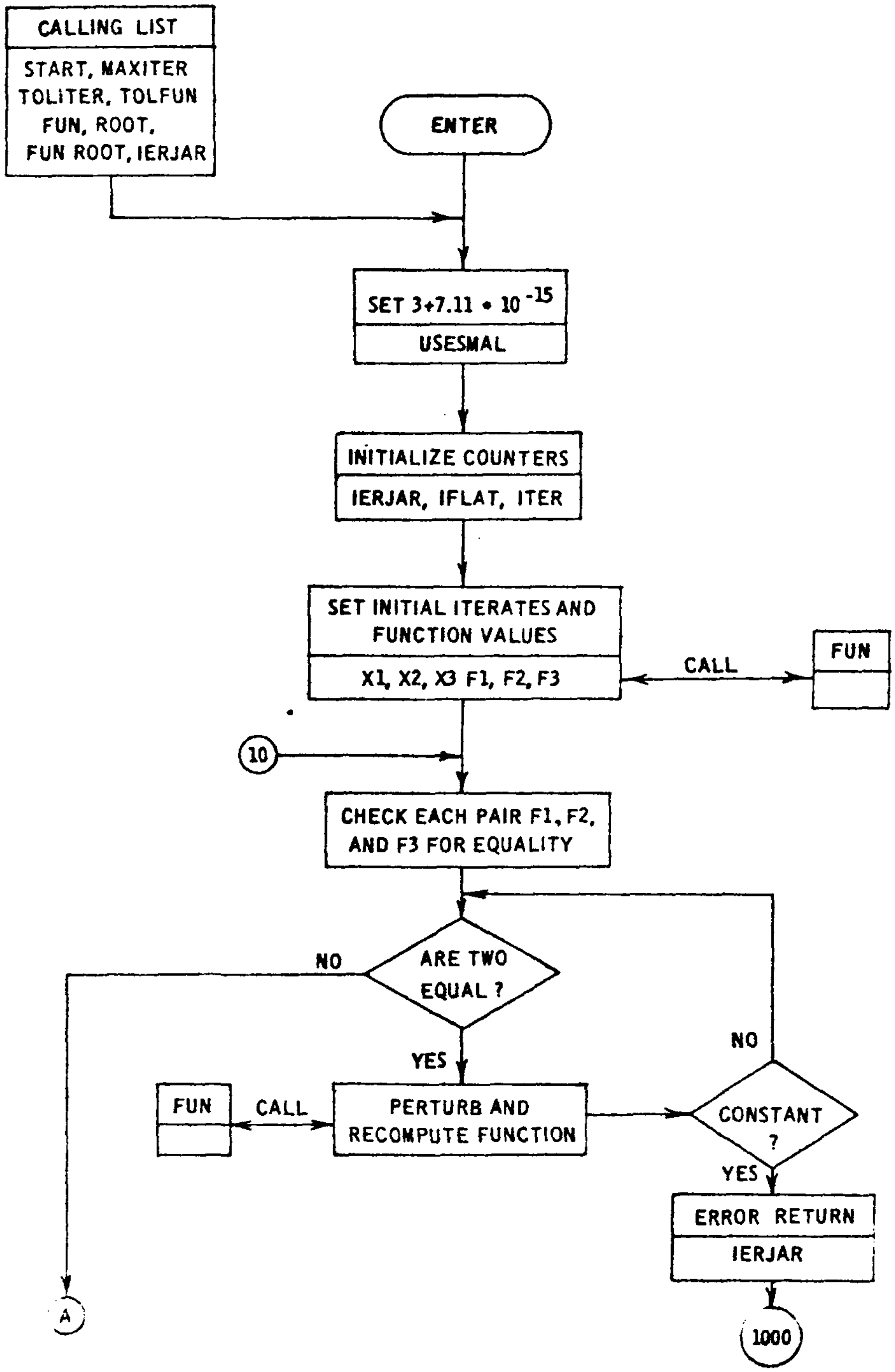
Errors: Upon return, the error return parameter is set as follows:

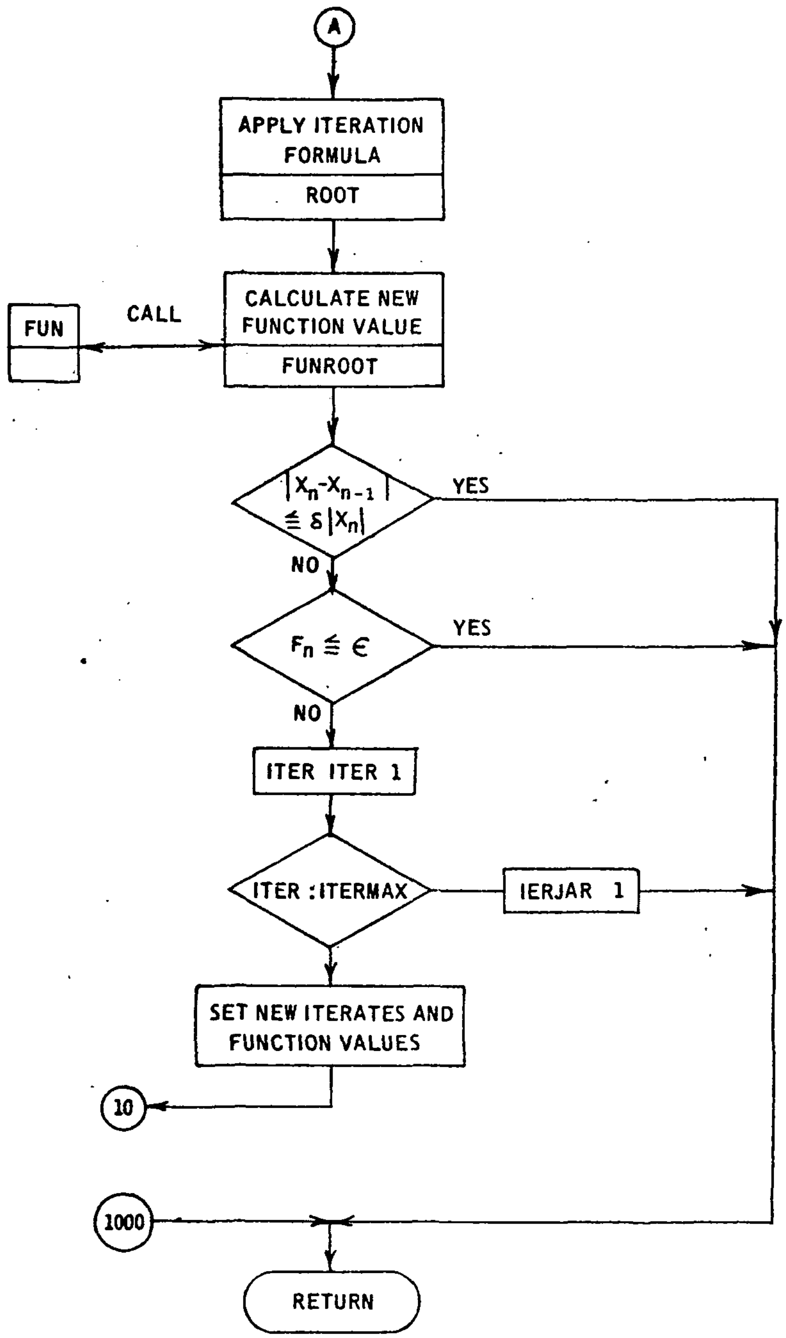
IERJAR =	0	successful
	1	the convergence criterion was not satisfied within the maximum number of iterations allowed
	2	the function appears constant

Restrictions: START(1) $\neq$ START(2) $\neq$ START(3)

Timing: The timing is proportional to the number of iterations multiplied by the execution time per call to FUN. Hence, good starting values are important to timing.

Accuracy: The accuracy is set by the input tolerances.


CALLING LIST
START, MAXITER
TOLITER, TOLFUN
FUN, ROOT,
FUN ROOT, IERJAR
ENTER
SET 3+7.11 • 10^-15
USESMAL
INITIALIZE COUNTERS
IERJAR, IFLAT, ITER
SET INITIAL ITERATES AND FUNCTION VALUES
X1, X2, X3 F1, F2, F3
CALL
FUN
10
CHECK EACH PAIR F1, F2, AND F3 FOR EQUALITY
ARE TWO EQUAL ?
NO
YES
FUN
CALL
PERTURB AND RECOMPUTE FUNCTION
CONSTANT ?
NO
YES
ERROR RETURN
IERJAR
1000
A


A
APPLY ITERATION FORMULA
ROOT
CALCULATE NEW FUNCTION VALUE
FUNROOT
FUN
CALL
$|X_n - X_{n-1}| \leqq \delta |X_n|$
YES
NO
$F_n \leqq \epsilon$
YES
NO
ITER ITER 1
ITER : ITERMAX
IERJAR 1
SET NEW ITERATES AND FUNCTION VALUES
10
1000
RETURN

```
      SUBROUTINE JARRATT(START,MAXITER,TOLITER,TOLFUN,FUN,ROOT,FUNROOT,
     1IERJAR)
C
CPURPOSE        TO FIND THE ZERO OF A SINGLE REAL VALUED FUNCTION OF
C               A REAL VARIABLE BASED UPON JARRATT METHOD
C
CINPUT          VARIABLE DEFINITION
C                START     AN ARRAY OF THREE STARTING VALUES FOR XERO
C                MAXITER   MAXIMUM NUMBER OF ITERATIONS USED
C                TOLITER   RELATIVE TOLERANCE ON THE CLOSENESS OF TWO
C                          SUCCESSIVE ITERATES
C                TOLFUN    TOLERANCE ON SMALLNESS OF FUNCTION VALUE
C                FUN       EXTERNAL FUNCTION EVALUATOR, WHERE Y = FUN(X)
COUTPUT          ROOT      ZERO CALCULATED
C                FUNROOT   FUNCTION VALUE CORRESPONDING TO ROOT
CERROR RETURN    IERJAR     = 0 SUCCESSFUL EXECUTION
C                             1 FAIL TO CONVERGE IN MAXITER ITERATIONS
C                             2 FUNCTION APPEARS CONSTANT
C
CREFERENCE      P. JARRATT AND D. NUDDS, THE USE OF RATIONAL FUNCTIONS
C               IN THE ITERATIVE SOLUTION OF EQUATIONS ON A DIGITAL
C               COMPUTER, THE COMPUTER JOURNAL, APRIL 1965, VOL. 8, NO. 1
C
C
      DIMENSION START(3)
C
C     C         THIS DATA STATEMENT DEFINES SMALLEST NUMBER
C               SIGNIFICANTLY ADDING TO 1.0
C
      DATA SMALL/7.11E-15/
      USESMAL = 3. + SMALL
C
C               INITIALIZE ERROR RETURN AND CONSTANT FUNCTION INDICATOR
C               AND ITERATION COUNTER
C
      IERJAR = 0
      IFLAT = 0
      ITER  = 0
C               SET STARTING FUNCTION VALUES
C
      X1 = START(1)
      X2 = START(2)
      X3 = START(3)
      F1 = FUN(X1)
      F2 = FUN(X2)
      F3 = FUN(X3)
C
C               CHECK FOR EQUAL FUNCTION VALUES, WHEN TWO ARE EQUAL
C               PERTURB THE STARTING VALUE AND RE-EVALUATE THE FUNCTION,
C               AND DO THIS AT MOST 3 TIMES
C
   10 IF(F1.EQ.F2) GO TO 20
```

```
      IF(F1.EQ.F3) GO TO 20
      IF(F2.EQ.F3) GO TO 20
      GO TO 50
   20 IFLAT = IFLAT + 1
      IF(IFLAT.LT.3) GO TO 25
      IERJAR = 2
      GO TO 1000
   25 IF(F1.NE.F2) GO TO 30
      X2 = (X1+X2)/USESMAL
      F2 = FUN(X2)
   30 IF(F1.NE.F3) GO TO 35
      X3 = (X1+X3)/USESMAL
      F3 = FUN(X3)
   35 IF(F2.NE.F3) GO TO 50
      X3 = (X2+X3)/USESMAL
      F3 = FUN(X3)
      GO TO 20
   50 CONTINUE
C
C                PERFORM JARRATT ITERATION
C
      X12 = X1-X2
      X13 = X1-X3
      ROOT=X1+(X12*X13*F1*(F2-F3))/(X12*(F3-F1)*F2+X13*(F1-F2)*F3)
      FUNROOT = FUN(ROOT)
C
C                CHECK FOR CONVERGENCE
C
      IF( ABS(ROOT-X1).LE.TOLITER*ABS(ROOT))GO TO 1000
      IF( ABS(FUNROOT).LE.TOLFUN) GO TO 1000
C
C                CHECK MAX ITERATION
C
      ITER = ITER+1
      IF(ITER-MAXITER)60,60,55
   55 IERJAR = 1
      GO TO 1000
C
C                UPDATE LIST OF VARIABLE AND FUNCTION VALUES
C
   60 X3 = X2
      X2 = X1
      X1 = ROOT
      F3 = F2
      F2 = F1
      F1 = FUNROOT
      GO TO 10
C
 1000 RETURN
      END
```

3.3.4 Subroutine GAUSS

Purpose: This subroutine computes the definite integral of a complex valued function of a single, real variable using either 4-, 8-, or 12-point Gaussian integration formulas (formula [25.4.30] on page 887 of ref. 30).

Method: The procedure is as follows:

1) Obtain the weights for 4-, 8-, and 12-point Gaussian integration.

2) Compute the half-width and midpoint of the integration interval.

3) Obtain the 4-, 8-, and 12-point abscissas.

4) If 4-point integration, go to step 5;
   If 8-point integration, go to step 6;
   If 12-point integration, go to step 7.

5) Perform 4-point Gaussian integration and go to step 8.

6) Perform 8-point Gaussian integration and go to step 8.

7) Perform 12-point Gaussian integration.

8) Change sign if integration was from right to left and return.

Usage: CALLING SEQUENCE

```
COMPLEX ANS,FUN
EXTERNAL FUN
•
•
•
CALL GAUSS(A,B,ANS,FUN,INT)
```

INPUT

A lower limit of integral

B upper limit of integral

FUN name of the complex function subprogram which calculates the integrand

INT indicator for order of Gaussian integration:

INT = 1 indicates 4 point
2 indicates 8 point
3 indicates 12 point

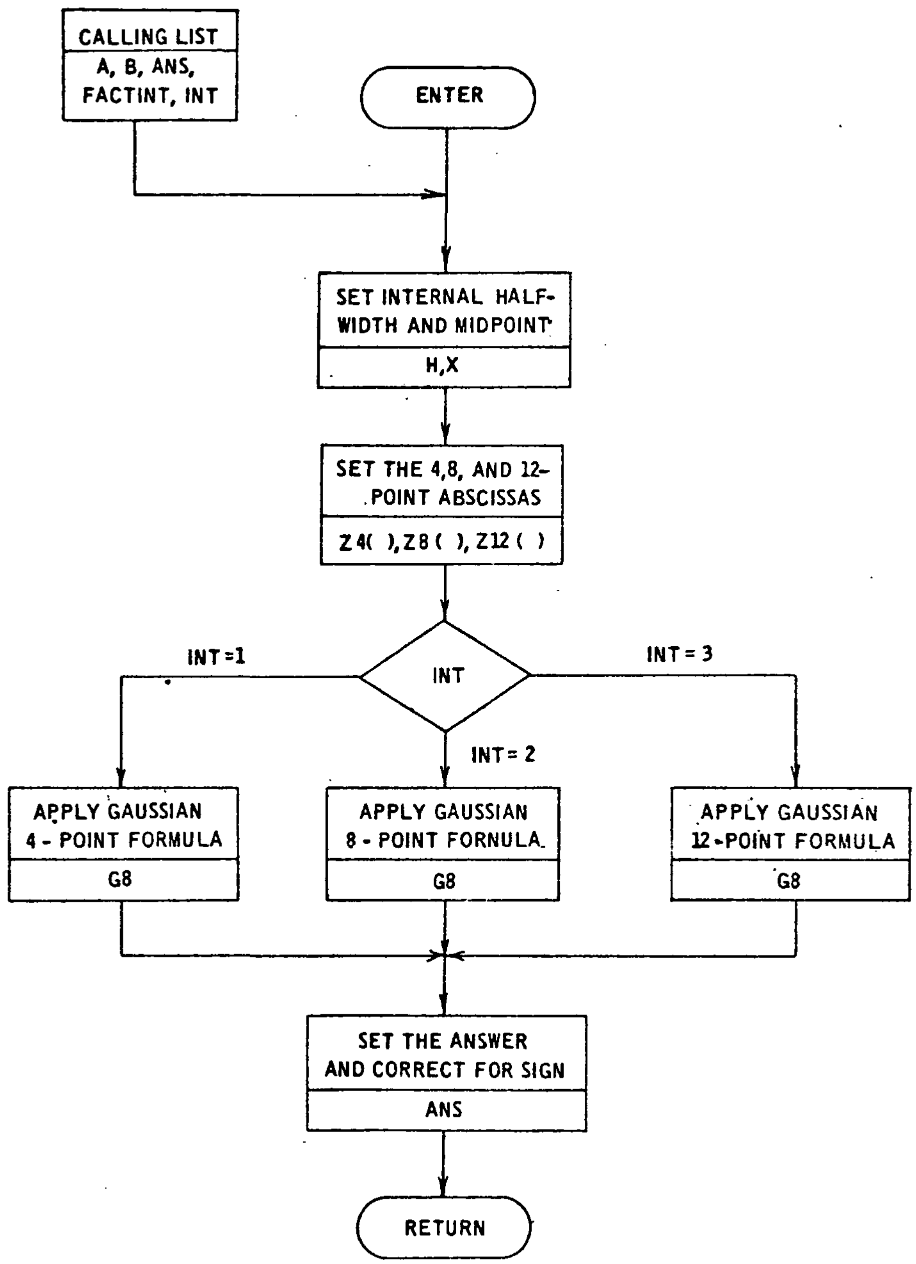
OUTPUT

ANS the value of the definite integral

Timing: N point--N x time for FUN

Accuracy: The remainder term for the N-point Gaussian integration formula is:

$$\frac{(B-A)^{2N+1}(N!)^4 2^{2N+1}}{(2N+1)\left[(2N)!\right]^3} f^{(2N)}(\xi)$$


CALLING LIST
A, B, ANS,
FACTINT, INT
ENTER
SET INTERNAL HALF-WIDTH AND MIDPOINT
H,X
SET THE 4,8, AND 12-POINT ABSCISSAS
Z4( ),Z8( ),Z12( )
INT
INT=1
INT=2
INT=3
APPLY GAUSSIAN 4-POINT FORMULA
G8
APPLY GAUSSIAN 8-POINT FORNULA
G8
APPLY GAUSSIAN 12-POINT FORMULA
G8
SET THE ANSWER AND CORRECT FOR SIGN
ANS
RETURN

```
      SUBROUTINE GAUSS(A,B,ANS,FACTINT,INT)
C
      COMPLEX ANS,FACTINT,G4,G8,G12,Z1,Z2
C
C               4-, 8-, AND 12-POINT GAUSSIAN WEIGHTING COEFFICIENTS
C
      DIMENSION W4(2),W8(4),W12(6), Z4(2),Z8(4), Z12(6)
      DATA W4(1),W4(2),(W8(I),I=1,4),(W12(I),I=1,6)/.652145154862546,
     1.347854845137454,.362683783378362,.313706645877387,.222381034453337
     14,  .101228536290376,.249147045813403,.233492536533355,
     1.203167426723066,.160078328543346,.106939325995318,
     1.047175336386512/
C
      Y = A
      H = (B-Y)/2.
      SGN=SIGN(1.,H)
      H=ABS(H)
      X = Y + H*SGN
C
C               4-POINT ABSCISSAE
C
      Z4(1)=.339981043584856*H
      Z4(2)=.861136311594053*H
C
C               8-POINT ABSCISSAE
C
      Z8(1)=.183434642495650*H
      Z8(2)=.525532409916329*H
       Z8(3)=.796666477413627*H
      Z8(4)=.960289856497536*H
C
C               12-POINT ABSCISSAE
C
      Z12(1)=.125233408511469*H
      Z12(2)=.367831498998180*H
      Z12(3)=.587317954286617*H
      Z12(4)=.769902674194305*H
      Z12(5)=.904117256370475*H
      Z12(6)=.981560634246719*H
C
C               EVALUATE FUNCTION AND PERFORM WEIGHTED SUM
C
      GO TO (10,20,40) INT
 10   CONTINUE
      G4=H*(W4(1)*(FACTINT(X+Z4(1))+FACTINT(X-Z4(1)))+
     1W4(2)*(FACTINT(X+Z4(2))+FACTINT(X-Z4(2))))
      G8=G4
      GO TO 60
 20   CONTINUE
      G8 = C.
      DO 30 I=1,4
      Z1=FACTINT(X+Z8(I))
```

```
      Z2=FACTINT(X-Z8(I))
30    G8=G8+W8(I)*(Z1+Z2)
      G8=G8*H
      GO TO 60
40    CONTINUE
      G12=0
      DO 50 I=1,6
50    G12=G12+W12(I)*(FACTINT(X+Z12(I))+FACTINT(X-Z12(I)))
      G12=G12*H
      G8=G12
60    CONTINUE
      ANS=G8
      IF(B-A.LT.O.) ANS=-ANS
      RETURN
      END
```

3.3.5 Subroutine GAUSS2

Purpose: This subroutine has the same purpose as subroutine GAUSS. It is a modification of GAUSS to pass the primary subroutine input to FACTINT, FACTIN2, FACTIN3, and FACTIN4.

Method: Same as subroutine GAUSS

Usage: CALLING SEQUENCE

```
COMPLEX ANS,FACTIN2
EXTERNAL FACTIN2
•
•
•
CALL GAUSS2(A,B,INT,ANS,FACTIN2,ARMISC,MAXDIM,MAXJ,AR)
```

INPUT

A	lower limit of integral
B	upper limit of integral
INT	indicator for order of Gaussian integration: INT = 1 indicates 4 point 2 indicates 8 point 3 indicates 12 point
FACTIN2	general name for any of the oscillatory factor evaluators named in the purpose

ARMISC, MAXDIM, MAXJ, AR (see FORTRAN dictionary, sec. 2.2)

OUTPUT

ANS	the value of the definite integral

Storage: 513 (octal)

Timing: N point--N x time for FACTIN2

Accuracy: Same as subroutine GAUSS

Flowchart: See subroutine GAUSS.

```
      SUBROUTINE GAUSS2(A,B,INT,ANS,FACTIN2,ARMISC,MAXDIM,MAXJ,AR)
C
      COMPLEX ANS,FACTIN2,G4,G8,G12,Z1,Z2
      DIMENSION AR(MAXDIM,MAXJ,3),ARMISC(1)
C
C              4-, 8-, AND 12-POINT GAUSSIAN WEIGHTING COEFFICIENTS
C
      DIMENSION W4(2),W8(4),W12(6), Z4(2),Z8(4), Z12(6)
      DATA W4(1),W4(2),(W8(I),I=1,4),(W12(I),I=1,6)/.652145154862546,
     1.347854845137454,.362683783378362,.313706645877887,.2223810344533
     14,  .101228536290376,.249147045813403,.233492536538355,
     1.203167426723066,.160078328543346,.106939325995318,
     1.047175336386512/
C
      Y = A
      H = (B-Y)/2.
      SGN=SIGN(1.,H)
      H=ABS(H)
      X = Y + H*SGN
C
C              4-POINT ABSCISSAE
C
      Z4(1)=.339981043584856*H
      Z4(2)=.861136311594053*H
C
C              8-POINT ABSCISSAE
C
      Z8(1)=.183434642495650*H
      Z8(2)=.525532409916329*H
       Z8(3)=.796666477413627*H
      Z8(4)=.960289856497536*H
C
C              12-POINT ABSCISSAE
C
      Z12(1)=.125233408511469*H
      Z12(2)=.367831498998180*H
      Z12(3)=.587317954286617*H
      Z12(4)=.769902674194305*H
      Z12(5)=.904117256370475*H
      Z12(6)=.981560634246719*H
C
C              EVALUATE FUNCTION AND PERFORM WEIGHTED SUM
C
      GO TO (10,20,40) INT
 10   CONTINUE
      G4 = H*( W4(1)*( FACTIN2(X+Z4(1),ARMISC,MAXDIM,MAXJ,AR)
     1                + FACTIN2(X-Z4(1),ARMISC,MAXDIM,MAXJ,AR) )
     2        + W4(2)*( FACTIN2(X+Z4(2),ARMISC,MAXDIM,MAXJ,AR)
     3                + FACTIN2(X-Z4(2),ARMISC,MAXDIM,MAXJ,AR) ) )
      G8=G4
      GO TO 60
 20   CONTINUE
```

```
      G8 = C.
      DO 30 I=1,4
      Z1 = FACTIN2(X+Z8(I),ARMISC,MAXDIM,MAXJ,AR)
      Z2 = FACTIN2(X-Z8(I),ARMISC,MAXDIM,MAXJ,AR)
30    G8=G8+W8(I)*(Z1+Z2)
      G8=G8*H
      GO TO 60
40    CONTINUE
      G12=0
      DO 50 I=1,6
50    G12 = G12 + W12(I)*( FACTIN2(X+Z12(I),ARMISC,MAXDIM,MAXJ,AR)
     1                   + FACTIN2(X-Z12(I),ARMISC,MAXDIM,MAXJ,AR) )
      G12=G12*H
      G8=G12
60    CONTINUE
      ANS=G8
      IF(B-A.LT.0.) ANS=-ANS
      RETURN
      END
```

3.3.6 Subroutine BSSLS

Purpose: This subroutine computes values for the first n Bessel functions of integer order for the real argument x: $J_o(x)$, $J_1(x)$, . . . , $J_n(x)$.

This subroutine is a modification of the NASA-Langley Research Center library subroutine BSSLS (see ref. 43). The restriction on the order has been removed from the library routine. The calling sequence has not been changed. The usage of the modified routine differs from the library routine in that:

1) Orders greater than 30 can be used while the error code remains equal to 0.

2) A deck of this modified routine must be loaded with the source deck.

Values produced by this routine for orders up to 100 for arguments up to 100 were compared with the published tables on page 407 of reference 30 and agreed in the first nine significant figures. This represents the justification for the use of this modified routine.

```
      SUBROUTINE BSSLS (X,F,N,IERR)
C
C           COMPUTES BESSEL FUNCTIONS OF THE FIRST KIND FOR POSITIVE
C           REAL ARGUMENT AND INTEGER ORDER
C
C           THIS IS A MODIFIED VERSION OF SUBROUTINE BSSLS.  THE
C           RESTRICTION ON THE ORDER,N, HAS BEEN REMOVED FROM THE
C           STANDARD LRC LIBRARY VERSION.  THE STATEMENTS REMOVED
C           FROM THE LRC LIBRARY VERSION HAVE BEEN MADE INTO COMMENT
C           STATEMENTS DELIMITTED BY **.  THIS WAS DONE ON
C           MAY 8, 1973 BY GEORGE A. GRAF OF BCS.
C
      DIMENSION F(1)
      COMMON/FIX/NPR,NP,NPP
      IERR=0
C **  NMAX=30   **
C **  IF(N.LE.NMAX)GO TO 701   **
C **  IERR=1 **
C **  RETURN **
  701 MX=X
C           NO. OF FUNCTIONS COMPUTED
      NPP=3*MX+12+10*(IABS(N-1)/10)
      IF(X.GT.N)NPP=3.0*X+12.
      IF(MOD(NPP,2).EQ.0)NPP=NPP+1
C           CLEAR COMPUTING AREA
      DO 702 I=1,NPP
  702 F(I)=0.0
      IF(X.NE.0.)GO TO 700
C           X=0
      F(1)=1.0
      RETURN
  700 IF(X.GE..1E-6)GO TO 703
C           SMALL VALUES OF X
C           J=Z**N/FACTORIAL N
      Z=X/2.0
      F(1)=1.0
      LPP=NPP-1
      DO 704 K=1,LPP
  704 F(K+1)=F(K)*(Z/FLOAT(K))
      RETURN
C           BACKWARD RECURSION
  703 NP=NPP+1
      NPR=NPP-1
      F(NP-1)=.1E-37
      F(NP)=0.0
      DO 11 I=1,NPR
      NP=NPP-I
      XN=NP
   11 F(NP)=2.0*XN/X*F(NP+1)-F(NP+2)
      XN=F(1)
      DO 7  I=3,NPP,2
    7 XN=2.0*F(I)+XN

      XN=1./XN
      DO 8  I=1,NPP
      F(I)=XN*F(I)
    8 CONTINUE
      RETURN
      END
```

3.3.7 Subroutine BESNX

Purpose: BESNX computes the Bessel function of the first kind, $J_n(x)$, for integer order, n, and real argument, x. In fact, if real argument, X, and integer order, N, is input, BESNX will compute:

$$J_o(X),\ J_1(X),\ .\ .\ .\ ,\ J_N(X)\ (\text{if } N \geq 0),\ \text{or}$$
$$J_o(X),\ J_{-1}(X),\ .\ .\ .\ ,\ J_N(X)\ (\text{if } N < 0).$$

Method: The step-by-step procedure is as follows:

Step 1: Determine index, NMAX, to start backward recursion from the equations:

$$IX = \max\ (5\ |X|^{1/3},\ 10)$$

$$NMAX = \max\ (|N| + IX + 2,\ |X| + IX + 1)$$

where N is the integer order and X is the real argument. For discussions of the algorithms used, see references 48 through 51.

Step 2: Determine overflow and underflow bounds using:

$$OVER = 2^{1068}\ \frac{|X|}{NMAX} \text{ and } UNDER = 2^{93}\ \frac{|X|}{NMAX}$$

Step 3: Calculate uncorrected $J_K(X)$, K = NMAX, NMAX-1, . . . , 1,0 by backward recursion using:

$$J_{K-1}(X) = \frac{2K}{X}\ J_K(X) - J_{K+1}(X)$$

where $J_{NMAX}(X) = 1$ and $J_{NMAX+1}(X) = 0$. When using this recursion formula, prevent from overflow by using OVER and UNDER.

Step 4: Calculate the correction relation REL from:

$$REL = J_o(X) + 2\sum_{j=1}^{N_o} J_{2j}(X)$$

where N_o = [NMAX/2], the largest integer, which is less than or equal to NMAX/2. When calculating REL, account for the preventive measures (for overflow and underflow) which were taken in step 3.

Step 5: Calculate the corrected $J_K(X)$ for K = 1,2, . . . , N by dividing the uncorrected values by REL.

Step 6: In case the argument X is zero, let $J_o(X) = 1$, $J_1(X) = J_2(X) = . . . = J_{|N|}(X) = 0$.

Step 7: If $N < 0$, then correct for sign using the equation $J_{-K}(X) = (-1)^K J_K(X)$.

Usage:

CALLING SEQUENCE

```
REAL JNX
DIMENSION JNX ( ≥ |N| + 1)
.
.
.
CALL BESNX (N, X, JNX)
```

INPUT

N   integer order of the Bessel function

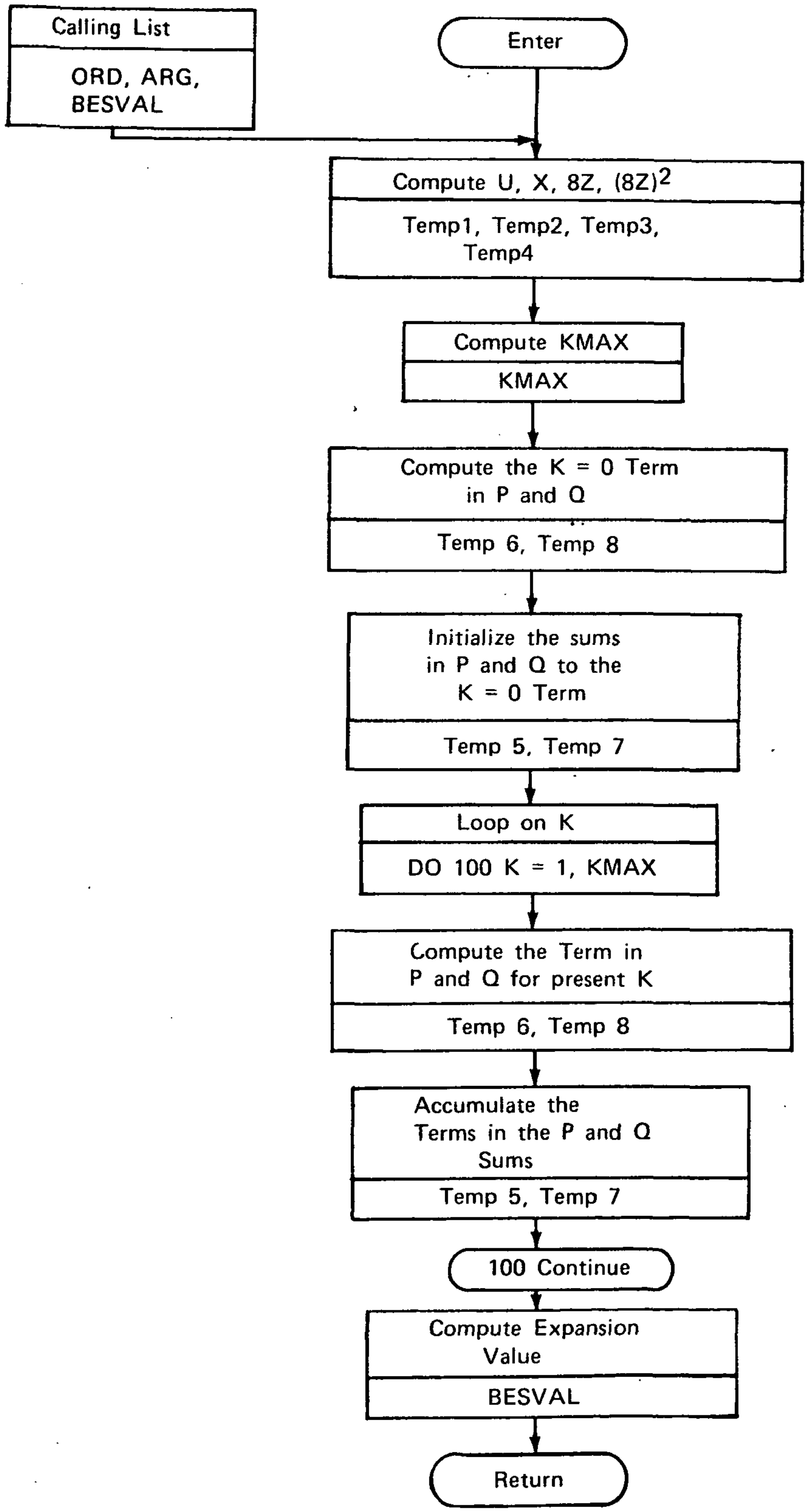
X   real argument of the Bessel function

OUTPUT

JNX array where JNX(1) to JNX(|N| + 1) contains the values of the Bessel function of the first kind for argument X and orders 0 to N, respectively

Accuracy: On the CDC 6600, the least number of significant figures for several ranges of arguments is given below (see refs. 48, 51, and 52).

Range of argument x and index n	Least number of significant figures
x = .001(.001).009 n = 0(1)50	9
x = .01(.01).09 n = 0(1)50	12
x = .1(.1).9 n = 0(1)50	12
x = 1.(1.)9. n = 0(1)50	12
x = 10.(10.)90. n = 0(1)50	5
x = 100.(100.)900. n = 0(1)50	5



Calling List
ORD, ARG, BESVAL
Enter
Compute U, X, 8Z, (8Z)2
Temp1, Temp2, Temp3, Temp4
Compute KMAX
KMAX
Compute the K = 0 Term in P and Q
Temp 6, Temp 8
Initialize the sums in P and Q to the K = 0 Term
Temp 5, Temp 7
Loop on K
DO 100 K = 1, KMAX
Compute the Term in P and Q for present K
Temp 6, Temp 8
Accumulate the Terms in the P and Q Sums
Temp 5, Temp 7
100 Continue
Compute Expansion Value
BESVAL
Return

```
      SUBROUTINE BESNX(N,X,JNX)
C
C PURPOSE       EVALUATE THE BESSEL FUNCTION OF THE FIRST KIND FOR
C               INTEGER ORDER AND REAL ARGUMENT. USING THE RECURRENCE
C               ALGORITHM OF MILLER AN ARRAY OF BESSEL FUNCTION
C               VALUES IS GENERATED.
C INPUT         VARIABLE DEFINITION
C                N          INTEGER ORDER
C                X          REAL ARGUMENT
C OUTPUT         JNX        ARRAY OF LENGTH AT LEAST  /N/ + 1  WHERE
C                           UPON RETURN JNX(1) TO JNX(/N/+1) CONTAIN THE
C                           BESSEL FUNCTION CORRESPONDING TO ARGUMENT X
C                           AND ORDERS 0 TO N, RESPECTIVELY
C
      REAL JN,JNM,JNP,JNX,JX
      DIMENSION JNX(1)
      DATA SML,BIG/00001400030000000000,03776777777777777777/
      IF(X .EQ. 0.)  GO TO 130
C              DETERMINE INDEX TO START BACKWARD RECURSION
      ABSX = ABS(X)
      IABSX = IFIX(ABSX)
      IABSN1 = IABS(N) + 1
      IF(ABSX .LT. 8.)  10,20
 10   IX = 10
      GO TO 30
 20   IX = 5.*(ABSX**.33333333333333)
 30   IF(IABSX .GE. IABSN1)  GO TO 40
      NMAX = IABSN1 + IX + 1
      GO TO 50
 40   NMAX = IABSX + IX + 1
C              DETERMINE OVERFLOW AND UNDERFLOW CONSTANTS
 50   OVER = (BIG/4.) * (ABSX/FLOAT(NMAX))
      UNDER = OVER*SML
C              CALCULATE UNCORRECTED JNX BY BACKWARD RECURSION
C              AND COMPUTE THE CORRECTION RELATION, REL
      JN = 1.
      JNP = 0.
      REL = 0.
      IIABSN1 = IABSN1
      DO 140 I=1,NMAX
      INDEX = NMAX - I + 1
      IF(INDEX .GT. IABSN1)  60,70
 60   NDEX = IABSN1
      GO TO 80
 70   NDEX = INDEX
 80   JNM = (2.*FLOAT(INDEX) / X)*JN - JNP
      JNX(NDEX) = JNM
C              PREVENT FROM OVERFLOW AND UNDERFLOW
      ABSJNM = ABS(JNM)
      IF(ABSJNM .LE. OVER) GO TO 110
      JN = JN / OVER
      JNM = JNM / OVER
```

```
      REL = REL / OVER
      IF(INDEX .GT. IABSN1) GO TO 110
      DO 90 II=INDEX,IIABSN1
      J = II
      JX = JNX(II)
      ABSJNX = ABS(JX)
      IF(ABSJNX .LE. UNDER)  GO TO 100
90    JNX(II) = JNX(II) / OVER
      GO TO 110
100   IIABSN1 = J - 1
110   IF(INDEX .NE. 1) GO TO 120
      REL = REL + JNM
      GO TO 130
120   L = MOD(INDEX,2)
      IF(L .EQ. 0)  GO TO 130
      REL = REL + 2.*JNM
130   JNP = JN
      JN = JNM
                CALCULATE CORRECTED JNX
      SMLREL = SML * REL
      DO 15C I=1,IIABSN1
      MAX = I
      JX = JNX(I)
      ABSJNX = ABS(JX)
      IF(ABSJNX .LE. SMLREL) GO TO 160
      JNX(I) = JNX(I) / REL
      IF(IIABSN1 .EQ. IABSN1) GO TO 200
      MAX = IIABSN1 + 1
      DO 17C I=MAX,IABSN1
      JNX(I) = 0.
      GO TO 200
160   JNX(1) = 1.
      DO 19C I=2,IABSN1
190   JNX(I) = 0.
200   IF(N .GE. 0) RETURN
      DO 210 I=2,IABSN1,2
210   JNX(I) = -JNX(I)
      RETJRN
      END
```

3.3.8 Subroutine BESJLA

Purpose: This subroutine evaluates Hankel's asymptotic expansion for the Bessel function $J_\nu(Z)$, for formulas (9.2.5), (9.2.9), and (9.2.10) of reference 30, where K_{max} is the larger of $\nu/2 + 1$ and 3.

$$J_\nu(Z) = \sqrt{\frac{2}{\pi Z}} \left\{ P(\nu,Z) \cos X - Q(\nu,Z) \sin X \right\}$$

$$P(\nu,Z) \sim \sum_{K=0}^{K_{MAX}} (-1)^K \frac{(\nu,2K)}{(2Z)^{2K}}$$

$$= 1 - \frac{(\mu-1)(\mu-9)}{2!(8Z)^2} + \frac{(\mu-1)(\mu-9)(\mu-25)(\mu-49)}{4!(8Z)^4} - \cdots$$

$$Q(\nu,Z) \sim \sum_{K=0}^{K_{MAX}} (-1)^K \frac{(\nu,2K+1)}{(2Z)^{2K+1}}$$

$$= \frac{\mu-1}{8Z} - \frac{(\mu-1)(\mu-9)(\mu-25)}{3!(8Z)^3} + \cdots$$

$$K_{MAX} = \text{MAX}\left\{ \frac{1}{2}\nu + 1,\ 3 \right\}$$

$$\mu = 4\nu^2$$

$$X = Z - \left(\frac{1}{2}\nu + \frac{1}{4} \right)\pi$$

Method: The procedure is as follows:

1) Evaluate μ, X, 8Z, and $(8Z)^2$.

2) Set K_{max}.

3) Compute the K = 0 term for P and Q and initialize P and Q to that term.

4) For each K, K = 1, . . . , K_{max}, compute P and Q (recursively) by multiplying the previous term by the appropriate factor and accumulate P and Q.

5) Compute Hankel's asymptotic expression.

Usage: CALLING SEQUENCE

```
CALL BESJLA (ORD,ARG,BESVAL)
```

INPUT

ORD	nonnegative order
ARG	real positive argument Z

OUTPUT

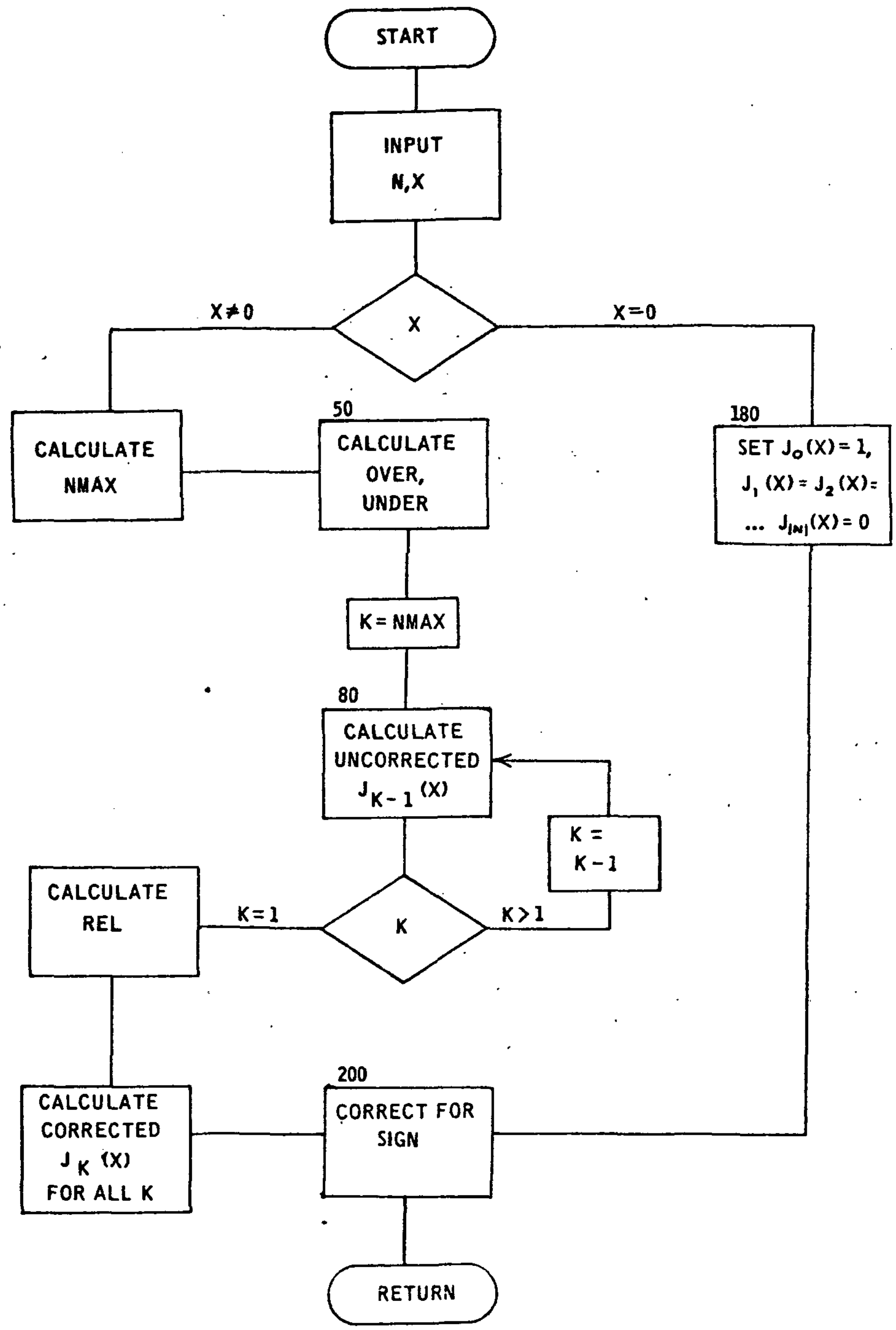
BESVAL value of the expansion

Restrictions: ORD $\geq$ 0
ARG > 0

Timing: The timing is proportional to the number of arithmetic operations which is 8 + 13 K_{max} multiplications, 2 + 2 K_{max} divisions, 2 + 2 K_{max} additions, and 3 + 6 K_{max} subtractions plus time for a call to SQRT.

Accuracy: The accuracy is of the computer type.

Note: The value of BESJLA compares well (five to seven places) with the results of subroutines BSSLS (sec. 3.3.6) and BESNX when the argument is at least $20 * e^{.025} * \text{ORD}$.


START
INPUT N,X
X
X≠0
X=0
CALCULATE NMAX
50
CALCULATE OVER, UNDER
180
SET $J_0(X) = 1$, $J_1(X) = J_2(X) = \dots J_{|N|}(X) = 0$
K = NMAX
80
CALCULATE UNCORRECTED $J_{K-1}(X)$
K = K - 1
CALCULATE REL
K=1
K
K>1
200
CALCULATE CORRECTED $J_K(X)$ FOR ALL K
CORRECT FOR SIGN
RETURN

```
      SUBROUTINE BESJLA(ORD,ARG,BESVAL)
C
C PURPOSE       COMPUTE HANKELS ASYMPTOTIC EXPANSION FOR LARGE
C               ARGUMENT TO THE BESSEL FUNCTION
C
C INPUT         ORD       NON-NEGATIVE ORDER
C               ARG       POSITIVE ARGUMENT
C OUTPUT        BESVAL    EVALUTION OF HANKELS EXPANSION
C
C REFERENCE     HANDBOOK OF MATHEMATICAL FUNCTIONS,
C               EDITED BY M. ABRAMOWITZ AND I. A. STEGUM
C               NATIONAL BUREAU OF STANDARDS APPLIED MATHEMATICS SERIED
C               NUMBER 55, ISSUED 1964,
C               SECTION 9.2, FORMULAS 9.2.5, 9.2.9 AND 9.2.10
C
C               BASIC FORMULA VARIABLES
C
      TEMP1 = 4.*ORD*ORD
      TEMP2 = ARG - (.5*ORD + .25)*3.14159265358979
      TEMP3 = 8.*ARG
      TEMP4 = TEMP3*TEMP3
C
C               COMPUTE MAXIMUM SUM INDEX
C
      KMAX = .5*ORD + 1.
      IF( KMAX.LT.3 ) KMAX=3
C
C               COMPUTE FORMULAS 9.2.9 AND 9.2.10
C
C               INITIALIZE SUM TO K=0 TERM
C
      TEMP5 = 1.
      TEMP6 = TEMP5
      TEMP7 = (TEMP1-1.)/TEMP3
      TEMP8 = TEMP7
C
C               ACCUMULATE THE SUM IN 9.2.9 AND 9.2.10 GENERATING EACH
C               ELEMENT IN THE SUM BY RECURSION
C
      DO 100 K = 1,KMAX
      TEMP9 = 4.*K
      TEMP10=2.*K
      TEMP6 = -TEMP6*(TEMP1-(TEMP9-3.)**2)*(TEMP1-(TEMP9-1.)**2)/
     1             ( TEMP10*(TEMP10-1.)*TEMP4 )
      TEMP5 = TEMP5 + TEMP6
C
      TEMP8 = -TEMP8*(TEMP1-(TEMP9-1.)**2)*(TEMP1-(TEMP9+1.)**2)/
     1             ((TEMP10+1.)*TEMP10*TEMP4 )
      TEMP7 = TEMP7 + TEMP8
  100 CONTINUE
C
C               COMPUTE HANKELS APPROXIMATION
C
C
      BESVAL = SQRT( 2./(3.14159265358979*ARG))*
     1         ( TEMP5*COS(TEMP2) - TEMP7*SIN(TEMP2) )
C
      RETURN
      END
```

3.3.9 Subroutine BESIE

Purpose: This subroutine evaluates approximation formulas for $I_\ell(x)e^{-x}$. The formulas are given in reference 30. For $x < 1$, formula (9.6.7) is used. For $1 \le x \le A$, formula (9.6.52) where $A = \max(20|\ell|,5)$, is used. For $A < x$, formula (9.6) is approximated. Formula (9.6.6), $I_{-\ell}(x) = I_\ell(x)$, is indirectly used.

Method: The procedure is as follows:

1) If ARG < 1, calculate:

$$\text{BESIEX} = \frac{1}{(|L|)!\,2^{|L|}} (\text{ARG})^{|L|} e^{-\text{ARG}}$$

and return.

2) If ARG > max (20|L|,5), to to step 6.

3) Set $\text{NMAX} = \begin{cases} (2.5)(\text{ARG}) + 1 & \text{if ARG} \le 50 \\ (1.25)(\text{ARG}) + 1 & \text{if ARG} > 50 \end{cases}$

4) Calculate $J_{|L|+n}(\text{ARG})$, n = 0,1,..., NMAX.
   a) if $|L| + \text{NMAX} \le 100$, use BSSLS.
   b) if $|L| + \text{NMAX} > 100$ and $\text{ARG} \geq 20\, e^{.025(|L| + \text{NMAX})}$, use BESJLA.
   c) if $|L| + \text{NMAX} > 100$ and $\text{ARG} > 20\, e^{.025(|L| + \text{NMAX})}$, use BESNX.

5) Calculate:

$$\text{BESIEX} = e^{-\text{ARG}} \sum_{n=0}^{N_{MAX}} \frac{(\text{ARG})^n}{n!} J_{|L|+n}(\text{ARG})$$

and return.

6) Calculate:

$$BESIEX = \frac{1}{\sqrt{2\pi}\,(ARG)^{1/2}}\left\{1-\frac{\mu-1}{8(ARG)}+\frac{(\mu-1)(\mu-9)}{2!\,(8\ ARG)^2}-\frac{(\mu-1)(\mu-9)(\mu-25)}{3!\ (8\ ARG)^3}\right\}$$

where $\mu = 4|L|^2$

Usage: CALLING SEQUENCE

```
COMMON/SCRATCH/BES(1000)
.
.
.
CALL BESIE(L,ARG,BESIEX)
```

INPUT

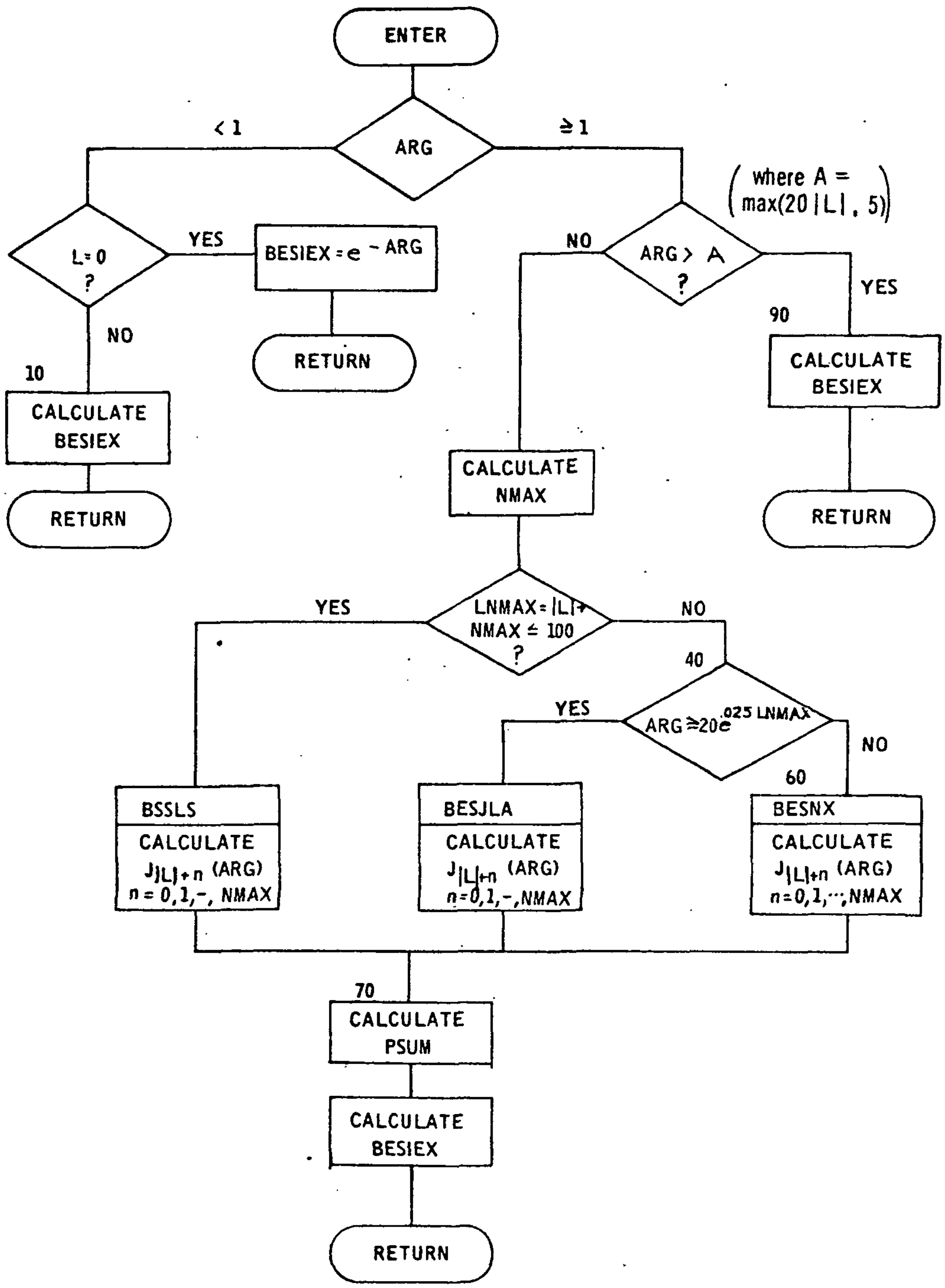
L	the order of the modified Bessel function
ARG	the argument of the function

OUTPUT

BESIEX	the value of the function

Restrictions: If ARG $\leq$ max $(20|L|,5)$, then array BES having dimension 1000 implies $|L| + 1.25(ARG) + 2 \leq 1000$.

Accuracy: The accuracy is of the computer type.


ENTER
ARG
< 1
≥ 1
L = 0 ?
YES
BESIEX = e^{-ARG}
RETURN
NO
10
CALCULATE BESIEX
RETURN
where A = max(20 |L|, 5)
ARG > A ?
NO
YES
90
CALCULATE BESIEX
RETURN
CALCULATE NMAX
LNMAX = |L| + NMAX ≤ 100 ?
YES
NO
40
ARG ≥ 20e^{.025 LNMAX}
YES
NO
60
BSSLS
CALCULATE J|L|+n (ARG) n = 0,1,-, NMAX
BESJLA
CALCULATE J|L|+n (ARG) n = 0,1,-, NMAX
BESNX
CALCULATE J|L|+n (ARG) n = 0,1,···, NMAX
70
CALCULATE PSUM
CALCULATE BESIEX
RETURN

```
      SUBROUTINE BESIE(L,ARG,BESIEX)
      COMMON/SCRATCH/BES(1000)
      DATA SQRT2PI/2.506628274631/
      LABS = IABS(L)
      IF( ARG .GE. 1. )  GO TO 30
      IF( L .NE. 0 )  GO TO 10
      BESIEX = 1.
      BESIEX = BESIEX*EXP(-ARG)
      RETURN
10    BESIEX = ARG/2.
      IF( LABS .EQ. 1 )  BESIEX = BESIEX*EXP(-ARG)
      IF( LABS .EQ. 1 )  RETURN
      DO 20 I=2,LABS
      BESIEX = BESIEX*ARG / (FLOAT(I)*2.)
      BESIEX = BESIEX*EXP(-ARG)
      RETURN
      TEMP1 = FLOAT( MAX0(20*LABS,5) )
      IF( ARG .GT. TEMP1 )    GO TO 90
      IF( ARG .LE. 50. )   NMAX = IFIX(2.5*ARG)  + 1
      IF( ARG .GT. 50. )   NMAX = IFIX(1.25*ARG) + 1
      NMAX = MAX0(NMAX,10)
      LNMAX = LABS + NMAX
      IF( LNMAX .GT. 100 )  GO TO 40
      CALL BSSLS(ARG,BES,LNMAX,IERR)
      GO TO 70
      TEMP1 = 20.*EXP(.025*LNMAX)
      IF( ARG .LT. TEMP1 )  GO TO 60
      DO 50 N=LABS,LNMAX
      ORDER = FLOAT(N)
      CALL BESJLA(ORDER,ARG,BES(N+1))
      GO TO 70
      CALL BESNX(LNMAX,ARG,BES)
      TEMP1 = 1.
      PSUM = BES(LABS + 1)
      DO 80 N=1,NMAX
      TEMP1 = TEMP1*(ARG/FLOAT(N))
      PSUM  = PSUM + TEMP1*BES(LABS+N + 1)
      BESIEX = PSUM*EXP(-ARG)
      RETURN
      TEMP1 = 1. / (SQRT2PI*SQRT(ARG))
      TEMP2 = ( 4.*( FLOAT(LABS)**2 ) - 1. ) / (8.*ARG)
      TEMP3 = TEMP2*( 4.*( FLOAT(LABS)**2 ) - 9. ) / (16.*ARG)
      TEMP4 = TEMP3*( 4.*( FLOAT(LABS)**2 ) - 25. ) / (24.*ARG)
      BESIEX = TEMP1*( 1. - TEMP2 + TEMP3 - TEMP4 )
      RETURN
      END
```

3.3.10 Subroutine BESIK

Purpose: This subroutine evaluates the modified Bessel functions I_0, I_1, K_0, and K_1 for a real argument.

Method: The procedure is as follows:

1) Set the error code and return when the argument is not positive.

2) Compute $I_0(x)$ using formulas (9.8.1) and (9.8.2) of reference 30 when $x \leqq 3.75$ and $x > 3.75$, respectively.

3) Compute $I_1(x)$ using formulas (9.8.3) and (9.8.4) of reference 30 when $x \leqq 3.75$ and > 3.75, respectively.

4) Compute $K_0(x)$ using formulas (9.8.5) and 9.8.6) of reference 30 when $x \leqq 2$ and $x > 2$, respectively.

5) Compute K_1 using formulas (9.8.7) and (9.8.8) of reference 30 when $x \leqq 2$ and $x > 2$, respectively.

Usage: CALLING SEQUENCE

```
CALL BESIK(X,IFCN,BESI0,BESI1,BESK0,BESK1,IERR)
```

INPUT

X positive argument

IFCN = 1 to compute I_0

2 to compute I_1

3 to compute I_0, K_0

4 to compute I_1, K_1

5 to compute I_0, I_1, K_0, K_1

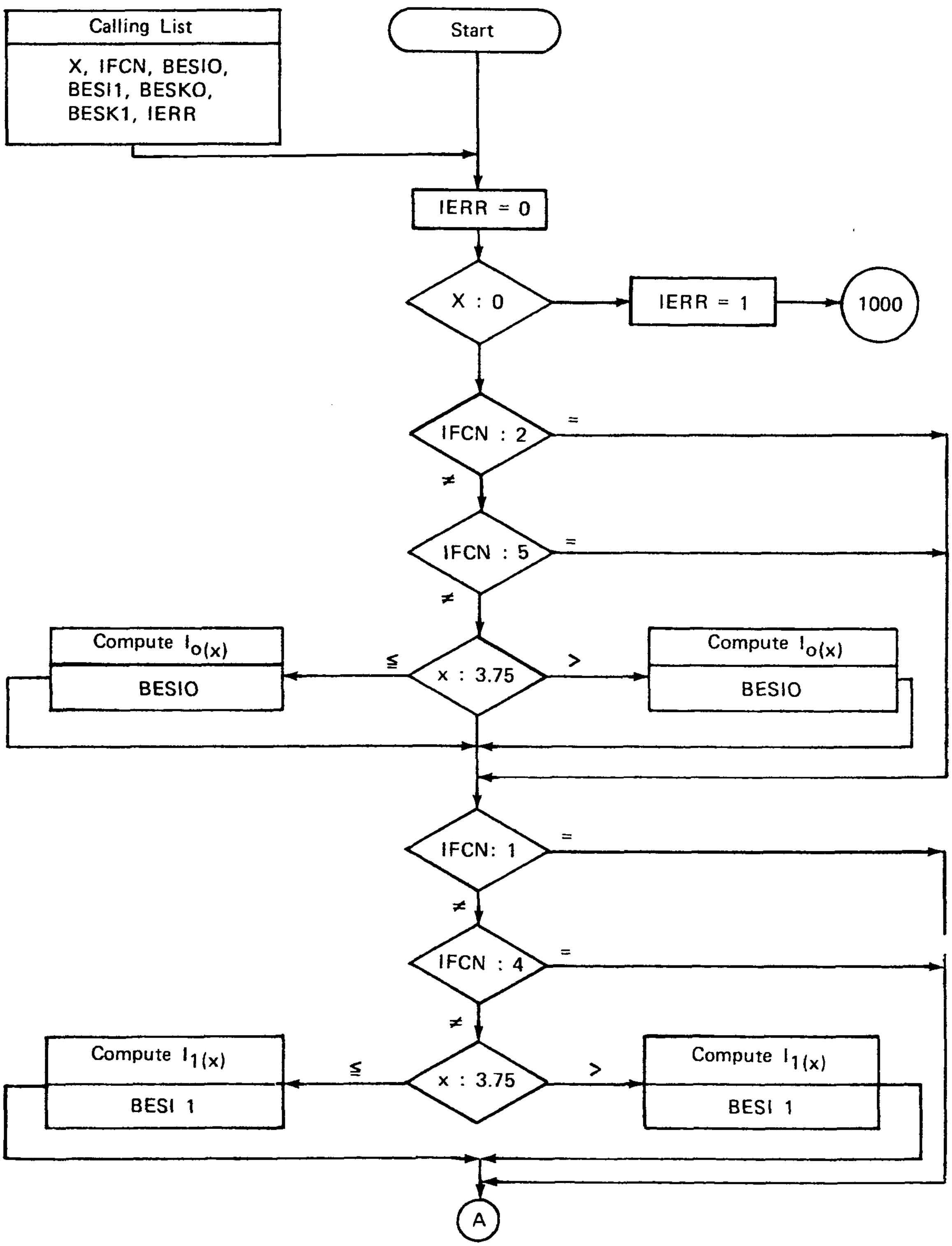
OUTPUT

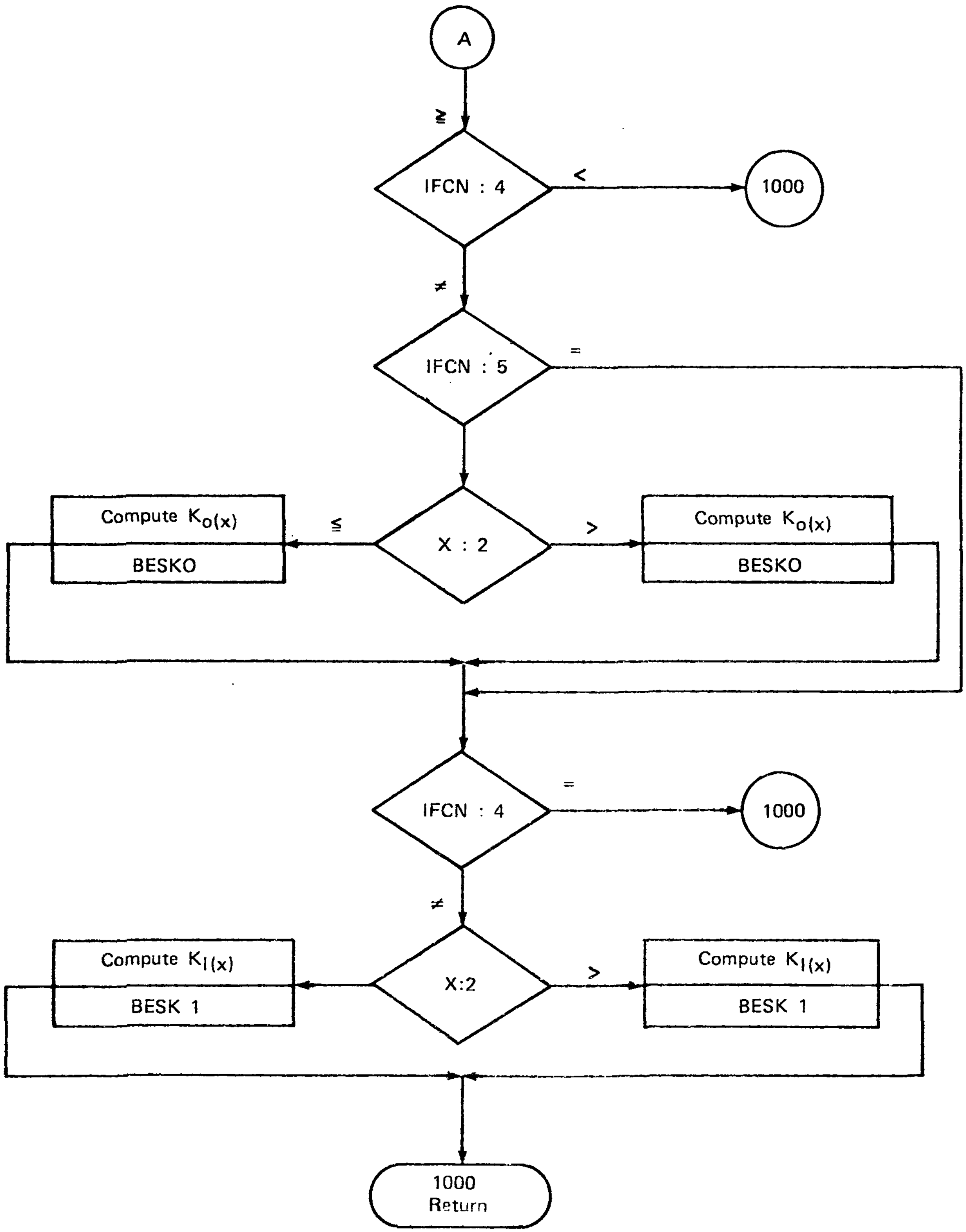
BESIO	I_0	depending on IFCN above
BESI1	I_1	
BESKO	K_0	
BESK1	K_1	
IERR		

Error Return: IERR = 0 unless $x \leqq 0$ and then IERR = 1 and no computations are made.

Timing: The timing is approximately equal to twice the time for 7 additions and 10 multiplies.

Accuracy: The accuracy is of the algorithmic type and, in particular, according to reference 30, the maximum error in using the above polynomial approximations is less than 2.2×10^{-7}.


Calling List
X, IFCN, BESIO,
BESI1, BESKO,
BESK1, IERR
Start
IERR = 0
X : 0
IERR = 1
1000
IFCN : 2
=
≠
IFCN : 5
=
≠
Compute $I_0(x)$
BESIO
≤
x : 3.75
>
Compute $I_0(x)$
BESIO
IFCN: 1
=
≠
IFCN : 4
=
≠
Compute $I_1(x)$
BESI 1
≤
x : 3.75
>
Compute $I_1(x)$
BESI 1
A


A
≧
IFCN : 4
<
1000
≠
IFCN : 5
=
Compute $K_{o(x)}$
BESKO
≦
X : 2
>
Compute $K_{o(x)}$
BESKO
IFCN : 4
=
1000
≠
Compute $K_{I(x)}$
BESK 1
X:2
>
Compute $K_{I(x)}$
BESK 1
1000
Return

```
      SUBROUTINE BESIK(X,IFCN,BESI0,BESI1,BESK0,BESK1,IERR)
C
C PURPOSE        COMPUTE THE MODIFIED BESSEL FUNCTIONS I AND K OF ORDERS
C                0 AND 1 USING POLYNOMIAL APPROXIMATIONS FROM REF.
C
C REFERENCE      M. ABRAMOWITZ AND I. A. STEGUM, HANBOOK OF MATHEMATICAL
C                FUNCTIONS, NATIONAL BUREAU OF STANDARDS APPLIED
C                MATHEMATICS SERIES 55
C
C
C
C                CHECK THE ARGUMENT
C
      IERR = 0
      IF( X ) 10,10,20
   10 IERR = 1
      GO TO 1000
   20 IF( IFCN.EQ.2 ) GO TO 100
      IF( IFCN.EQ.5 ) GO TO 100
      IF( X.GT. 3.75 ) GO TO 50
C
C                COMPUTE I FOR ORDER 0 AND ARGUMENT AT MOST 3.75
C                USING FORMULA 9.8.1 OF REF.
C
      T = X/3.75
      T2 = T*T
      BESI0 = 1.+T2*(3.5156229+T2*(3.0899424+T2*(1.2067492+T2*(.2659732
     1           +T2*(.0360768 +T2* .0045813)))))
      GO TO 100
C
C                COMPUTE I FOR ORDER 0 AND ARGUMENT AT LEAST 3.75
C                USING FORMULA 9.8.2 OF REF.
C
   50 T1 = 3.75/X
      BESI0 = .39894228 + T1*(.01328592 +T1*(.00225319+T1*(-.00157565
     1+T1*(.00916281+T1*(-.02057706+T1*(.02635537 +T1*(-.01647633
     2+T1* .00392377)))))))
      BESI0 = BESI0*EXP(X)/SQRT(X)
C
  100 IF( IFCN.EQ.1 ) GO TO 200
      IF( IFCN.EQ.4 ) GO TO 200
      IF( X.GT.3.75 ) GO TO 150
C
C                COMPUTE I FOR ORDER 1 AND ARGUMENT  AT MOST 3.75
C                USING FORMULA 9.8.3 OF REF.
C
      T = X/3.75
      T2 = T*T
      BESI1 = .5 +T2*(.87890594 +T2*(.51498869 +T2*(.15084934
     1             +T2*(.02658733 +T2*(.00301532 +T2*.00032411)))))
      BESI1 = BESI1*X
```

```
      GO TO 200
C
C               COMPUTE I FOR ORDER 1 AND ARGUMENT AT LEAST 3.75
C               USING FORMULA 9.8.4 OF REF
C
  150 T1 =3.75/X
      BESI1 = .39894228 +T1*(-.03988024 +T1*(-.00362018 +T1*(.00163801
     1   +T1*(-.01031555+T1*( .02282967 +T1*(-.02895312 +T1*(.01787654
     2   +T1*(-.00420059))))))))
      BESI1 = BESI1*EXP(X)/SQRT(X)
C
  200 IF( IFCN.LT.4) GO TO 1000
      IF( IFCN.EQ.5) GO TO 300
      IF( X.GT.2. ) GO TO 250
C
C               COMPUTE K FOR ORDER 1 AND ARGUMENT AT MOST 2.
C               USING REF. FORMULA 9.8.5
C
      X2 = .25*X*X
      BESK0 =-.57721566 + X2*(.42278420+X2*(.23069756 +X2*(.03488590
     1    +X2*(.00262698+ X2*(.00010750+X2* .00000740 )))))
      BESK0 = -ALOG(X*.5)*BESI0 + BESK0
      GO TO 300
C
C               COMPUTE K FOR ORDER 1 AND ARGUMENT AT LEAST 2.
C               USING
C
  250 X2 = 2./X
      BESK0 = 1.25331414 + X2*(-.07832358 +X2*(.02189568+X2*(-.01062446
     1   +X2*(.00587872  + X2*(-.00251540+X2*.00053208)))))
      BESK0 = BESK0*EXP(-X)/SQRT(X)
C
  300 IF(IFCN.EQ.4) GO TO 1000
      IF( X.GT. 2.) GO TO 350
C
C               COMPUTE K FOR ORDER 1 AND ARGUMENT AT MOST 2
C               USING REF FORMULA 9.8.7
C
      X2 = .25*X*X
      BESK1 = 1.+X2*(.15443144 +X2*(-.67278579 +X2*(-.18156897
     1          +X2*(-.01919402+X2*(-.00110404 +X2*(-.00004686))))))
      BESK1 = ALOG(X*.5)*BESI1 + BESK1/X
      GO TO 1000
C
C               COMPUTE K FOR ORDER 1 AND ARGUMENT AT LEAST 2.
C               USING REF. FORMULA 9.8.8
C
  350 X2 = 2./X
      BESK1 = 1.25331414+X2*(.23498619+X2*(-.03655620+X2*(.01504268
     1   +X2*(-.00780353+X2*(.00325614+X2*  (-.00068245))))))
      BESK1 = BESK1*EXP(-X)/SQRT(X)
C
 1000 RETURN
      END
```

3.3.11 Subroutine ROCABES

Purpose: ROCABES computes the Bessel functions of the first and second kinds for real order and complex argument.

Discussion: This subroutine returns a table of $|N| + 1$ values of these Bessel functions of the first and second kinds for complex arguments and real orders where N is a user-assigned parameter. ROCABES is a modification of subroutine NYU BES4 (see ref. 53), including a change from complex order to real order.

Method: The method is the same as that of reference 8 but modified for real order. Let the Bessel functions of the first and second kinds be $J_W(z)$ and $Y_W(z)$ where the argument is $z = x + i\,y$, and the order W is real. For $W > 0$, define $N = [W]$ (the greatest integer less than or equal to W), $\alpha = W - N$, and the orders

$$W = \alpha + n \quad , \quad n = 0,1,\ldots, N \ ,$$

and for $W < 0$, define $N = [W]+1$, $\alpha = W - N$, and the orders

$$W = \alpha + n \quad , \quad n = 0,-1,\ldots, -|N| .$$

The Bessel functions $J_W(z)$ and $Y_W(z)$ are computed for all orders as defined above.

The results are stored in the following arrays: BJRE contains the real part of $J_W(z)$; BJIM contains the imaginary part of $J_W(z)$; YRE contains the real part of $Y_W(z)$; and YIM contains the imaginary part of $Y_W(z)$ as follows:

	$N \geq 0$	$N < 0$
BJRE(1)	Re $J_{(\alpha+0)}(z)$	Re $J_{(\alpha+0)}(z)$
BJRE(2)	Re $J_{(\alpha+1)}(z)$	Re $J_{(\alpha+1)}(z)$
.	.	.
.	.	.
.	.	.
BJRE(N+1)	Re $J_{(\alpha+N)}(z)$	Re $J_{(\alpha+\|N\|)}(z)$

and similarly for the arrays BJIM, YRE, and YIM.

Usage: CALLING SEQUENCE

```
DIMENSION BJRE(K),BJIM(K),YRE(|N|+1),YIM(|N|+1)
   where:   K = max (|z| + 25, |N| + 15)
CALL ROCABES (X,Y,ALPHA,N,BJRE,BJIM,YRE,YIM)
```

INPUT

X	the real part of the argument z
Y	the imaginary part of the argument z
ALPHA	the fractional part of the real part of the order W
N	the integral part of the real order W and $\|N\| + 1$ is the number of values computed

OUTPUT

BJRE, BJIM, YRE, YIM — as defined above

SUBPROGRAMS CALLED

ALGAMF; see reference 54

Note that ROCABES uses the following subroutines of its own: MBEGIN, MJRECUR, MJSUM, MFACTOR, MCOMLOG, MCOMEXP, MJNORM, MYSUM, MYGNU, MYZERO, MWRONSK, MNEGN, MYRECUR, MYGNUP, MYSUMP

Storage: 2455 octal, which includes all subroutines listed above under SUBPROGRAMS CALLED except ALGAMF

```
      SUBROUTINEROCABES(X,Y,ALPHA,N,BJRE,BJIM,YRE,YIM)
C
      DIMENSION BJRE(1),BJIM(1),YRE(1),YIM(1)
      CALLMBEGIN(X,Y,N,K,R)                                           BES40030
      CALLMJRECUR(X,Y,ALPHA,K,R,BJRE,BJIM)
      CALLMJSUM(ALPHA,K,BJRE,BJIM,SUMRA,SUMIA)
      CALLMFACTOR(X,Y,ALPHA,Q,R)
      CALLMJNORM(K,Q,R,SUMRA,SUMIA,BJRE,BJIM)                         BES40070
    7 CALLMYSUM(X,Y,ALPHA,K,BJRE,BJIM,ASUMR,ASUMI)
    8 CALLMYGNU(X,Y,ALPHA,Q,R,ASUMR,ASUMI,BJRE,BJIM,YRE,YIM)
 9    CALLMWRONSK (X,Y,BJRE,BJIM,YRE,YIM)                             BES40100
       BJSQ=BJRE(1)**2+BJIM(1)**2                                     BES40110
      IF(BJSQ-.5E-14) 14,14,15
   14 CALLMYSUMP(X,Y,ALPHA,K,BJRE,BJIM,ASUMR,ASUMI)
      CALLMYGNUP(X,Y,ALPHA,Q,R,ASUMR,ASUMI,BJRE,BJIM,YRE,YIM)
 15   IF (N-1)10,12,11                                                BES40150
 10   IF (N)13,12,12                                                  BES40160
   13 CALLMNEGN(X,Y,ALPHA,N,BJRE,BJIM,YRE,YIM)
      GO TO 12                                                        BES40180
   11 CALLMYRECUR(X,Y,N,BJRE,BJIM,YRE,YIM)                            BES40190
 12   RETURN                                                          BES40200
       END                                                            BES40210


      SUBROUTINEMBEGIN(X,Y,N,K,R)
      SSQ=X*X+Y*Y
      KTEN=SQRT(SSQ)+20.0
      NTEN=IABS(N)+10
      M=MAXO(KTEN,NTEN)/2
      K=2*M+1                                                         BES40280
      R=K+1
      RETURN                                                          BES40300
       END

      SUBROUTINEMJRECUR(X,Y,ALPHA,K,R,BJRE,BJIM)
      DIMENSION BJRE(1),BJIM(1)
      RALPHA=R+ALPHA                                                  BES40350
      SSQ=X*X+Y*Y
      BJRE(K+2)=0.                                                    BES40370
      BJIM(K+2)=0.                                                    BES4 380
      BJRE(K+1)=1.0E-37                                               BES40390
      BJIM(K+1)=0.0                                                   BES40400
      TX=2.*X/SSQ
      TY=2.*Y/SSQ
      DO4I=1,K                                                        BES40410
      L1=K+1-I                                                        BES40420
      RALPHA=RALPHA-1.0                                               BES40430
      A=RALPHA*TX
      B=-RALPHA*TY
```

```
      BJRE(L1)=(A*BJRE(L1+1))-(B*BJIM(L1+1))-BJRE(L1+2)                BES40460
    4 BJIM(L1)=(B*BJRE(L1+1))+(A*BJIM(L1+1))-BJIM(L1+2)                BES40470
      RETURN                                                            BES40480
       END


      SUBROUTINEMJSUM(ALPHA,K,BJRE,BJIM,SUMRA,SUMIA)
      DIMENSION BJRE(1),BJIM(1)
  801 SUMRA=BJRE(3)*(ALPHA+2.0)
      SUMIA=(ALPHA+2.0)*BJIM(3)
      GRE=1.0                                                           BES40550
      GIM=0.                                                            BES40560
      S=1.0                                                             BES40570
      DO6I=5,K,2                                                        BES40580
      S=S+1.0                                                           BES40590
      GREN=GRE*(ALPHA+S-1.0)/S
      GIM=GIM*(ALPHA+S-1.0)/S
      GRE=GREN                                                          BES40620
      ALPTS=ALPHA+2.0*S                                                 BES40630
      GJR=GRE*BJRE(I)                                                   BES40640
      GJI=GIM*BJIM(I)                                                   BES40650
      GJRI=GRE*BJIM(I)                                                  BES40660
      GJIR=GIM*BJRE(I)                                                  BES40670
      SUMRB=ALPTS*(GJR-GJI)+SUMRA
      SUMIB=ALPTS*(GJIR+GJRI)+SUMIA
C
C     THE FOLLOWING STATEMENT IS ADDED TO COMPENSATE THE DEFFICIENCY
C     FOUND IN THE PURE IMAGINARY CASE
C
      IF(SUMRA) 19,21,19
   19 IF(ABS((SUMRB/SUMRA)-1.0)-.5E-14) 21,21,10
   21 IF(SUMIA)20,11,20                                                 BES40710
   20 IF(ABS((SUMIB/SUMIA)-1.0)-.5E-14) 11,11,10
   10 SUMRA=SUMRB                                                       BES40730
    6 SUMIA=SUMIB                                                       BES40740
   11 RETURN                                                            BES40750
       END                                                              BES40760


      SUBROUTINEMFACTOR(X,Y,ALPHA,Q,R)
      CALL ALGAMF(ALPHA+1.0,0.,U,V)
      CALLMCOMLOG(X,Y,A1,B1)                                            BES40800
      A2=ALPHA*A1
      B2=ALPHA*B1
      A2=-A2                                                            BES40830
      B2=-B2                                                            BES40840
      CALLMCOMEXP(A2,B2,A3,B3)                                          BES40850
      A4=.6931471806*ALPHA                                              BES40860
      CALLMCOMEXP(A4,0.,A5,B5)
      A6=A3*A5-B3*B5                                                    BES40890
      B6=B3*A5+A3*B5                                                    BES40900
      CALLMCOMEXP(U,V,A7,B7)                                            BES40910
```

```
      Q=A5*A7-B6*B7                                                     BES40920
      R=B5*A7+A6*B7                                                     BES40930
      RETJRN                                                            BES40940
       END                                                              BES40950

      SUBROUTINEMCOMLOG(X,Y,A,B)                                        BES40980
      PI=3.141592654                                                    BES40990
      A=.5*ALOG(X*X+Y*Y)
      IF(X)5,1,4                                                        BES41010
1     B=.5*PI                                                           BES41020
      IF(Y)2,3,8                                                        BES41030
2     B=-3                                                              BES41040
      GO TO 8                                                           BES41050
3     B=0.                                                              BES41060
      GO TO 8                                                           BES41070
    4 B=ATAN(Y/X)
      GO TO 8                                                           BES41090
    5 B=ATAN(Y/X)
      IF(Y)6,7,7                                                        BES41110
6     B=B-PI                                                            BES41120
      GO TO 8                                                           BES41130
7     B=B+PI                                                            BES41140
8     RETJRN                                                            BES41150
       END                                                              BES41160

      SUBROUTINEMCOMEXP(X,Y,A,B)                                        BES41180
      C=EXP(X)
      A=C*COS(Y)
      B=C*SIN(Y)
      RETJRN                                                            BES41220
       END                                                              BES41230

      SUBROUTINEMJNORM(K,Q,R,SUMRA,SUMIA,BJRE,BJIM)                     BES41250
      DIMENSION BJRE(1),BJIM(1)
      S=((SUMRA+BJRE(1))*Q)-((SUMIA+BJIM(1))*R)                         BES41270
      T=((SUMIA+BJIM(1))*Q)+((SUMRA+BJRE(1))*R)                         BES41280
      IF(ABS(S)-ABS(T)) 100,101,101
101   TS=T/S                                                            BES41300
      TSSQ=S*(1.+(TS*TS))
12    DO13I=1,K                                                         BES41320
      BJREN=(BJRE(I)+BJIM(I)*TS)/TSSQ                                   BES41330
      BJIM(I)=(BJIM(I)-BJRE(I)*TS)/TSSQ                                 BES41340
13    BJRE(I)=BJREN                                                     BES41350
      GO  TO  14                                                        BES41360
100   ST=S/T                                                            BES41370
      STSQ=T*((ST*ST)+1.)
102   DO103I=1,K                                                        BES41390
      BJREN=(BJRE(I)*ST+BJIM(I))/STSQ                                   BES41400
      BJIM(I)=(BJIM(I)*ST-BJRE(I))/STSQ                                 BES41410
```

```
103   BJRE(I)=BJREN
  14  RETURN
      END

      SUBROUTINE MYSUM(X,Y,ALPHA,K,BJRE,BJIM,ASUMR,ASUMI)
      DIMENSION BJRE(1),BJIM(1)
      A1=ALPHA-1.0
      A2=A1-1.0
      A3=A1+ALPHA
      GAMRE=-(2.0+ALPHA)/A1
      GAMIM=0.
      ASUMR=GAMRE*BJRE(3)
      ASUMI=GAMRE*BJIM(3)
      T=1.0
      DO 500 I=5,K,2
      T=T+1.0
      B1=2.0*T
      F1=B1+ALPHA
      F2=A3+T
      F3=A1+T
      F5=T-ALPHA
      F6=A2+B1
      G1=F1*F2
      H1=G1*F3
      P1=F5*F6
      CRE=H1/(P1*T)
      TEMP=-CRE*GAMRE
      GAMIM=-CRE*GAMIM
      GAMRE=TEMP
      BSUMR=GAMRE*BJRE(I)-GAMIM*BJIM(I)+ASUMR
      BSUMI=GAMIM*BJRE(I)+GAMRE*BJIM(I)+ASUMI
      IF(ABS((BSUMR/ASUMR)-1.0)-.5E-14) 521,521,510
521   IF(ASUMI)520,511,520
 520  IF(ABS((BSUMI/ASUMI)-1.0)-.5E-14) 511,511,510
510   ASUMR=BSUMR
500   ASUMI=BSUMI
511   RETURN
      END

      SUBROUTINE MYGNU(X,Y,ALPHA,Q,R,ASUMR,ASUMI,BJRE,BJIM,YRE,YIM)
      DIMENSION BJRE(1),BJIM(1),YRE(1),YIM(1)
      PI=3.141592654
      TPI=2.0/PI
      QRE=TPI*(Q*Q-R*R)
      QIM=TPI*2.0*Q*R
      DRE=QRE*ASUMR-QIM*ASUMI
      DIM=QIM*ASUMR+QRE*ASUMI
      IF(ALPHA) 1,3,1
3     CALL MYZERO(X,Y,ALPRE,ALPIM)
      GO TO 720
```

```
1     PALPHA=PI*ALPHA                                              BES42010
      COX=COS(PALPHA)
      SIX=SIN(PALPHA)
      ERE=COX/SIX
      ABSQ3=2.0*ALPHA*ALPHA
      ALPRE=ERE-(QRE*ALPHA/ABSQ3)
      ALPIM=-QIM*ALPHA/ABSQ3
720   YRE(1)=ALPRE*BJRE(1)-ALPIM*BJIM(1)+DRE                       BES42140
      YIM(1)=ALPIM*BJRE(1)+ALPRE*BJIM(1)+DIM                       BES42150
      RETURN                                                       BES42160
       END                                                         BES42170

      SUBROUTINEMYZERO(X,Y,ALPRE,ALPIM)                            BES42190
      TPI=2.0/3.141592654                                          BES42200
      CALLMCOMLOG(X,Y,A,B)                                         BES42210
      ALPRE=TPI*(-.1159315157+A)                                   BES42220
      ALPIM=TPI*B                                                  BES42230
      RETURN                                                       BES42240
       END                                                         BES42250

      SUBROUTINEMWRONSK(X,Y,BJRE,BJIM,YRE,YIM)                     BES42270
      DIMENSION BJRE(1),BJIM(1),YRE(1),YIM(1)
      SSQ=X*X+Y*Y
      TPI=2.0/3.141592654                                          BES42300
      AZRE=TPI*X/SSQ                                               BES42310
      AZIM=-TPI*Y/SSQ                                              BES42320
      ZRE=BJRE(2)*YRE(1)-BJIM(2)*YIM(1)                            BES42330
      ZIM=BJIM(2)*YRE(1)+BJRE(2)*YIM(1)                            BES42340
      BZRE=ZRE-AZRE                                                BES42350
      BZIM=ZIM-AZIM                                                BES42360
      BJSQ=BJRE(1)*BJRE(1)+BJIM(1)*BJIM(1)
      CZRE=BJRE(1)/BJSQ                                            BES42380
      CZIM=-BJIM(1)/BJSQ
      YRE(2)=BZRE*CZRE-BZIM*CZIM                                   BES42400
      YIM(2)=BZIM*CZRE+BZRE*CZIM                                   BES42410
      RETURN                                                       BES42420
       END                                                         BES42430

      SUBROUTINEMNEGN(X,Y,ALPHA,N,BJRE,BJIM,YRE,YIM)
      DIMENSION BJRE(1),BJIM(1),YRE(1),YIM(1)
      L=IABS(N)+1
      SSQ=X*X+Y*Y
      TX=2.*X/SSQ
      TY=2.*Y/SSQ
      RALPHA=ALPHA                                                 BES42510
      A=RALPHA*TX
      B=-RALPHA*TY
      BJRE(2)=A*BJRE(1)-B*BJIM(1)-BJRE(2)                          BES42540
      BJIM(2)=B*BJRE(1)+A*BJIM(1)-BJIM(2)                          BES42550
```

```
      YRE(2)=A*YRE(1)-B*YIM(1)-YRE(2)
      YIM(2)=B*YRE(1)+A*YIM(1)-YIM(2)
      DO 1 I=3,L
      RALPHA=RALPHA-1.0
      A=RALPHA*TX
      B=-RALPHA*TY
      BJRE(I)=A*BJRE(I-1)-B*BJIM(I-1)-BJRE(I-2)
      BJIM(I)=B*BJRE(I-1)+A*BJIM(I-1)-BJIM(I-2)
      YRE(I)=A*YRE(I-1)-B*YIM(I-1)-YRE(I-2)
1     YIM(I)=B*YRE(I-1)+A*YIM(I-1)-YIM(I-2)
      RETURN
       END

      SUBROUTINEMYRECUR(X,Y,N,BJRE,BJIM,YRE,YIM)
      DIMENSION BJRE(1),BJIM(1),YRE(1),YIM(1)
      SSQ=X*X+Y*Y
      TPI=2.0/3.141592654
      AZRE=TPI*X/SSQ
      AZIM=-TPI*Y/SSQ
      L=IABS(N)+1
      DO 1 I=3,L
      ZRE=BJRE(I)*YRE(I-1)-BJIM(I)*YIM(I-1)
      ZIM=BJIM(I)*YRE(I-1)+BJRE(I)*YIM(I-1)
      BZRE=ZRE-AZRE
      BZIM=ZIM-AZIM
      BJSQ=BJRE(I-1)*BJRE(I-1)+BJIM(I-1)*BJIM(I-1)
      CZRE=BJRE(I-1)/BJSQ
      CZIM=-BJIM(I-1)/BJSQ
      YRE(I)=BZRE*CZRE-BZIM*CZIM
    1 YIM(I)=BZIM*CZRE+BZRE*CZIM
      RETURN
       END

      SUBROUTINEMYGNUP(X,Y,ALPHA,Q,R,ASUMR,ASUMI,BJRE,BJIM,YRE,YIM)
      DIMENSION BJRE(1),BJIM(1),YRE(1),YIM(1)
      PI=3.141592654
      TPI=2.0/PI
      QRE=TPI*(Q*Q-R*R)
      QIM=TPI*2.0*Q*R
      DRE=QRE*ASUMR-QIM*ASUMI
      DIM=QIM*ASUMR+QRE*ASUMI
      IF(ALPHA) 1,3,1
3     CALLMYZERO(X,Y,ALPRE,ALPIM)
      GO TO 720
1     PALPHA=PI*ALPHA
      COX=COS(PALPHA)
      SIX=SIN(PALPHA)
      ERE=COX/SIX
      ABSQ3=2.0*ALPHA*ALPHA
      ALPRE=ERE-(QRE*ALPHA/ABSQ3)
```

```
      ALPIM=-QIM*ALPHA/ABSQ3
720   TRE=ALPRE*BJRE(2)-ALPIM*BJIM(2)+DRE                               BES43140
      TIM=ALPIM*BJRE(2)+ALPRE*BJIM(2)+DIM                               BES43150
      SSQ=X*X+Y*Y
      ALPRE=-(Q*X+R*Y)/SSQ
      ALPIM=-(X*R-Q*Y)/SSQ
      YRE(2)=ALPRE*BJRE(1)-ALPIM*BJIM(1)+TRE                            BES43180
      YIM(2)=ALPIM*BJRE(1)+ALPRE*BJIM(1)+TIM                            BES43190
      RETURN                                                            BES43200
       END                                                              BES43210

      SUBROUTINEMYSUMP(X,Y,ALPHA,K,BJRE,BJIM,ASUMR,ASUMI)
      DIMENSION BJRE(1),BJIM(1)
      A1=ALPHA-1.0                                                      BES43250
      A2=A1-1.0                                                         BES43260
      A3=A1+ALPHA                                                       BES43270
      ABSQ=A1*A1
      ROLDRE=-A1*(2.0*ALPHA)/ABSQ
      ROLDIM=0.
      RES1=-ROLDRE/2.0                                                  BES43330
      VMS1=0.
      SSQ=X*X+Y*Y
      STORE=3.*ALPHA*X/SSQ
      STOIM=-3.*ALPHA*Y/SSQ
      RES2=(ROLDRE*STORE-ROLDIM*STOIM)                                  BES43370
      VMS2=(ROLDRE*STOIM+ROLDIM*STORE)                                  BES43380
      ASUMR=RES1*BJRE(2)
      ASUMR=ASUMR+RES2*BJRE(3)-VMS2*BJIM(3)                             BES43400
      ASUMI=RES1*BJIM(2)
      ASUMI=ASUMI+VMS2*BJRE(3)+RES2*BJIM(3)                             BES43420
      T=1.0                                                             BES43430
      DO 500 I=3,K,2                                                    BES43440
      T=T+1.0                                                           BES43450
      B1=2.C*T                                                          BES43460
      F1=B1+ALPHA                                                       BES43470
      F2=A3+T                                                           BES43480
      F3=A1+T                                                           BES43490
      F5=T-ALPHA                                                        BES43500
      F6=A2+B1                                                          BES43510
      G1=F1*F2
      H1=G1*F3
      P1=F5*F6
      CRE=H1/(P1*T)
      TEMP=-CRE*ROLDRE
      RNEWIM=-CRE*ROLDIM
      RNEWRE=TEMP                                                       BES43630
      RES1=(ROLDRE-RNEWRE)/2.0                                          BES43640
      VMS1=(ROLDIM-RNEWIM)/2.0                                          BES43650
      RES2=(RNEWRE*STORE-RNEWIM*STOIM)                                  BES43660
      VMS2=(RNEWRE*STOIM+RNEWIM*STORE)                                  BES43670
      BSUMR=RES1*BJRE(I+1)-VMS1*BJIM(I+1)+ASUMR                         BES43680
```

```
      BSUMI=VMS1*BJRE(I+1)+RES1*BJIM(I+1)+ASUMI                          BES43690
      BSUMR=RES2*BJRE(I+2)-VMS2*BJIM(I+2)+BSUMR                          BES43700
      BSUMI=VMS2*BJRE(I+2)+RES2*BJIM(I+2)+BSUMI                          BES43710
      IF(ABS((BSUMR/ASUMR)-1.0)-.5E-14) 521,521,510
521   IF(ASUMI)520,511,520                                               BES43730
 520  IF(ABS((BSUMI/ASUMI)-1.0)-.5E-14) 511,511,510
510   ASUMR=BSUMR                                                        BES43750
      ASUMI=BSUMI                                                        BES43760
      ROLDIM=RNEWIM                                                      BES43770
500   ROLDRE=RNEWRE                                                      BES43780
511   RETURN                                                             BES43790
       END                                                               BES43800
```

3.3.12 Subroutine SICI

Purpose: This subroutine evaluates the sine and cosine integrals

$$Si(X) = \int_{\infty}^{X} \frac{SIN(t)}{t}\,dt \quad , X \geqq 0$$

$$Ci(X) = \int_{\infty}^{X} \frac{COS(t)}{t}\,dt \quad , X > 0$$

as taken from reference 55.

```
      SUBROUTINE SICI(SI,CI,X)
C
C     ..................................................................
C
C  PURPOSE
C     COMPUTES THE SINE AND COSINE INTEGRAL
C
C  USAGE
C     CALL SICI(SI,CI,X)
C
C  DESCRIPTION OF PARAMETERS
C     SI     - THE RESULTANT VALUE SI(X)
C     CI     - THE RESULTANT VALUE CI(X)
C     X      - THE ARGUMENT OF SI(X) AND CI(X)
C
C  REMARKS
C     THE ARGUMENT VALUE REMAINS UNCHANGED
C
C  SUBROUTINES AND FUNCTION SUBPROGRAMS CALLED
C     NONE
C
C  METHOD
C     DEFINITION
C     SI(X)=INTEGRAL(SIN(T)/T)
C     CI(X)=INTEGRAL(COS(T)/T)
C     EVALUATION
C     REDUCTION OF RANGE USING SYMMETRY
C     DIFFERENT APPROXIMATIONS ARE USED FOR ABS(X) GREATER
C     THAN 4 AND FOR ABS(X) LESS THAN 4.
C     REFERENCE
C     LUKE AND WIMP, *POLYNOMIAL APPROXIMATIONS TO INTEGRAL
C     TRANSFORMS*, MATHEMATICAL TABLES AND OTHER AIDS TO
C     COMPUTATIONS, VOL. 15, 1961, ISSURE 74, PP. 174-178.
C
C     ..................................................................
C
C
C     TEST ARGUMENT RANGE
C
      Z=ABS(X)
      IF(Z-4.)1,1,4
    1 Y=(4.-Z)*(4.+Z)
      SI=-1.570796326
      IF(Z)3,2,3
    2 CI=-1.E75
      RETURN
    3 SI=X*(((((1.753141E-9*Y+1.568988E-7)*Y+1.374168E-5)*Y+6.939889E-4)
     1*Y+1.964882E-2)*Y+4.395509E-1+SI/X)
      CI=((5.772156E-1+ALOG(Z))/Z-Z*(((((1.386985E-10*Y+1.584996E-8)*Y
     1+1.725752E-6)*Y+1.185999E-4)*Y+4.990920E-3)*Y+1.315308E-1))*Z
      RETURN
    4 SI=SIN(Z)
      Y=COS(Z)
```

```
      Z=4./Z
      U=((((((((4.048069E-3*Z-2.279143E-2)*Z+5.515070E-2)*Z-7.261642E-2)
     1*Z+4.987716E-2)*Z-3.332519E-3)*Z-2.314617E-2)*Z-1.134953E-5)*Z
     2+6.250011E-2)*Z+2.583989E-10
      V=(((((((((-5.108699E-3*Z+2.819179E-2)*Z-6.537283E-2)*Z
     1+7.902034E-2)*Z-4.400416E-2)*Z-7.945556E-3)*Z+2.601293E-2)*Z
     2-3.764000E-4)*Z-3.122418E-2)*Z-6.646441E-7)*Z+2.500000E-1
      CI=Z*(SI*V-Y*U)
      SI=-Z*(SI*U+Y*V)
      IF(K)5,6,6
    5 SI=3.141593E0-SI
    6 RETURN
      END
```

3.3.13 Function GRTHFCN

Purpose: This function evaluates:

$$e^{i\alpha z} K_o(z)$$

where K_o is the modified Bessel function.

Method: The procedure is as follows:

1) Compute $e^{i\alpha z}$.

2) Compute $K_o(z)$.

3) Compute the function value.

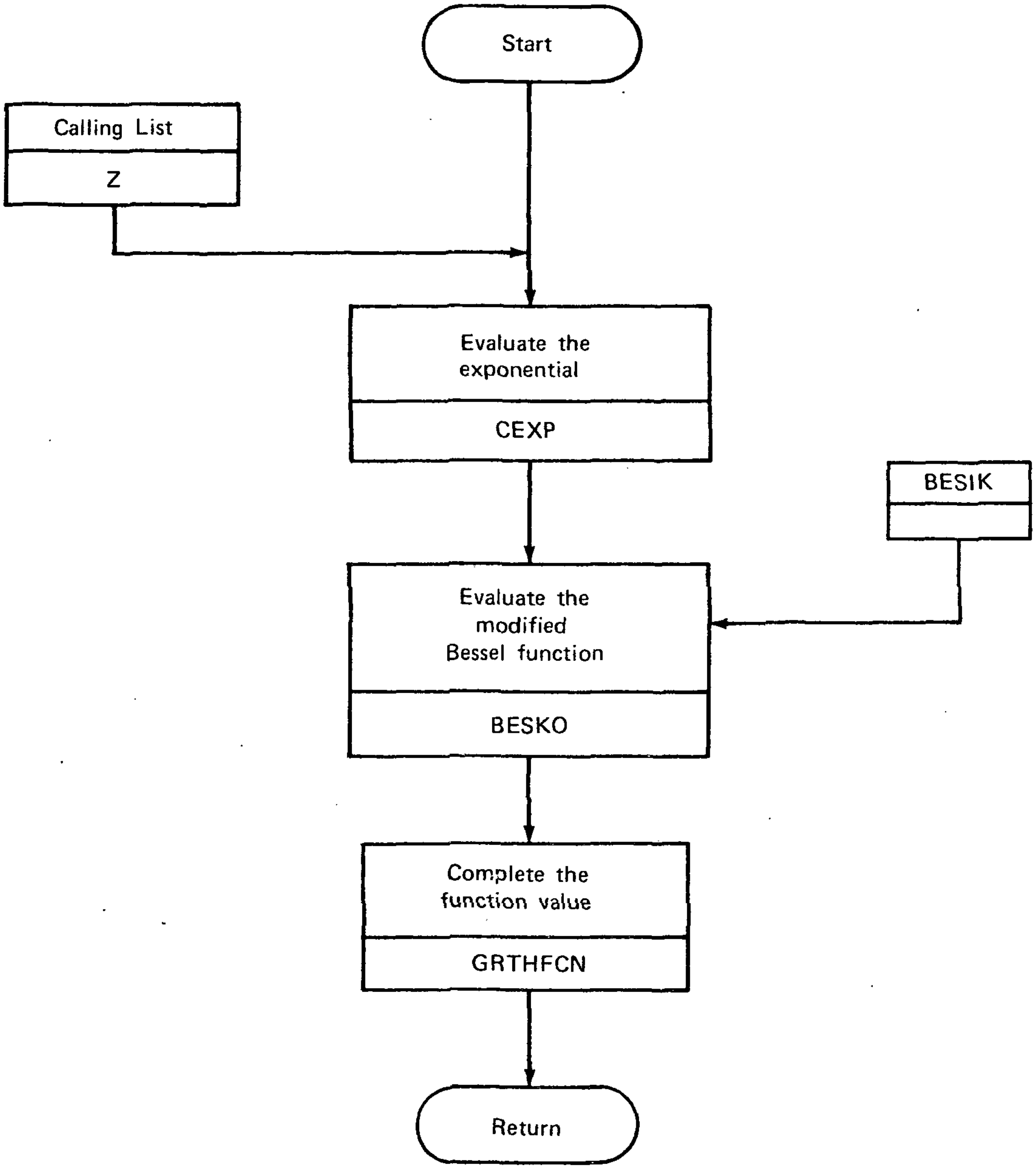
Usage: CALLING SEQUENCE

```
COMPLEX GRTHFCN,VALFCN
COMMON/ALPHA/ALPHA
•
•
•
VALFCN = GRTHFCN(Z)
```

Timing: The timing is approximately equal to one unit call to subroutine BESIK.

Accuracy: The accuracy is of the algorithmic type and, in particular, is dominated by subroutine BESIK.

Boeing Commercial Airplane Company
P.O. Box 3707
Seattle, Washington 98124, May 31, 1974.


Start
Calling List
Z
Evaluate the exponential
CEXP
BESIK
Evaluate the modified Bessel function
BESKO
Complete the function value
GRTHFCN
Return

```
      COMPLEX FUNCTION GRTHFCN(Z)
C
      COMMON/ALPHA/ALPHA
      COMPLEX CEXP
C
      ARG = ALPHA*Z
      CEXP = CMPLX( COS(ARG),-SIN(ARG) )
C
      CALL BESIK(Z,4,BESI0,BESI1,BESK0,BESK1,IERBES)
C
      GRTHFCN = CEXP*BESK0
C
      RETURN
      END
```

REFERENCES

1. Gutin, L.: On the Sound Field of a Rotating Propeller. Physik. Z. Sowjetunion, vol. 9, 1936, p. 57. Also, NACA TN 1194 (Trans.), Oct. 1948.

2. Van de Vooren, A. I.; and Zandbergen, P. J.: Noise Field of a Rotating Propeller in Forward Flight. AIAA J., vol. 1, no. 7, 1963.

3. Slutsky, S.; Mariano, S.; Baronti, P.; and Esses, H.: Analysis of Turbofan Sound Generation and Propagation. Technical report 234, Gen. App. Sci. Lab, Inc., 1961. Also, Slutsky, S.: Discrete Noise Generation and Propagation by a Fan Engine. Presented at the AFOSR-UTIAS Symposium on Aerodynamic Noise, Toronto, 1968.

4. Kemp, N. H.; and Sears, W. R.: Aerodynamic Interference Between Moving Blade Rows. J. Aeron. Sci., vol. 20, no. 9, 1953, p. 585.

5. Kemp, N. H.; and Sears, W. R.: The Unsteady Forces Due to Viscous Wakes in Turbomachines. J. Aeron. Sci., vol. 22, no. 7, 1955.

6. Tyler, J. M.; and Sofrin, T. G.: Axial Flow Compressor Noise Studies. SAE Trans., vol. 70, 1962, p. 309.

7. Hetherington, R.: Compressor Noise Generated by Fluctuating Lift Resulting From Rotor-Stator Interactions. AIAA J., vol. 1, 1963, p. 473.

8. Griffith, J. W. R.: The Spectrum of Compressor Noise of a Jet Engine. J. Sound Vib., vol. 2, 1964, p. 127.

9. Hulse, B.; Pearson, C.; Abbona, M.; and Andersson, A.: Some Effects of Blade Characteristics on Compressor Noise Levels. FAA-ADS-82, 1966.

10. Lighthill, M. J.: On Sound Generated Aerodynamically, Vol. I—General Theory. Proc. Roy. Soc., Series A, vol. 211, 1952.

11. Curle, N.: The Influence of Solid Boundaries Upon Aerodynamic Sound. Proc. Roy. Soc., Series A, vol. 231, 1955, p. 505.

12. Lowson, M. V.: The Sound Field of Singularities in Motion. Proc. Roy. Soc., Series A, vol. 286, 1965.

13. Lowson, M. V.: Theoretical Studies of Compressor Noise. NASA CR-1287, 1969.

14. Ffowcs Williams, J. E.; and Hawkings, D. L.: Theory Relating to the Noise of Rotating Machinery. J. Sound Vib., vol. 10, no. 1, 1969, p. 10.

15. Wright, S. E.: Sound Radiation From a Lifting Rotor Generated by Asymmetric Loading. J. Sound Vib., vol. 9, 1969, p. 223.

16. Morse, P. M.; and Ingard, K. U.: Theoretical Acoustics. McGraw-Hill, 1968.

17. Lowson, M. V.: Rotor Noise Radiation in Non-Uniform Flow. Presented at the Symposium on Aerodynamic Noise, Loughborough Univ. of Tech., England, 1970.

18. Pfenninger, W.: Note About Modulated Rotor Tone Noise of Subsonic Fans and Compressor Induced by Atmospheric Turbulence, Its Possible Reduction Through IGV's, and Design Consequences for Subsonic Fans. Technical document D6-25147, The Boeing Company, 1971.

19. Hanson, D. B.: The Spectrum of Turbomachine Rotor Noise Caused by Inlet Guide Vane Wakes and Atmospheric Turbulence. Document HSER 6191, Hamilton-Standard, 1973.

20. Hanson, D. B.: Unified Analysis of Fan Stator Noise. J. Acoust. Soc. Am., vol. 54, no. 6, 1974, p. 1571.

21. Lordi, J. A.: Report on a Study of Noise Generation by a Rotating Blade Row in an Infinite Annulus. Interim scientific report AI-2836-A-1, U.S. Air Force Office of Scientific Research, May 1971.

22. Drischler, J.: Analytic Studies of Sound Pressures Inside the Duct of Ducted Propellers. NASA TN D-6345, 1971.

23. Schaut, L. A.: Analytical Fan Tone Program. Technical document D6-25356TN, The Boeing Company, 1971.

24. Zorumski, W.: Acoustic Theory of Axisymmetric Multisectioned Ducts. NASA TR R-419, 1974.

25. Lansing, D.: Exact Solution for Radiation of Sound From a Semi-Infinite Circular Duct With Application to Fan and Compressor Noise. Presented at symposium held at Ames Research Center, Moffett Field, Calif., NASA SP-228, 1969, p. 323.

26. Clark, T. L.: Theory of Sound Generation by Ducted Turbomachinery, Vol. I--Acoustic Propagators. Technical document D6-40610, The Boeing Company, 1974.

27. Lansing, D.; and Zorumski, W. E.: Effects of Wall Admittance Changes on Duct Transmission and Radiation of Sound. J. Sound Vib., vol. 27, no. 1, 1971, p. 85.

28. Feynman, R. P.: Theory of Fundamental Processes. W. A. Benjamin, Inc., pub., 1962.

29. Morse, P. M.; and Feshbach, H.: Methods of Theoretical Physics. McGraw-Hill, 1953.

30. Abramowitz, M.; and Stegun, I. A.: Handbood of Mathematical Functions. AMS-55, Nat. Bur. Std., 1964.

31. Atassi, H.; and Goldstein, M.: Unsteady Lift Forces on Highly Cambered Airfoils Moving Through a Gust. Preprint 74-88, AIAA 12th Aerospace Sciences Meeting, 1974.

32. Sears, W. R.: Some Aspects of Nonstationary Airfoil Theory and Its Practical Applications. J. Aeron. Sci., vol. 8, no. 2, 1941, p. 104.

33. Kemp, N. H.: On the Lift and Circulation of Airfoils in Some Unsteady-Flow Problems. J. Aeron. Sci., vol. 19, no. 10, 1952, p. 713.

34. Horlock, J. H.: Fluctuation Lift Forces on Airfoils Moving Through Transverse and Chordwise Gusts. Trans. ASME, J. of Basic Eng., Series D, vol. 90, no. 4, 1968, p. 494.

35. Naumann, H.; and Yeh, H.: Lift and Pressure Fluctuations of a Cambered Airfoil Under Periodic Gusts and Applications in Turbomachinery. J. Eng. for Power, Jan. 1973, p. 1.

36. Filotas, L. T.: Theory of Airfoil Response in a Gusty Atmosphere, Vol. 1--Aerodynamic Transfer Function. UTIAS report 139, 1969; and Response of an Infinite Wing to an Oblique Sinusoidal Gust. 1969 Basic Aerodynamic Noise Research. NASA SP-207, 1969, p. 231.

37. Silverstein, A.; Katzoff, S.; and Bullivant, W. K.: Downwash and Wake Behind Plain and Flapped Airfoils. NACA report 651, 1939.

38. Glauert, H.: The Elements of Airfoil and Airscrew Theory. Second ed., Cambridge Univ. Press, England, 1948.

39. Lowson, M. W.; and Ollerhead, J. B.: A Theoretical Study of Helicopter Rotor Noise. J. Sound Vib., vol. 9, 1969, p. 197.

40. Barry, B.; and Moore, C. J.: Subsonic Fan Noise. J. Sound Vib., vol. 17, no. 2, 1971, p. 207.

41. Anon.: Subroutine BF4F. Vol. I of Computer Programming Manual, sec. C3.3, NASA-Langley Research Center, Aug. 1, 1968.

42. Anon.: Subroutine MTLUP. Vol. I of Computer Programming Manual, sec. E1.5, NASA-Langley Research Center, Sept. 12, 1969.

43. Anon.: Subroutine BSSLS. Vol. I of Computer Programming Manual, sec. C3.2, NASA-Langley Research Center, Aug. 1, 1968.

44. Levine, H.; and Schwinger, J.: On the Radiation of Sound From an Unflanged Circular Pipe. Phys. Rev., vol. 73, no. 4, 1948, p. 383.

45. Kaji, S.; and Okasaki, T.: Generation of Sound by Rotor-Stator Interaction. J. Sound Vib., vol. 13, no. 3, 1970, p. 281.

46. Mani, R.: Compressibility Effects in the Kemp-Sears Problem. Presented at the International Symposium on the Fluid Mechanics and Design of Turbomachinery, Pennsylvania State Univ., 1970.

47. Jarratt, P.; and Nudds, D.: The Use of Rational Functions in the Iterative Solution of Equations on a Digital Computer. The Computer Journal, vol. 8, no. 1, April 1965.

48. Anon.: Bessel Functions, Part II. Vol. X of Mathematical Tables, British Assoc. for Advance. Sci., Cambridge Univ. Press, 1952, p. xvii.

49. Olver, F. W. J.: Error Analysis of Miller's Recurrence Algorithm. Math. Computation, vol. 18, 1964, pp. 65-70.

50. Goldstein, M.; and Thaler, R. M.: Recurrence Techniques for the Calculation of Bessel Functions. MTAC, vol. 13, 1959, pp. 102-103.

51. Abramowitz, M.; and Stegun, I. A.: Generation of Bessel Functions on High Speed Computers. MTAC, vol. ii, 1957, pp. 255-257.

52. Anon.: Bessel Functions, Part I. Vol. X of Mathematical Tables, British Assoc. for Advance. Sci., Cambridge Univ. Press, 1958, pp. 202-212.

53. Goldstein, M.; and Abramson, H.: Subroutine NYU BES4. Control Data 6000 Series Users Group, VIM C3.

54. Anon.: Subroutine ALGAMF. Vol. I of Computer Programming Manual, sec. C3.5, NASA-Langley Research Center, August 1, 1968.

55. Anon.: Subroutine SICI, System/360, Scientific Subroutine Package (360A-CM-03X), Version II. Programmer.s Manual, H20-0205-3.