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
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.
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0   INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>2.0   DICTIONARY</td>
<td>2</td>
</tr>
<tr>
<td>2.1   Guide to Dictionary</td>
<td>2</td>
</tr>
<tr>
<td>2.2   Dictionary of FORTRAN name</td>
<td>2</td>
</tr>
<tr>
<td>3.0   SUBPROGRAM DOCUMENTATION</td>
<td>55</td>
</tr>
<tr>
<td>3.1   Primary Subroutine Descriptions</td>
<td>57</td>
</tr>
<tr>
<td>3.1.1 Subroutine AAAAAA</td>
<td>57</td>
</tr>
<tr>
<td>3.1.2 Subroutine AABAA</td>
<td>75</td>
</tr>
<tr>
<td>3.1.3 Subroutine BCDAA</td>
<td>88</td>
</tr>
<tr>
<td>3.1.4 Subroutine BBCAA</td>
<td>103</td>
</tr>
<tr>
<td>3.2   Secondary Special-Purpose Subprogram Descriptions</td>
<td>117</td>
</tr>
<tr>
<td>3.2.1 Subroutine EGNVAL</td>
<td>117</td>
</tr>
<tr>
<td>3.2.2 Subroutine ZEROS</td>
<td>127</td>
</tr>
<tr>
<td>3.2.3 Function EQATION</td>
<td>135</td>
</tr>
<tr>
<td>3.2.4 Function UNEQFORM</td>
<td>141</td>
</tr>
<tr>
<td>3.2.5 Function EGNORM</td>
<td>146</td>
</tr>
<tr>
<td>3.2.6 Function FACTINT</td>
<td>149</td>
</tr>
<tr>
<td>3.2.7 Function FACTIN2</td>
<td>154</td>
</tr>
<tr>
<td>3.2.8 Function FACTIN3</td>
<td>164</td>
</tr>
<tr>
<td>3.2.9 Function FACTIN4</td>
<td>172</td>
</tr>
<tr>
<td>3.2.10 Subroutine LIFTFN2</td>
<td>185</td>
</tr>
<tr>
<td>3.2.11 Subroutine LIFTFN3</td>
<td>191</td>
</tr>
<tr>
<td>3.2.12 Subroutine LIFTFN4</td>
<td>196</td>
</tr>
<tr>
<td>3.2.13 Function DISINT</td>
<td>204</td>
</tr>
<tr>
<td>3.2.14 Function FUNIN4</td>
<td>207</td>
</tr>
<tr>
<td>3.2.15 Subroutine NONCPT</td>
<td>209</td>
</tr>
</tbody>
</table>
3.0 SUBPROGRAM DOCUMENTATION

3.3 Secondary General-Purpose Subprogram Descriptions...... 217

3.3.1 Subroutine APROX1............................................ 217
3.3.2 Subroutine APROX2............................................ 221
3.3.3 Subroutine JARRATT.......................................... 226
3.3.4 Subroutine GAUSS............................................. 233
3.3.5 Subroutine GAUSS2............................................ 238
3.3.6 BSSLs............................................................... 242
3.3.7 BSNX............................................................... 242
3.3.8 BESJLA............................................................ 250
3.3.9 BESIE............................................................. 255
3.3.10 BESIK............................................................ 259
3.3.11 ROCABES......................................................... 265
3.3.12 SICI............................................................... 276
3.3.13 GRTHFCN......................................................... 279

REFERENCES..................................................................... 282
SYMBOLS

a

$\alpha_z$, $\alpha_{\phi}$

A

$A_n$, $A_{n_{\kappa2}}$

$A_{\frac{\text{t}}{\text{m}}}$

b₁, b₂

C, C₁, C₂, Cₖ₁, Cₖ₂

C₃ ..., C₁₄

C_D₁, C_D₂

d

$\frac{\text{d}C_L}{\text{d}x}$, $\frac{\text{d}C_{L1}}{\text{d}x}$, $\frac{\text{d}C_{L2}}{\text{d}x}$

$\frac{\text{d}C_{L3}}{\text{d}x}$, \left( \frac{\text{d}C_L}{\text{d}x} \right) _1$

\left( \frac{\text{d}C_L}{\text{d}x} \right) _{k1}$

$\hat{\text{d}}_{\text{mn}}$

$\hat{D}_{\text{mn}}$

$D_{\text{mn}}$

$\vec{e}$

$e_{\phi}$

$e_z$

$E_z$, $E_{\phi}$, $E_{z\phi}$

a constant

eddy transverse length scale

radial location of the maximum distortion in the cone model

n-th order Glauert coefficients

a mode amplitude, or coefficient of eigenfunction expansion of the pressure spectral density

blade spacings of a two-dimensional cascade of blades

airfoil chord lengths

switches used in the generalized equations for the potential flow field interactions

airfoil section drag coefficient

axial spacing between midchords of two blade rows

slope of steady lift versus angle of attack

a function defined by equation (B41)

a function defined by equation (2.1.22)

a function defined by equation (2.1.15)

unit vector perpendicular to airfoil chordline

z-component of $\vec{e}$

$\phi$-component of $\vec{e}$

bandpass filter factor; see equation (3.3.23)
ratio of maximum camber to the half-chord of a thin airfoil

strength of single dipole or a surface distribution of dipoles; also, function defined by equation (3.1.15)

spectral density of the strength of a single dipole, or a surface distribution of dipoles

function defined by equation (3.1.20)

function defined by equation (3.1.21)

complex term in the equation for the induced velocities resulting from a potential flow field interaction

airfoil acoustic response function, defined by equation (3.1.39)

oblique gust wave number

the nondimensional dipole surface density response function for convected, harmonic gusts

Hankel function of the first kind of order \( m \)

Hankel function of the second kind of zero\(^{th}\) order

Hankel function of the second kind of order one

complex term in the equation for the induced velocities resulting from a potential flow field interaction, equation (3.2.31)

modified Bessel function of the first kind and order \( \ell \)

dummy index or blade number index

\( J_0 + iJ_1 \)

Bessel function of order \( m \)

Bessel function of order zero

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  axial propagation wave number of mn th mode
K_0,1, K_k, K_l  complex conjugate of Kemp-Sears lift response function
L  dummy index, or harmonic index
L  airfoil section lift force
L_j  spectral density of airfoil section lift force
L_z, L_¿  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_EK  mean exit velocity from a blade row, relative to the blades of the row
M_i, M_ii, M_2i, M_3i, M_iKl  mean inlet velocity to a blade row, relative to the blades of the row
M_M, M_1M, M_2M, M_3M  mean velocity through a blade row, relative to the blades of the row
M_k, M_1k, M_2k, M_K, M_kl  tip velocity of rotating blades
M_T  mean axial flow velocity
M_z, M_1z, M_2z, M_3z, M_zK  mean axial flow velocity when there is steady distortion
M_z  dummy index, harmonic index, or radial mode index
n  number of blades in a blade row
N, N_1, N_2, N_3, N_K2  number of blades in a blade row
linear pressure perturbation

spectral density of linear pressure perturbation

exponent in the "power law," equation (3.3.8)

complex term in the equation for the induced velocities resulting from a potential flow field interaction, equation (3.2.3)

position vector, field location

position vector, source location

radial location of eddy center

unnormalized radial eigenfunction

distance from origin in Prandtl-Glauert scaled coordinates

axial propagation wave number of mn th mode in Prandtl-Glauert scaled coordinates and frequency

Sears function

sound pressure level, measured in decibels

time at field location

time at source location

function defined by equation (3.1.17)

Filotas lift response function

eddy temporal length scale

relative velocity of two-dimensional cascade of blades

spectral density of perturbation velocity

gust convection velocity

value of the maximum velocity distortion in the cone model
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V_1 )</td>
<td>value of the velocity at the outer radius in the cone model</td>
</tr>
<tr>
<td>( W, W_j )</td>
<td>perturbation velocity, viscous wake defect, unsteady induced velocity</td>
</tr>
<tr>
<td>( \hat{W} )</td>
<td>single spatial Fourier transform of ( W )</td>
</tr>
<tr>
<td>( \tilde{W} )</td>
<td>double spatial Fourier transform of ( W )</td>
</tr>
<tr>
<td>( W_{zl}, W_{\phi l} )</td>
<td>perturbation velocity components at the eddy center</td>
</tr>
<tr>
<td>( \hat{W}<em>{zl}, \hat{W}</em>{\phi l} )</td>
<td>( l )th Fourier series coefficient of the eddy velocity components</td>
</tr>
<tr>
<td>( Y, Y', Y_1', Y_j' )</td>
<td>rectangular coordinate</td>
</tr>
<tr>
<td>( Y_{l,j} )</td>
<td>rectangular coordinate</td>
</tr>
<tr>
<td>( Y_m )</td>
<td>Neumann function of order ( m )</td>
</tr>
<tr>
<td>( Y_o )</td>
<td>width of viscous wake; also, Neumann function of order ( o )</td>
</tr>
<tr>
<td>( z, z', z_1', z'_1 )</td>
<td>rectangular coordinate</td>
</tr>
<tr>
<td>( z_{l,j} )</td>
<td>rectangular coordinate</td>
</tr>
<tr>
<td>( z_{M.C.} )</td>
<td>axial position of midchord plane</td>
</tr>
<tr>
<td>( * )</td>
<td>multiplier</td>
</tr>
<tr>
<td>( \alpha )</td>
<td>mean blade angle of attack</td>
</tr>
<tr>
<td>( \beta )</td>
<td>relative stagger angle</td>
</tr>
<tr>
<td>( \beta_{mn} )</td>
<td>square root term defined by equation (2.1.16)</td>
</tr>
<tr>
<td>( \gamma, \gamma_1, \gamma_2, \gamma_{K1}, \gamma_{K2} )</td>
<td>stagger angle</td>
</tr>
<tr>
<td>( \hat{\gamma} )</td>
<td>Helmholtz equation Green's function</td>
</tr>
<tr>
<td>( \Gamma )</td>
<td>acoustic propagator</td>
</tr>
<tr>
<td>( \Gamma_2, \Gamma_{K2} )</td>
<td>steady-state circulation of cascade airfoil</td>
</tr>
</tbody>
</table>
δ

δ_{m-2,σN}

Δ₀

Δ_{z,ϕ}

ε

ζ

η

Θ

κ

κ^{±}_{mn}

λ, λ', λ_{σ,1}, λ_{κ,K1}

Λ_{k, σ}

ν_{0,l}

ν_{mn}

ν, ν', ν_{σ,1}, ν_{σ,1}', ν_{κ,K1}

ξ

ξ'

ρ, ρ_σ, ρ_s

σ

τ

Dirac delta function

Kronecker delta symbol

factor in viscous wakes formula, equation (3.1.35)

axial eddy strength modulation function

small, positive constant; also, unit step function

rectangular coordinate normal to the airfoil

hub-to-tip ratio, annular duct inner radius

inverse cosine of nondimensionalized chordwise coordinate

temporal or spacial harmonic index

chordwise compactness parameter; see equation (2.2.18)

wave number; also, reduced complex frequency of the chordwise velocity distribution

Fourier coefficients of typical wake profile

complex frequency of the chordwise velocity distribution

annular duct eigenvalue

reduced frequency

chordwise rectangular coordinate

chordwise rectangular coordinate nondimensionalized to the half-chord

polar radial coordinate of cylindrical coordinate system

temporal Fourier series coefficient, dummy index used in summations

time delay resulting from the axial distance between the midchord plane and the eddy center position at the temporal origin; also, τ = τ₀
\( \phi, \phi_o, \phi_s, \phi_{\text{M.C.}} \)  
- polar angle coordinate of cylindrical coordinate system

\( \phi_j, \phi_s, \phi \)  
- polar angle coordinate of eddy center

\( \psi \)  
- relative exit flow angle; also, oblique gust angle

\( \omega, \omega^*, \omega_{s, l} \)  
- angular frequency

\( \Omega \)  
- angular velocity of rotor

Subscripts:

\( E \)  
- blade row exit flow

\( I \)  
- blade row inlet flow

\( j \)  
- \( j \)th blade

\( K \)  
- either Kl or K2

\( K_l \)  
- sound-producing blade row

\( K_2 \)  
- velocity-inducing blade row

\( \ell \)  
- spatial harmonic index

\( m \)  
- spinning mode index

\( M \)  
- blade row mean flow

\( \text{M. C.} \)  
- midchord point location

\( n \)  
- radial mode index

\( S \)  
- source

\( z \)  
- axial direction, axial eddy velocity component

\( \kappa \)  
- either \( \sigma \) or \( \ell \)

\( \sigma \)  
- temporal Fourier series index

\( \phi \)  
- 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:

† downstream (+) and upstream (−) propagation

† complex conjugate

* complex conjugate; also, generalized Prandtl-Glauert transform, variable

; blade-attached rectangular coordinate non-dimensionalized to the half-chord

→ vector

^- temporal Fourier integral transform

- 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
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, BCDA, and BCAA 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

<table>
<thead>
<tr>
<th>Subprogram</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAAAA</td>
<td>1</td>
</tr>
<tr>
<td>ZEROS</td>
<td>3</td>
</tr>
<tr>
<td>EQUATION</td>
<td>4</td>
</tr>
<tr>
<td>UNCONFN</td>
<td>5</td>
</tr>
<tr>
<td>EGNORM</td>
<td>6</td>
</tr>
<tr>
<td>FACTINT</td>
<td>7</td>
</tr>
<tr>
<td>AAABAA</td>
<td>9</td>
</tr>
<tr>
<td>FACTIN2</td>
<td>10</td>
</tr>
<tr>
<td>LIFTFN2</td>
<td>11</td>
</tr>
<tr>
<td>BCDAA</td>
<td>12</td>
</tr>
<tr>
<td>EGNVAL2</td>
<td>13</td>
</tr>
<tr>
<td>FACTIN3</td>
<td>14</td>
</tr>
<tr>
<td>LIFTFN3</td>
<td>15</td>
</tr>
<tr>
<td>DISINT</td>
<td>16</td>
</tr>
<tr>
<td>BCCAA</td>
<td>17</td>
</tr>
<tr>
<td>NONCPT</td>
<td>18</td>
</tr>
<tr>
<td>FACTIN4</td>
<td>19</td>
</tr>
<tr>
<td>FUNIN4</td>
<td>20</td>
</tr>
<tr>
<td>LIFTFN4</td>
<td>21</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Subprogram</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>APROX1</td>
<td>50</td>
</tr>
<tr>
<td>APROX2</td>
<td>51</td>
</tr>
<tr>
<td>JARRATT</td>
<td>52</td>
</tr>
<tr>
<td>GAUSS</td>
<td>53</td>
</tr>
<tr>
<td>BSSL5</td>
<td>54</td>
</tr>
<tr>
<td>BF4F</td>
<td>55</td>
</tr>
<tr>
<td>MLTUP</td>
<td>56</td>
</tr>
<tr>
<td>GAUSS2</td>
<td>57</td>
</tr>
<tr>
<td>ROCABES</td>
<td>58</td>
</tr>
<tr>
<td>ALGAMF</td>
<td>59</td>
</tr>
<tr>
<td>BESIE</td>
<td>60</td>
</tr>
<tr>
<td>BESJLA</td>
<td>61</td>
</tr>
<tr>
<td>BESIX</td>
<td>62</td>
</tr>
<tr>
<td>BESIK</td>
<td>63</td>
</tr>
<tr>
<td>SICI</td>
<td>64</td>
</tr>
<tr>
<td>GRTFCN</td>
<td>65</td>
</tr>
</tbody>
</table>
### FUNCTION CODE

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

### 2.2 Dictionary of FORTRAN Names

<table>
<thead>
<tr>
<th>FORTRAN name</th>
<th>LOC-FNC code</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAAAA</td>
<td>1-6</td>
<td>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</td>
</tr>
<tr>
<td>AABAA</td>
<td>9-1</td>
<td>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</td>
</tr>
<tr>
<td>FORTRAN name</td>
<td>LOC-FNC code</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>------------</td>
</tr>
<tr>
<td>AMN</td>
<td>14-4</td>
<td>Array of dimension 2 which contains the cosine and sine distortion coefficients $a_i(\rho)$, $b_i(\rho)$ of index $</td>
</tr>
<tr>
<td>ABSLAM</td>
<td>11-4</td>
<td>Absolute value of the complex variable LAMDA, $</td>
</tr>
<tr>
<td>ABSKAPA</td>
<td>18-4</td>
<td>Absolute value of RKAPA, $</td>
</tr>
<tr>
<td>ABSNU</td>
<td>15-4,18-4,21-4</td>
<td>Absolute value of the reduced frequency RNU, $</td>
</tr>
<tr>
<td>ALFA</td>
<td>19-4</td>
<td>The blade angle of attack, $\alpha$; this quantity is input in AR(I,11,K) as a function of radial position</td>
</tr>
<tr>
<td>ALGAMF</td>
<td>59-6</td>
<td>The standard LRC subroutine which computes the log of the gamma function for complex arguments</td>
</tr>
<tr>
<td>ALPHA</td>
<td>21-4,21-5,65-5</td>
<td>Tan $\theta$, where $\theta$ is the gust yaw angle used in the Filotas lift response function; also, name of common block containing ALPHA</td>
</tr>
<tr>
<td>ALPHAMN</td>
<td>1-3,9-3,12-3,17-3</td>
<td>Complex array of dimension NDIM x MDIM which contains the matrix of mode amplitudes, $a_{mn}$, where $a_{mn} = ALPHAMN(N,I)$ with $M = MUSE(I)$</td>
</tr>
<tr>
<td>APROX1</td>
<td>3-4,50-6</td>
<td>Subroutine which calculates approximate zeros of equation (6), where $0.2 &lt; \eta &lt; 1.0$ ($\eta$ is the hub-to-tip ratio)</td>
</tr>
<tr>
<td>FORTRAN name</td>
<td>LOC-FNC code</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>------------</td>
</tr>
<tr>
<td>APROX2</td>
<td>3-4,51-6</td>
<td>Subroutine which calculates approximate zeros of equation (6), where $0 &lt; \eta &lt; .2$</td>
</tr>
<tr>
<td>AR</td>
<td>1-1,7-1,9-1, 10-1,12-1,14-1, 17-1,19-1</td>
<td>Array of dimension MAXDIM x MAXJ x 3, where AR(I,J,K) contains data described as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>K = 1: Inlet stator data</td>
</tr>
<tr>
<td></td>
<td></td>
<td>K = 2: Rotor data</td>
</tr>
<tr>
<td></td>
<td></td>
<td>K = 3: Outlet stator data</td>
</tr>
<tr>
<td></td>
<td></td>
<td>J = 1: Nondimensional duct radial position, $\rho$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>J = 2: Nondimensional chord, $C(\rho)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>J = 3: Not used</td>
</tr>
<tr>
<td></td>
<td></td>
<td>J = 4: Drag coefficient, $C_D(\rho)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>J = 5: Steady-state lift coefficient, $C_L(\rho)$, which is not required as an input for any of the existing primary subroutines</td>
</tr>
<tr>
<td></td>
<td></td>
<td>J = 6: Derivative of $C_L$ with respect to incident angle, $\alpha$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\frac{dC_L}{d\alpha}(\rho)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>J = 7: Relative inflow Mach number of a blade row, $M_{I}(\rho)$; see figure 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>J = 8: Relative exit flow Mach number of a blade row, $M_{E}(\rho)$; see figure 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>J = 9: Axial flow Mach number, $M_{Z}(\rho)$; see figure 3</td>
</tr>
<tr>
<td>FORTRAN name</td>
<td>LOC-FNC code</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Note: ARMISC(18+K) can be zero, in which case no Glauert coefficients are input.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>J = 10: Glauert coefficients of order 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>J = 11: Glauert coefficients of order 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>J = ARMISC(18+K) + 9: Glauert coefficients of order ARMISC(18+K) -1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If ARMISC(25) = 3,

| Description |
| 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,

| Description |
| 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 |
| 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 \( 2\cdot r \), \( a_{2\cdot r} \) |

If ARMISC(22) = 3
<table>
<thead>
<tr>
<th>FORTRAN name</th>
<th>LOC-FNC code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARG</td>
<td>16-4</td>
<td>Argument (-2\phi) of the exponential in equation (48)</td>
</tr>
<tr>
<td>ARGETA</td>
<td>5-4</td>
<td>Product of eigenvalue and hub-to-tip ratio, (\nu_{mn} \cdot \eta)</td>
</tr>
<tr>
<td>ARGEXP</td>
<td>1-4, 9-4</td>
<td>Argument of the exponential in the constant factor of the integrand; see equations (33), (45), (53), (57)</td>
</tr>
</tbody>
</table>

If \(\text{ARMISC}(22) = 3\) (concluded)

\[
J = \text{ARMISC}(18+K) + P + 2\cdot \text{MAXCOEF} + 1: \text{Sine distortion coefficient of index MAXCOEF}\cdot r, \quad b_{\text{MAXCOEF}\cdot r}, \text{where:}
\]

\[
P = 10 \text{ if } \text{ARMISC}(25) \neq 3
\]
\[
P = 12 \text{ if } \text{ARMISC}(25) = 3
\]
\[
r = \text{MULTFCT}
\]

\(I = 1:\) 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

\(I = 3:\) Set of values corresponding to

\(I = 4:\) nondimensional duct radial positions. The nondimensional duct radial positions must be in increasing order.
<table>
<thead>
<tr>
<th>FORTRAN name</th>
<th>LOC-FNC code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARGEXP2</td>
<td>7-4</td>
<td>Argument of the exponential in the oscillatory factor of the integrand; see equation (36)</td>
</tr>
<tr>
<td>ARGS</td>
<td>5-4</td>
<td>Argument $(\mu_{mn} \cdot S)$ of the unnormalized duct radial eigenfunction $R_m$ in equation (6). It is actually the product of an eigenvalue $\mu_{mn}$ and the dummy argument $S$ of subroutine UNEGFMN.</td>
</tr>
<tr>
<td>ARHO</td>
<td>10-4</td>
<td>Variable $a_{\kappa \mathbf{K} \mathbf{L}}(\rho)$ as defined in equation (38)</td>
</tr>
<tr>
<td>ARMISC</td>
<td>1-1,7-1,9-1, 10-1,12-1,14-1, 17-1,19-1</td>
<td>Array of dimension 40, where ARMISC(I) contains data described as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$I = 1$: Nondimensionalized average distance between the midchord planes of the inlet guide vanes and the rotor; see figure 4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$I = 2$: Nondimensional average distance between the midchord planes of the rotor and the outlet guide vanes; see figure 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$I = 3$: Hub-to-tip ratio, $n$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$I = 4$: Option IFLOW, where:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-1 indicates upstream sound propagation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 indicates downstream sound propagation</td>
</tr>
<tr>
<td>FORTRAN name</td>
<td>LOC-FNC code</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| 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  
<p>|              |              | 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, $\phi_{OS}$, in radians; see figure 14 |</p>
<table>
<thead>
<tr>
<th>FORTRAN name</th>
<th>LOC-FNC code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARMISC (continued)</td>
<td>13:</td>
<td>Phase angle for adjustment of skewness of the incident wake at the rotor, $\phi_R$, in radians; see figure 14</td>
</tr>
<tr>
<td></td>
<td>14:</td>
<td>Harmonic index, $\sigma$</td>
</tr>
<tr>
<td></td>
<td>15:</td>
<td>Axial position of the inlet stator, $Z_{IS}$</td>
</tr>
<tr>
<td></td>
<td>16:</td>
<td>Axial position of the outlet stator, $Z_{QS}$</td>
</tr>
<tr>
<td></td>
<td>17:</td>
<td>Axial position of the rotor, $Z_R$</td>
</tr>
<tr>
<td></td>
<td>18:</td>
<td>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</td>
</tr>
<tr>
<td></td>
<td>19:</td>
<td>Number of inlet stator vane Glauert coefficients</td>
</tr>
<tr>
<td></td>
<td>20:</td>
<td>Number of rotor blade Glauert coefficients</td>
</tr>
<tr>
<td></td>
<td>21:</td>
<td>Number of outlet stator vane Glauert coefficients</td>
</tr>
<tr>
<td>FORTRAN name</td>
<td>LOC-FNC code</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>ARMISC</td>
<td></td>
<td><strong>I = 22</strong>: Distortion model selector, where:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 0 indicates no distortion</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 1 indicates distortion is represented by the cone model; see equation (48)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 2 indicates distortion is represented by the power model; see equation (49)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 3 indicates that the distortion coefficients are input; see equation (50)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>I = 23</strong>: Distortion input, where:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If ARMISC(22) = 1, ARMISC(23) = VADBV1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If ARMISC(22) = 2, ARMISC(23) = $Q$.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If ARMISC(22) = 3, ARMISC(23) = MAXCOEF.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>I = 24</strong>: Distortion input, where:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If ARMISC(22) = 1, ARMISC(24) = CAPADIS.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If ARMISC(22) = 3, ARMISC(24) = MULTFCT.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>I = 25</strong>: Lift response function selector, where:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 indicates the generalized Sears lift response function (LIFTFN2) used with the primary subroutine AABAA; see equation (24)</td>
</tr>
<tr>
<td>FORTRAN name</td>
<td>LOC-FNC code</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>ARMISC (continued)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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).

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, \( \phi \), 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, \( W_\phi \), 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, \( a_n \); see figure 10
<table>
<thead>
<tr>
<th>FORTRAN name</th>
<th>LOC-FNC code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARMISC (concluded)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>I = 33:</td>
<td></td>
<td>Nondimensional eddy length scale in the direction normal to the average flow velocity for the angular eddy velocity component, $a_\phi$; see figure 10</td>
</tr>
<tr>
<td>I = 34:</td>
<td></td>
<td>Nondimensional eddy length scale in the direction of the average flow velocity for the axial eddy velocity component; $L_z$; see figure 10</td>
</tr>
<tr>
<td>I = 35:</td>
<td></td>
<td>Nondimensional eddy length scale in the direction of the average flow velocity for the angular eddy velocity component, $L_\phi$; see figure 10</td>
</tr>
<tr>
<td>I = 36:</td>
<td></td>
<td>Upper bound of the frequency band considered in the generation of tone duct mode amplitudes by non-steady distortion, $B$; see figure 11</td>
</tr>
<tr>
<td>I = 37:</td>
<td></td>
<td>Time when eddy center is located in rotor plane, $\tau$</td>
</tr>
<tr>
<td>I = 38:</td>
<td></td>
<td>Compactness selector</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 indicates compact source option (LIFTFN3 is used)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>≠0 indicates noncompact source option (NONCPT is used); can be used only if ARMISC(25) = 3</td>
</tr>
<tr>
<td>I = 39:</td>
<td></td>
<td>Not used</td>
</tr>
<tr>
<td>I = 40:</td>
<td></td>
<td>Not used</td>
</tr>
<tr>
<td>FORTRAN</td>
<td>LOC-FNC</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>code</td>
<td>Description</td>
</tr>
<tr>
<td>ARMUMN</td>
<td>1-2S,3-3,9-2S, 12-2S,13-3, 17-2S</td>
<td>Array of dimension NDIM x MDIM which contains the matrix of eigenvalues where ARMUMN(N,M) = ( \mu_{MN} )</td>
</tr>
<tr>
<td>AV</td>
<td>1-4,12-4,17-4</td>
<td>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:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I = 1: Midpoint of the subinterval locally used in the integration of the integral of equation (9)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Average inlet stator vane chord if ISOROS = 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I = 2: Average rotor blade chord if ISOROS = 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Used with primary subroutine AAAAA only.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I = 3: Average blade chord of the sound-generating blade row</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Average inlet stator vane drag coefficient if ISOROS = 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I = 4: Average rotor blade drag coefficient if ISOROS = 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Used with primary subroutine AAAAA only.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I = 5: Not used</td>
</tr>
<tr>
<td>FORTRAN name (concluded)</td>
<td>LOC-FNC code</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>AV</td>
<td></td>
<td>I = 6: Average derivative of the steady-state lift coefficient with respect to the angle of incidence for the sound-generating blade row</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I = 7: Average relative inflow Mach number of the sound-generating blade row</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I = 8: Average relative exit flow Mach number of the sound-generating blade row</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I = 9: Average axial flow Mach number of the sound-generating blade row</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I = 10: Average value of $\ell$, the ratio of the maximum camber to the half-chord for the sound-generating blade row</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I = 11: Average value of $\alpha$, the blade angle of attack for the sound-generating blade row</td>
</tr>
<tr>
<td>AVSPAN</td>
<td>1-4, 12-4, 17-4</td>
<td>Midpoint of the subinterval locally used in the computation of the integral of equation (9). It is equivalenced to AV(1).</td>
</tr>
<tr>
<td>AXIALM</td>
<td>1-4, 9-4, 12-4, 17-4</td>
<td>Average axial Mach number of the sound-generating blade row</td>
</tr>
<tr>
<td>A1</td>
<td>4-4</td>
<td>A value of $J_M(x)$, Bessel function of first kind and order M</td>
</tr>
<tr>
<td>A2</td>
<td>4-4</td>
<td>A value of $J_{M+1}(x)$, Bessel function of first kind and order $(M+1)$</td>
</tr>
<tr>
<td>FORTRAN name</td>
<td>LOC-FNC code</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>A3</td>
<td>4-4</td>
<td>A value of ( Y_M(x) ), Bessel function of second kind (the Neumann function) and order ( M )</td>
</tr>
<tr>
<td>A4</td>
<td>4-4</td>
<td>A value of ( Y_{M+1}(x) ), Bessel function of second kind (the Neumann function) and order ( M+1 )</td>
</tr>
<tr>
<td>B</td>
<td>9-5,10-5</td>
<td>In the case of an inlet guide vane-rotor interaction (( K1+K2 = 3 )), ( B = \text{ARMISC}(1) ). In the case of a rotor-outlet guide vane interaction (( K1+K2 = 5 )), ( B = \text{ARMISC}(2) ).</td>
</tr>
<tr>
<td>BBCAA</td>
<td>17-6</td>
<td>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</td>
</tr>
<tr>
<td>BCDAA</td>
<td>12-6</td>
<td>Primary subroutine which computes the acoustic pressure mode amplitudes resulting from the interaction of a rotor with a distorted inflow</td>
</tr>
<tr>
<td>BES</td>
<td>4-5,5-5,10-5</td>
<td>Array of dimension 1000 in common block SCRATCH used as a scratch array by Bessel function subroutines</td>
</tr>
<tr>
<td></td>
<td>11-5,15-5,18-5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>19-5,21-5</td>
<td></td>
</tr>
<tr>
<td>BESIE</td>
<td>19-4</td>
<td>The subroutine which computes ( I_{\ell}(x)e^{-x} ) when ( I ) is the modified Bessel function, ( \ell ) is an integer, and ( x ) is a real argument</td>
</tr>
</tbody>
</table>

17
<table>
<thead>
<tr>
<th>FORTRAN name</th>
<th>LOC-FNC code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BESIEJ</td>
<td>19-4</td>
<td>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 $I_\ell$ a modified Bessel function of order $\ell$; see equation (68).</td>
</tr>
<tr>
<td>BESIK</td>
<td>21-4,63-6</td>
<td>Subroutine which computes modified Bessel functions with real argument $x$, $I_0(x)$, $I_1(x)$, $K_0(x)$, $K_1(x)$</td>
</tr>
<tr>
<td>BESIO</td>
<td>21-4</td>
<td>A value of $I_0(x)$, a modified Bessel function of order 0</td>
</tr>
<tr>
<td>BESI1</td>
<td>21-4</td>
<td>A value of $I_1(x)$, a modified Bessel function of order 1</td>
</tr>
<tr>
<td>BESJLA</td>
<td>61-6</td>
<td>Subroutine that computes $J_\nu(x)$, a Bessel function of the first kind and order $\nu$, where $x \gg \nu$</td>
</tr>
<tr>
<td>BESJLAM</td>
<td>11-4</td>
<td>A variable which contains $J_0(\text{LAMDA}) - i J_1(\text{LAMDA})$</td>
</tr>
<tr>
<td>BESKO</td>
<td>21-4</td>
<td>A value of $K_0(x)$, a modified Bessel function of order 0</td>
</tr>
<tr>
<td>BESK1</td>
<td>21-4</td>
<td>A value of $K_1(x)$, a modified Bessel function of order 1</td>
</tr>
<tr>
<td>BESNX</td>
<td>18-4,62-6</td>
<td>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 $x$</td>
</tr>
<tr>
<td>FORTRAN name</td>
<td>LOC-FNC code</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>BETAMN</td>
<td>1-4,9-4,12-4,17-4</td>
<td>$\beta_{mn}$, which is defined in equation (1)</td>
</tr>
<tr>
<td>BF4F</td>
<td>4-4,5-4,11-4,15-4,18-4,55-6</td>
<td>Standard LRC library subroutine which calculates the Bessel function of the second kind, or Neumann function, $Y$</td>
</tr>
<tr>
<td>BJHI</td>
<td>10-4</td>
<td>Array of dimension 250 containing the imaginary parts of the Bessel function of the first kind with complex arguments, $J_0(h_{K2}[\rho]), J_1(h_{K2}[\rho]), \ldots$</td>
</tr>
<tr>
<td>BJHR</td>
<td>10-4</td>
<td>Array of dimension 250 containing the real parts of the Bessel function of the first kind with complex arguments, $J_0(h_{K2}[\rho]), J_1(h_{K2}[\rho]), \ldots$</td>
</tr>
<tr>
<td>BJLAMI</td>
<td>11-4</td>
<td>Array of dimension 250 containing the imaginary parts of the Bessel function of the first kind with complex arguments, $J_0(\text{LAMDA}), J_1(\text{LAMDA})$</td>
</tr>
<tr>
<td>BJLAMR</td>
<td>11-4</td>
<td>Array of dimension 250 containing real parts of the Bessel function of the first kind with complex arguments, $J_0(\text{LAMDA}), J_1(\text{LAMDA})$</td>
</tr>
<tr>
<td>BJ1</td>
<td>15-4,18-4</td>
<td>$J_0(x)$, a value of the Bessel function of the first kind, with order zero and real argument $x$</td>
</tr>
<tr>
<td>FORTRAN name</td>
<td>LOC-FNC code</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>BJ1LAM</td>
<td>11-4</td>
<td>$J_0(\text{LAMDA})$, a value of the Bessel function of the first kind, with order zero and complex argument LAMDA</td>
</tr>
<tr>
<td>BJ1RNU</td>
<td>11-4</td>
<td>$J_0(\text{RNU})$, a value of the Bessel function of the first kind, with order zero and real argument RNU</td>
</tr>
<tr>
<td>BJ2</td>
<td>15-4,18-4</td>
<td>$J_1(x)$, a value of the Bessel function of the first kind, with order one and real argument x</td>
</tr>
<tr>
<td>BJ2LAM</td>
<td>11-4</td>
<td>$J_1(\text{LAMDA})$, a value of the Bessel function of the first kind, with order one and complex argument LAMDA</td>
</tr>
<tr>
<td>BJ2RNU</td>
<td>11-4</td>
<td>$J_1(\text{RNU})$, a value of the Bessel function of the first kind, with order one and real argument RNU</td>
</tr>
<tr>
<td>BSSLS</td>
<td>4-4,5-4,11-4, 15-4,18-4,5-6</td>
<td>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.</td>
</tr>
<tr>
<td>BTAU</td>
<td>19-4</td>
<td>Variable containing $B\cdot \tau$; see equation (61).</td>
</tr>
<tr>
<td>BTJ</td>
<td>19-4</td>
<td>Array of dimension 2 which contains $B\cdot T_j$ for $j = 1,2$; see equation (61).</td>
</tr>
<tr>
<td>FORTRAN name</td>
<td>LOC-FNC code</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>BYLAMI</td>
<td>11-4</td>
<td>Array of dimension 50 which is required in calling subroutine ROCABES</td>
</tr>
<tr>
<td>BYLAMR</td>
<td>11-4</td>
<td>Array of dimension 50 which is required in calling subroutine ROCABES</td>
</tr>
<tr>
<td>BY1</td>
<td>15-4,18-4</td>
<td>$Y_0(x)$, a value of the Bessel function of the second kind, with order zero and argument $x$</td>
</tr>
<tr>
<td>BY1RNU</td>
<td>11-4</td>
<td>$Y_0(RNU)$, a value of the Bessel function of the second kind, with order zero and real argument $RNU$</td>
</tr>
<tr>
<td>BY2</td>
<td>15-4,18-4</td>
<td>$Y_1(x)$, a value of the Bessel function of the second kind, with order one and argument $x$</td>
</tr>
<tr>
<td>BY2RNU</td>
<td>11-4</td>
<td>$Y_1(RNU)$, a value of the Bessel function of the second kind, with order one and real argument $RNU$</td>
</tr>
<tr>
<td>B1, B2, B3</td>
<td>1-4,12-4,15-1,18-1,19-4</td>
<td>Coefficients used with subroutines LIFTFW3 and NONCPT; see equations (21) and (22)</td>
</tr>
<tr>
<td>CAPA</td>
<td>10-4</td>
<td>Array of dimension 15 which contains average Glauert coefficients for component K2</td>
</tr>
<tr>
<td>CAPADIS</td>
<td>14-5,16-5</td>
<td>Contains $A$ used in the cone model of distortion; see equation (33)</td>
</tr>
<tr>
<td>FORTRAN name</td>
<td>LOC-FNC code</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>CAPF1</td>
<td>19-4</td>
<td>Variable containing $F_1(p)$; see equations (64) and (65)</td>
</tr>
<tr>
<td>CAPF2</td>
<td>19-4</td>
<td>Variable containing $F_2(p)$; see equations (66) and (67)</td>
</tr>
<tr>
<td>CAPHRHO</td>
<td>10-4</td>
<td>A variable which contains $K_{\kappa K_2}(p)$; see equation (41)</td>
</tr>
<tr>
<td>CAPKL</td>
<td>11-4</td>
<td>A variable containing $K_L(v, \lambda)$; see equation (24) where $v &gt; 0$</td>
</tr>
<tr>
<td>CAPKMN</td>
<td>1-5,7-5,9-5, 10-5,12-5,14-5, 16-5,17-5,19-5</td>
<td>$K_{mn}$, variable defined in equation (2)</td>
</tr>
<tr>
<td>CAPKRHO</td>
<td>10-4</td>
<td>A variable containing $K_{\kappa KL}(p)$</td>
</tr>
<tr>
<td>CAPLT</td>
<td>1-4,12-4,15-3,18-3</td>
<td>Contains a value of the combined lift response function. If $\text{ARMISC}(38) = 0$, $\text{CAPLT} = L(v)$; see equation (21). If $\text{ARMISC}(38) \neq 0$, $\text{CAPLT} = L'(v)$; see equation (22).</td>
</tr>
<tr>
<td>CAPNMN</td>
<td>1-5,7-5,9-5, 10-5,12-5,14-5, 16-5,17-5,19-5</td>
<td>The normalization factor for the duct radial eigenfunction, $N_{mn}$, which is defined by equation (7)</td>
</tr>
<tr>
<td>CAPRETA</td>
<td>6-4</td>
<td>$R_m(u_{mn})$, value of unnormalized eigenfunctions with argument the product of an eigenvalue times the hub-to-tip ratio</td>
</tr>
</tbody>
</table>
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 | 1-5,16-5 | 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, n

CFACT | 1-5,12-5,14-5, 7-5,16-5,17-5, 19-5 | Common block containing CAPNHN, 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, CAPNHN, 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
<table>
<thead>
<tr>
<th>FORTRAN name</th>
<th>LOC-PNC code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHORD</td>
<td>10-4,19-4</td>
<td>Array of dimension 3 which contains values of nondimensional chords for the three components</td>
</tr>
<tr>
<td>CI</td>
<td>21-4</td>
<td>Value of the cosine integral CI(x), where x &gt; 0</td>
</tr>
<tr>
<td>CLIFTU</td>
<td>19-4</td>
<td>Variable containing T*(x,y), the complex conjugate of a value of the Pilotas lift response function (LIFTFTU)</td>
</tr>
<tr>
<td>CMACH</td>
<td>1-4,9-4,12-4,17-4</td>
<td>1. - (AXIALM)^2</td>
</tr>
<tr>
<td>COEFA1</td>
<td>14-4</td>
<td>Contains a1 used in the power model of distortion; see equation (49)</td>
</tr>
</tbody>
</table>
| CONLIFT      | 10-4,11-3    | If v > 0, CONLIFT = K_L (v, \lambda). If v < 0, CONLIFT = [K_L (-v, -\lambda^*)]^*.
<p>|              |              | See equation (28). |
| COSPSI       | 1-4,7-4      | Cosine of the angle \phi, 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 \gamma; see equation (11) |
| COTBETA      | 1-4,12-4     | Cotangent of \beta, the relative stagger angle; see figure 3 |</p>
<table>
<thead>
<tr>
<th>FORTRAN name</th>
<th>LOC-FNC code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COTTHS</td>
<td>19-4</td>
<td>Cotangent of the mean flow angle; see equation (11)</td>
</tr>
<tr>
<td>CTEMP1</td>
<td>10-4,15-4,18-4,19-4</td>
<td>Variables used for temporary storage of complex numbers</td>
</tr>
<tr>
<td>CTEMP2</td>
<td>7-4,10-4,19-4,21-4</td>
<td>Variables used for temporary storage of complex numbers</td>
</tr>
<tr>
<td>CTEMP3</td>
<td>21-4</td>
<td>Variables used for temporary storage of complex numbers</td>
</tr>
<tr>
<td>CTJ</td>
<td>19-5,20-5</td>
<td>Variable containing a value of $T_j$; see equation (60)</td>
</tr>
<tr>
<td>C1</td>
<td>1-4</td>
<td>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).</td>
</tr>
<tr>
<td>C2</td>
<td>1-4,12-4,17-4,18-1</td>
<td>Contains the average blade chord of the sound-generating blade row. C2 is made equivalent to AV(3).</td>
</tr>
<tr>
<td>C3,C6,C7,C8</td>
<td>9-5,10-5</td>
<td>Variables in common block CFACT2 which are defined by the table on the following page.</td>
</tr>
<tr>
<td>C9,C11,C12,C13,C14</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Description

<table>
<thead>
<tr>
<th>FORTRAN name</th>
<th>LOC-FNC code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCL</td>
<td>1-4,12-4, 17-4</td>
<td>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).</td>
</tr>
<tr>
<td>DCSBL</td>
<td>10-4</td>
<td>An array of dimension 3 which contains slopes of the steady-state lift coefficients for each of the three components</td>
</tr>
<tr>
<td>DELK</td>
<td>19-4</td>
<td>Variable containing $\Delta = a_j/R$; see equation (70)</td>
</tr>
<tr>
<td>DELTALO</td>
<td>14-4</td>
<td>Contains $\delta_{L,0}$ where $\delta_{j,k}$ is the Kronecker delta</td>
</tr>
<tr>
<td>DELTAL1</td>
<td>14-4</td>
<td>Contains $(\delta_{L,1} + \delta_{L,-1})$ where $\delta_{j,k}$ is the Kronecker delta</td>
</tr>
</tbody>
</table>

| ISOROS | 1 | 2 | 2 | 1 |
| IAERO | 1 | 1 | -1 | -1 |
| C3 | -1 | 1 | 1 | -1 |
| C6 | -1 | 1 | -1 | 1 |
| C7 | -1 | -1 | -1 | 1 |
| C8 | 1 | -1 | -1 | 1 |
| C9 | 1 | -1 | -1 | 1 |
| C11 | 1 | 1 | -1 | -1 |
| C12 | -1 | -1 | -1 | 1 |
| C13 | -1 | -1 | -1 | -1 |
| C14 | 1 | -1 | -1 | 1 |

<table>
<thead>
<tr>
<th>SIGOL ≤ -1</th>
<th>SIGOL ≥ 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>FORTRAN name</td>
<td>LOC-FNC code</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
</tr>
<tr>
<td>DISINT</td>
<td>14-4,16-7</td>
</tr>
<tr>
<td>DRHO</td>
<td>10-4</td>
</tr>
<tr>
<td>DSPAC</td>
<td>1-4,7-4</td>
</tr>
<tr>
<td>DVALUE</td>
<td>14-4</td>
</tr>
<tr>
<td>D1</td>
<td>4-4</td>
</tr>
<tr>
<td>D2</td>
<td>4-4</td>
</tr>
<tr>
<td>FORTRAN name</td>
<td>LOC-FNC code</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------</td>
</tr>
<tr>
<td>D3</td>
<td>4-4</td>
</tr>
<tr>
<td>D4</td>
<td>4-4</td>
</tr>
<tr>
<td>EGNBND</td>
<td>1-4,9-4,12-4, 13-1,17-4</td>
</tr>
<tr>
<td>EGNBND0</td>
<td>1-4,9-4, 12-4,17-4</td>
</tr>
<tr>
<td>EGNNORM</td>
<td>1-4,9-4,12-4, 17-4,6-7</td>
</tr>
<tr>
<td>EGNVAL2</td>
<td>1-4,9-4,12-4, 13-6,17-4</td>
</tr>
<tr>
<td>EJ</td>
<td>19-4</td>
</tr>
<tr>
<td>EPS</td>
<td>21-4</td>
</tr>
<tr>
<td>EP1</td>
<td>3-4</td>
</tr>
<tr>
<td>FORTRAN name</td>
<td>LOC-FNC code</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
</tr>
<tr>
<td>EP2</td>
<td>3-4</td>
</tr>
<tr>
<td>EQUATION</td>
<td>3-4,4-7</td>
</tr>
<tr>
<td>ETA</td>
<td>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</td>
</tr>
<tr>
<td>ETOA</td>
<td>1-4,9-4, 12-4,17-4</td>
</tr>
<tr>
<td>EXPDRHO</td>
<td>10-4</td>
</tr>
<tr>
<td>EXPFACT</td>
<td>7-4</td>
</tr>
<tr>
<td>FACTAV</td>
<td>1-4,12-4,17-4</td>
</tr>
<tr>
<td>FACTCON</td>
<td>1-4,9-4, 12-4,17-4</td>
</tr>
<tr>
<td>FORTRAN name</td>
<td>LOC-FNC code</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------</td>
</tr>
<tr>
<td>FACTINT</td>
<td>1-4,7-7</td>
</tr>
<tr>
<td>FACTIN2</td>
<td>9-4,10-7</td>
</tr>
<tr>
<td>FACTIN3</td>
<td>12-4,14-7</td>
</tr>
<tr>
<td>FACTIN4</td>
<td>17-4,19-7</td>
</tr>
<tr>
<td>FALPHNU</td>
<td>15-4</td>
</tr>
<tr>
<td>FFNU</td>
<td>15-4</td>
</tr>
<tr>
<td>FJFP</td>
<td>4-4</td>
</tr>
<tr>
<td>FORTRAN name</td>
<td>LOC-FNC code</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
</tr>
<tr>
<td>FJP</td>
<td>4-4</td>
</tr>
<tr>
<td>FKMAX</td>
<td>19-4</td>
</tr>
<tr>
<td>FME</td>
<td>1-4,12-4, 17-4</td>
</tr>
<tr>
<td>FMI</td>
<td>1-4,12-4, 17-4</td>
</tr>
<tr>
<td>FMM</td>
<td>1-4,12-4</td>
</tr>
<tr>
<td>FMZ</td>
<td>1-4,12-4, 17-4, 7-4</td>
</tr>
<tr>
<td>FMIE</td>
<td>1-4,7-4</td>
</tr>
<tr>
<td>FNFP</td>
<td>4-4</td>
</tr>
<tr>
<td>FORTRAN name</td>
<td>LOC-FNC code</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
</tr>
<tr>
<td>FNP</td>
<td>4-4</td>
</tr>
<tr>
<td>FNU</td>
<td>18-4</td>
</tr>
<tr>
<td>FRHO</td>
<td>19-4</td>
</tr>
<tr>
<td>FRTH</td>
<td>21-4</td>
</tr>
<tr>
<td>FT</td>
<td>3-4</td>
</tr>
<tr>
<td>FUNIN(4)</td>
<td>19-4,20-7</td>
</tr>
<tr>
<td>FUNPHI</td>
<td>16-4</td>
</tr>
<tr>
<td>GAMMA</td>
<td>10-4</td>
</tr>
<tr>
<td>GAUSS</td>
<td>17-4,19-4, 21-4,53-6</td>
</tr>
<tr>
<td>FORTRAN name</td>
<td>LOC-FNC code</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
</tr>
<tr>
<td>GAUSS2</td>
<td>1-4, 9-4, 12-4, 17-4, 57-6</td>
</tr>
<tr>
<td>GJ</td>
<td>19-4</td>
</tr>
<tr>
<td>GNRHO</td>
<td>10-4</td>
</tr>
<tr>
<td>GRTHFCN</td>
<td>21-4, 65-7</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>GUESS</td>
<td>3-4</td>
</tr>
<tr>
<td>HALFP1</td>
<td>21-4</td>
</tr>
<tr>
<td>FORTRAN name</td>
<td>LOC-FNC code</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
</tr>
</tbody>
</table>
| HANKEL       | 11-4         | Variable which contains the ratio \[
\frac{H_{1}^{(2)}(\text{RNU})}{H_{1}^{(2)}(\text{RNU}) + i \, H_{0}^{(2)}(\text{RNU})}\]

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 \(h_{\text{K2}}(\rho)\); 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 \(H_{0}^{(2)}(\text{RNU}) = J_{0}(\text{RNU}) - i \, Y_{0}(\text{RNU})\), a Hankel function of the second kind and order zero |
| H2RNU        | 11-4,15-4, 18-4 | Variable containing \(H_{1}^{(2)}(\text{RNU}) = J_{1}(\text{RNU}) - i \, Y_{1}(\text{RNU})\), a Hankel function of the second kind and order one |
| IABSL        | 14-4         | Absolute value of \(L\), the incidence velocity Fourier series index |

34
<table>
<thead>
<tr>
<th>FORTRAN name</th>
<th>LOC-PNC code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IABSM</td>
<td>5-4,13-4</td>
<td>The absolute value of a spinning node index, $</td>
</tr>
<tr>
<td>IAERO</td>
<td>9-4</td>
<td>Option IAERO = ARMISC(10)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-1 indicates the upstream component is sound generator</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 indicates the downstream component is sound generator</td>
</tr>
<tr>
<td>IERBES</td>
<td>21-4</td>
<td>Error return from BESIK, where:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 indicates no error</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 indicates input argument is nonpositive; no calculation is possible</td>
</tr>
<tr>
<td>IEREGNV</td>
<td>1-4,9-4,12-4, 13-3,17-4</td>
<td>Error return from EGNVAL2, where:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 indicates successful execution</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 indicates that there are more eigenvalues required than there is space for (i.e., MDIM and/or MDIM are not large enough); as many as possible are returned</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 indicates that there are no eigenvalues</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IEREGNV is equivalent to IERROR.</td>
</tr>
<tr>
<td>IERJAR</td>
<td>3-4</td>
<td>Error return from JARRATT</td>
</tr>
<tr>
<td>IERLIFT$^4$</td>
<td>19-4,21-3</td>
<td>Error return from LIFTFN$^4$, where:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 indicates no error</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 indicates integral in FRTH did not converge according to EPS on the interval $[X, 1000]$</td>
</tr>
<tr>
<td>FORTRAN name</td>
<td>LOC-FNC code</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>IERR</td>
<td>4-4,11-4,5-4, 15-4,18-4</td>
<td>Error return from BSSLS and BF4F. It is not used.</td>
</tr>
<tr>
<td>IERROR</td>
<td>1-3,9-3, 12-3,17-3</td>
<td>Error return from primary subroutines, where:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 indicates successful execution</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 indicates that there are no eigenvalues; IERROR is equivalent to IEREGNV.</td>
</tr>
<tr>
<td>IFLOW</td>
<td>1-4,9-4, 12-4,17-4</td>
<td>Option IFLOW = ARMISC(4)</td>
</tr>
<tr>
<td>IFORM</td>
<td>19-4,21-1</td>
<td>Option IFORM, where:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 indicates that the exact form of the Filotas lift response function is used</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 indicates that the approximate form of the Filotas lift response function is used</td>
</tr>
<tr>
<td></td>
<td></td>
<td>At the present, IFORM is set equal to 2.</td>
</tr>
<tr>
<td>IGO</td>
<td>11-4</td>
<td>If RNU &gt; 0, IGO = 1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If RNU &lt; 0, IGO = 2.</td>
</tr>
<tr>
<td>ILOGIC</td>
<td>9-4</td>
<td>Internal switch which is defined by</td>
</tr>
<tr>
<td></td>
<td>ISOROS</td>
<td>1 2 2 1</td>
</tr>
<tr>
<td></td>
<td>IAERO</td>
<td>1 1 -1 -1</td>
</tr>
<tr>
<td></td>
<td>ILOGIC</td>
<td>1 2 1 2</td>
</tr>
<tr>
<td>FORTRAN name</td>
<td>LOC-FNC code</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>ILOGICO</td>
<td>9-4</td>
<td>Previous value (of last call to primary subroutine) of ILOGIC. It is initially set to 0.</td>
</tr>
<tr>
<td>INDX</td>
<td>9-4</td>
<td>Variable containing the sum of ILOGIC and IAERO</td>
</tr>
<tr>
<td>INDX2</td>
<td>10-4</td>
<td>Variable containing the sum of ISOROS and IAERO</td>
</tr>
<tr>
<td>INTEGJ</td>
<td>19-4</td>
<td>Array of dimension 2 containing the integrals $I_1$ and $I_2$; see equation (70)</td>
</tr>
<tr>
<td>INTEGRL</td>
<td>1-4,9-4,12-4, 12-4,17-4</td>
<td>The value of the constant factor times the integral over the interval $(\eta,1)$. It is equal to a mode amplitude, $a_{mn}$; see equation (9).</td>
</tr>
<tr>
<td>IORDGS</td>
<td>1-4,9-4,12-4, 14-4,17-4,19-4</td>
<td>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</td>
</tr>
<tr>
<td>IP</td>
<td>10-4,19-4</td>
<td>Variable set equal to -1 which is used in MTLUP</td>
</tr>
<tr>
<td>IPA</td>
<td>1-4,7-4,12-4, 14-4,17-4</td>
<td>Variable set equal to -1 which is used in MTLUP</td>
</tr>
<tr>
<td>ISIGN</td>
<td>4-4,11-4,5-4, 15-4,18-4</td>
<td>Variable set equal to -1 which is used in BF4F</td>
</tr>
<tr>
<td>FORTRAN name</td>
<td>LOC-FNC code</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>ISOROS</td>
<td>1-4,7-4,9-4</td>
<td>Option ISOROS = ARMISC(5), where:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 indicates inlet stator-rotor interaction</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 indicates rotor-outlet stator interaction</td>
</tr>
<tr>
<td>ISOROSO</td>
<td>1-4</td>
<td>Previous value (of last call to primary subroutine) of ISOROS. It is initially set equal to 0.</td>
</tr>
<tr>
<td>ITLIM</td>
<td>3-4</td>
<td>Variable set equal to 30 which is used in JARRATT</td>
</tr>
<tr>
<td>ITRACE</td>
<td>1-4,3-1,9-4, 12-4,13-1,17-4</td>
<td>Option ITRACE = ARMISC(6), where:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 indicates no printout</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 indicates printout from primary subroutine</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 indicates printout from primary subroutines and subroutine ZEROS</td>
</tr>
<tr>
<td>JARRATT</td>
<td>3-4,52-6</td>
<td>Subroutine which calculates the zeros of equation (5)</td>
</tr>
<tr>
<td>JMAX</td>
<td>18-4</td>
<td>Upper limit of the summation in the equation for the noncompact acoustic response function; see equation (23). JMAX = MAX(RKAPA,RNU) + 1.</td>
</tr>
<tr>
<td>JMAX1</td>
<td>18-4</td>
<td>JMAX + 1</td>
</tr>
<tr>
<td>JMAX2</td>
<td>18-4</td>
<td>JMAX + 2</td>
</tr>
<tr>
<td>FORTRAN name</td>
<td>LOC-FNC code</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>KI</td>
<td>10-4</td>
<td>Variable used as temporary storage for K1 and K2</td>
</tr>
<tr>
<td>KMAX</td>
<td>19-4</td>
<td>Maximum value of K: KMAX = (FKMAX/DELK) + 1. See equation (70).</td>
</tr>
<tr>
<td>K1,K2</td>
<td>9-5,10-5</td>
<td>Variables which are defined by:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L</td>
<td>1-5,7-5,9-5,10-5,12-5,13-4,14-5,16-5,17-5,19-5</td>
<td>Fourier series index of the incident velocity, l</td>
</tr>
<tr>
<td>LAMDA</td>
<td>10-4,11-1</td>
<td>A variable which contains ( \lambda_{k,K1} ); see equation (44)</td>
</tr>
<tr>
<td>LAMDAI</td>
<td>11-4</td>
<td>A variable containing the imaginary part of LAMDA</td>
</tr>
<tr>
<td>LAMDAR</td>
<td>11-4</td>
<td>A variable containing the real part of LAMDA</td>
</tr>
<tr>
<td>LIFT</td>
<td>10-4,11-3</td>
<td>If ( v &gt; 0 ), LIFT = ( [K_L(v,\lambda)]^* )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If ( v &lt; 0 ), LIFT = ( K_L(-v, -\lambda^*) )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>See equation (24).</td>
</tr>
<tr>
<td>LIFTFN2</td>
<td>10-4,11-6</td>
<td>The subroutine which computes the lift response function used with the primary subroutine AABAA; see reference 4</td>
</tr>
<tr>
<td>FORTRAN name</td>
<td>LOC-FNC code</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>LIFTFN3</td>
<td>1-4,12-4, 15-6,19-4</td>
<td>The subroutine which computes the combined lift response function used with the primary subroutines AAAAA, BCDAA, and BBCAA; see reference 35</td>
</tr>
<tr>
<td>LIFTFN4</td>
<td>19-4,21-6</td>
<td>The subroutine which computes the Filotas lift response function; see reference 36</td>
</tr>
<tr>
<td>LIFT</td>
<td>19-4,21-3</td>
<td>Variable containing $T(X,Y)$, a value of the Filotas lift response function LIFTFN4</td>
</tr>
<tr>
<td>LUSE</td>
<td>13-4</td>
<td>A variable which is used as a counter in computing NOFM</td>
</tr>
<tr>
<td>LZERO</td>
<td>13-1</td>
<td>An option where: 0 indicates that $L = 0$ is acceptable 1 indicates that $L = 0$ is not acceptable</td>
</tr>
<tr>
<td>M</td>
<td>1-5,3-5,9-5, 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</td>
<td>Spinning mode index $m$; see RM</td>
</tr>
<tr>
<td>MAXCOEF</td>
<td>14-4</td>
<td>The number of indexes of the distortion coefficients; see MULTFCT also</td>
</tr>
<tr>
<td>MAXDIM</td>
<td>1-1,9-1,12-1, 17-1</td>
<td>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.</td>
</tr>
<tr>
<td>FORTRAN name</td>
<td>LOC-FNC code</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>MAXIND</td>
<td>10-4</td>
<td>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).</td>
</tr>
<tr>
<td>MAXJ</td>
<td>1-1,9-1, 12-1,17-1</td>
<td>A variable dimension of array AR</td>
</tr>
<tr>
<td>MAXN</td>
<td>1-25,9-25, 12-25,13-3, 17-25</td>
<td>An array of dimension MDIM which contains for each spinning mode index a maximum radial mode index</td>
</tr>
<tr>
<td>MBES</td>
<td>13-4</td>
<td>Variable set equal to 100. It indicates the maximum order of the Bessel function which can be safely calculated by the subroutine BSSLS.</td>
</tr>
<tr>
<td>MBESSEGN</td>
<td>13-4</td>
<td>Bound or magnitude of spinning mode index M due to MBES and EGNBND</td>
</tr>
<tr>
<td>MDIM</td>
<td>1-1,9-1,12-1, 13-1,17-1</td>
<td>A variable column dimension of ALPHAMN and ARMUMN</td>
</tr>
<tr>
<td>MDIMO</td>
<td>1-4,9-4, 12-4,17-4</td>
<td>Previous value (of last call to primary subroutine) of ETA. It is initially set to 0.</td>
</tr>
<tr>
<td>MEGN</td>
<td>13-4</td>
<td>Maximum spinning mode index due to the eigenvalue bound, EGNBND</td>
</tr>
<tr>
<td>MMAX</td>
<td>3-4</td>
<td>Contains the value max (</td>
</tr>
<tr>
<td>FORTRAN name</td>
<td>LOC-FNC code</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>MP1</td>
<td>3-4,4-4</td>
<td>Contains $M + 1$, where $M$ is the spinning mode index</td>
</tr>
<tr>
<td>MP1MAX</td>
<td>3-4</td>
<td>Contains $</td>
</tr>
<tr>
<td>MP2</td>
<td>4-4,5-4</td>
<td>Contains $MP1 + 1$</td>
</tr>
<tr>
<td>MSAVE</td>
<td>3-4</td>
<td>Contains $</td>
</tr>
<tr>
<td>MSBE</td>
<td>10-4,19-4</td>
<td>An array of dimension 3 which contains relative exit flow Mach numbers for the three components</td>
</tr>
<tr>
<td>MSBI</td>
<td>10-4,19-4</td>
<td>An array of dimension 3 which contains relative inlet flow Mach numbers for the three components</td>
</tr>
<tr>
<td>MSBM</td>
<td>10-4,19-4</td>
<td>An array of dimension 3 which contains relative mean flow Mach numbers for the three components; see equation (10)</td>
</tr>
<tr>
<td>MSBT</td>
<td>1-4,9-4,12-4, 17-4,19-4</td>
<td>Rotor blade tip Mach number, $M_T = ARMISC(7)$</td>
</tr>
<tr>
<td>MSBZ</td>
<td>10-4,19-4</td>
<td>An array of dimension 3 which contains axial Mach numbers for the three components</td>
</tr>
<tr>
<td>MTLUP</td>
<td>1-4,7-4,56-6, 10-4,12-4,14-4, 17-4,19-4</td>
<td>Standard LRC library subroutine which performs multiple table lookup; see reference 42</td>
</tr>
<tr>
<td>FORTRAN name</td>
<td>LOC-FNC code</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>MULTFCT</td>
<td>14-4</td>
<td>The multiplicative factor in the indexes of distortion coefficients. That is, if ( r = \text{MULTFCT} ), then the distortion coefficients are ( a_r, b_r, a_{2r}, b_{2r}, \ldots, a_{r \cdot \text{MAX COEF}}, b_{r \cdot \text{MAX COEF}} ).</td>
</tr>
<tr>
<td>MUSE</td>
<td>1-2S,9-2S,12-2S,13-3,17-2S,3-1</td>
<td>An array of dimension MMIM which contains the set of spinning mode indexes</td>
</tr>
<tr>
<td>N</td>
<td>1-5,7-5,9-5,10-5,12-5,13-4,14-5,16-5,17-5,19-5</td>
<td>( N = n+1 ), where ( n ) is the radial mode index</td>
</tr>
<tr>
<td>NB</td>
<td>11-4</td>
<td>Variable set equal to 1. which is used in BSSLS, BP4F, and ROCABES</td>
</tr>
<tr>
<td>NBESEGN</td>
<td>13-4</td>
<td>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</td>
</tr>
<tr>
<td>NDIM</td>
<td>1-1,13-1,3-1,9-1,12-1,17-1</td>
<td>Variable row dimension of ALPHAMN and ARMUMN</td>
</tr>
<tr>
<td>NDIMO</td>
<td>1-4,9-4,12-4,17-4</td>
<td>Previous value (of last call to primary subroutine) of NDIM. It is initially set to 0.</td>
</tr>
<tr>
<td>FORTRAN name</td>
<td>LOC-FNC code</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>NK1</td>
<td>9-4</td>
<td>If ( KI = 1 ), ( NKI ) is the number of inlet stator vanes.</td>
</tr>
<tr>
<td>NK2</td>
<td>9-5,10-5</td>
<td>If ( KI = 2 ), ( NKI ) is the number of rotor blades.</td>
</tr>
<tr>
<td>NMAX</td>
<td>3-1,13-4</td>
<td>If ( KI = 3 ), ( NKI ) is the number of outlet stator vanes.</td>
</tr>
<tr>
<td>NM1</td>
<td>13-4</td>
<td>Contains ( N-1 ).</td>
</tr>
<tr>
<td>NOFM</td>
<td>1-25,9-28,3-1,12-28,13-3,17-28</td>
<td>Number of spinning mode indexes</td>
</tr>
<tr>
<td>NOFN</td>
<td>1-4,9-4,12-4,13-4,17-4</td>
<td>Contains the maximum radial mode index corresponding to a spinning mode index</td>
</tr>
<tr>
<td>NONCPT</td>
<td>1-4,12-4,18-6,19-4</td>
<td>Subroutine which computes the noncompact acoustic response function for the noncompact source theory; see equation (22)</td>
</tr>
<tr>
<td>NOSCE</td>
<td>1-4,9-4,12-4,14-4,17-4,19-4</td>
<td>Number of equal subintervals used in evaluating an integral</td>
</tr>
<tr>
<td>NPTS</td>
<td>10-4,19-4</td>
<td>A variable containing the number of points in an array</td>
</tr>
<tr>
<td>NSBIR</td>
<td>1-5,7-5</td>
<td>If ( ISOROS = 1 ), ( NSBIR ) contains the number of inlet stator vanes, ( N_{ISV} ).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If ( ISOROS = 2 ), ( NSBIR ) contains the number of rotor blades, ( N_{RB} ).</td>
</tr>
<tr>
<td>FORTRAN name</td>
<td>LOC-PNC code</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| NSBNK1       | 9-4          | If ILOGIC = 1, NSBNK1 is NK1.  
                           If ILOGIC = 2, NSBNK1 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 $AR(I,J,K)$, where:
                                  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 integra-
                                  tion of equation (A1) in Reference 36.
                                  Either:
                                  $$N1*\text{WIDTHI} < x \leq (N1+1)*\text{WIDTHI}$$
                                  $$\text{(N1-1)*WIDTHI} < x \leq N1*\text{WIDTHI}$$ |
| PHISBIR      | 1-5, 7-5     | If ISOROS = 1, PHISBIR contains the span-
                                  dependent phase angle at the rotor, $\phi_R$.
                                  If ISOROS = 2, PHISBIR contains the span-
                                  dependent phase angle at the outlet
                                  stator, $\phi_QS$. |
<table>
<thead>
<tr>
<th>FORTRAN name</th>
<th>LOC-FNC code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PI</td>
<td>1-1,9-1,12-1, 14-1,15-1,17-1, 18-1,19-1,21-1</td>
<td>Contains the value $\pi$</td>
</tr>
<tr>
<td>PJ</td>
<td>19-1</td>
<td>Array of dimension 2 which contains $P_j$, $j = 1,2$; see equation (61)</td>
</tr>
<tr>
<td>PSI</td>
<td>19-1</td>
<td>Variable containing $\psi(p, k)$, the gust yaw angle used in Filotas lift response function; see equation (72)</td>
</tr>
<tr>
<td>Q</td>
<td>14-1</td>
<td>Contains the exponent $q$ used in the power model of distortion; see equation (49). $q = \text{ARMISC}(23)$ if $\text{ARMISC}(22) = 2$.</td>
</tr>
<tr>
<td>RATIO</td>
<td>6-1</td>
<td>Contains the ratio, $m^2/\nu_{mn}^2$, of the square of a spinning mode $m$ over the square of an eigenvalue $\nu_{mn}$</td>
</tr>
<tr>
<td>RHO</td>
<td>7-1,10-1,14-1, 18-1,19-1</td>
<td>Nondimensional duct radial coordinate, $\rho$</td>
</tr>
<tr>
<td>RHOINC</td>
<td>14-5,16-5</td>
<td>Same as RHO</td>
</tr>
<tr>
<td>RK</td>
<td>1-1,9-1,12-1, 17-1</td>
<td>Nondimensional frequency, $\omega$</td>
</tr>
<tr>
<td>RKAPA</td>
<td>18-1</td>
<td>$K_{mn\sigma}^+ = \frac{c}{2} \left[ K_{mn}^+ e_\phi - \frac{m}{\rho} e_z \right]$</td>
</tr>
<tr>
<td>RKSQD</td>
<td>1-1,9-1,12-1, 17-1</td>
<td>$\omega^2$, the square of the nondimensional frequency</td>
</tr>
<tr>
<td>FORTRAN name</td>
<td>LOC-FNC code</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>RLOW</td>
<td>1-4, 9-4, 12-4, 14-4, 17-4, 19-4, 21-4</td>
<td>Lower bound of a subinterval used in an integration</td>
</tr>
<tr>
<td>RM</td>
<td>3-4</td>
<td>Spinning mode index in floating point; see M</td>
</tr>
<tr>
<td>RMUMN</td>
<td>1-5, 5-1, 6-1, 7-5, 9-5, 10-5, 12-5, 14-5, 16-5, 17-5, 19-5</td>
<td>An eigenvalue, $u_{mn}$</td>
</tr>
<tr>
<td>RNOFSV</td>
<td>1-4, 12-4, 13-1, 17-4</td>
<td>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.</td>
</tr>
<tr>
<td>RNOFSVO</td>
<td>1-4, 12-4, 17-4</td>
<td>Previous value (of last call to primary subroutine) of RNOFSV. It is initially set to 0.</td>
</tr>
<tr>
<td>RNU</td>
<td>1-4, 11-1, 12-4, 15-1, 18-1, 21-1</td>
<td>Reduced frequency, $v$; see equation (32)</td>
</tr>
<tr>
<td>RNUKAPA</td>
<td>10-4</td>
<td>Reduced frequency, $v_{k,kl}(\rho)$; see equation (43)</td>
</tr>
<tr>
<td>FORTRAN name</td>
<td>LOC-FNC code</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>ROCABES</td>
<td>10-4, 11-4, 21-4, 58-6</td>
<td>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 )</td>
</tr>
<tr>
<td>RSBNKII</td>
<td>9-4</td>
<td>If ILOGIC = 1, RSBNKI is NK2. If ILOGIC = 2, RSBNKI is NK1.</td>
</tr>
<tr>
<td>RSBNKIO</td>
<td>9-4</td>
<td>Previous value (of last call to primary subroutine) of RSBNKI. It is initially set equal to 0.</td>
</tr>
<tr>
<td>RUP</td>
<td>1-4, 9-4, 12-4, 14-4, 17-4, 19-4, 21-4</td>
<td>Upper bound of a subinterval used in evaluating an integral</td>
</tr>
<tr>
<td>S</td>
<td>5-1</td>
<td>Dummy argument of UNEGNFN, where ( 0 &lt; S ). UNEGNFN calculates ( R_m(\mu \ast S) ), the unnormalized duct radial eigenfunction with argument the product of an eigenvalue times ( S ).</td>
</tr>
<tr>
<td>SAVELAM</td>
<td>11-4</td>
<td>A variable used to temporarily save LAMDA</td>
</tr>
<tr>
<td>SAVERNNU</td>
<td>11-4</td>
<td>A variable used to temporarily save RNU</td>
</tr>
<tr>
<td>SC</td>
<td>3-1, 13-4</td>
<td>Array of dimension 40 that is used as a scratch array in ZEROS</td>
</tr>
<tr>
<td>SCRATCH</td>
<td>4-5, 5-5, 10-5, 11-5, 15-5, 18-5, 19-5, 21-5</td>
<td>Common block name which contains the array BES, an array of dimension 1000 that is used as a scratch array in Bessel function subroutines</td>
</tr>
<tr>
<td>FORTRAN name</td>
<td>LOC-FNC code</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>SCPTRMN</td>
<td>7-4,10-4,14-4, 19-4</td>
<td>A value of the normalized duct radial eigenfunction, $R_m (\nu_{mn} \rho)$</td>
</tr>
<tr>
<td>SI</td>
<td>21-4</td>
<td>Value of sine integral $Si(x) = \int_0^x \sin \frac{t}{\tau} d\tau$ where $x$ is real</td>
</tr>
<tr>
<td>SICI</td>
<td>21-4,64-6</td>
<td>Subroutine which computes the sine and cosine integrals</td>
</tr>
<tr>
<td>SIGMA</td>
<td>1-4,9-4,12-4, 17-4</td>
<td>Harmonic index, $\sigma$</td>
</tr>
<tr>
<td>SIGN</td>
<td>1-5,7-5,12-5, 14-5,16-5,19-5</td>
<td>If ISOROS = 1, SIGN contains -1. If ISOROS = 2, SIGN contains 1.</td>
</tr>
<tr>
<td>SIGNKI</td>
<td>9-4</td>
<td>A variable which contains the product of SIGMA and NSBNKI</td>
</tr>
<tr>
<td>SIGNKIO</td>
<td>9-4</td>
<td>Previous value (of last call to primary subroutine) of SIGNKI. It is initially set to 0.</td>
</tr>
<tr>
<td>SIGNRB</td>
<td>1-4,12-4, 13-1,17-4</td>
<td>$\sigma \times N_{RB}$, the product of the harmonic index times the number of rotor blades</td>
</tr>
<tr>
<td>SIGNRBO</td>
<td>1-4,12-4, 17-4</td>
<td>Previous value (of last call to primary subroutine) of SIGNRB; initially set to 0.</td>
</tr>
</tbody>
</table>
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 ≠ 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(v), where S is the Sears lift response function and v 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:
\[
\text{Sum} = \frac{2}{\nu} \sum_{J=1}^{J_{\text{max}}} (-1)^J J_J (\nu_L) \left[ J_{J+1} (\kappa_{mnc}) + J_{J-1} (\kappa_{mnc}) \right]
\]
See equation (23).
<table>
<thead>
<tr>
<th>FORTRAN name</th>
<th>LOC-FNC code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TAU</td>
<td>19-5,20-5</td>
<td>Time delay resulting from the distance between the axial position of the eddy center at the temporal origin and the rotor plane</td>
</tr>
<tr>
<td>TEMP1</td>
<td>1-4,7-4,10-4,12-4,15-4,17-4,18-4,19-4,20-4,21-4</td>
<td></td>
</tr>
<tr>
<td>TEMP2</td>
<td>1-4,10-4,12-4,17-4,18-4,19-4,20-4,21-4</td>
<td>Variables which are used for temporary storage in calculations</td>
</tr>
<tr>
<td>TEMP3</td>
<td>1-4,10-4,12-4,18-4,19-4,21-4</td>
<td></td>
</tr>
<tr>
<td>TEMP4</td>
<td>1-4,10-4,18-4,21-4</td>
<td></td>
</tr>
<tr>
<td>TERM</td>
<td>10-4</td>
<td>Array of dimension 15 which contains the terms of a summation</td>
</tr>
</tbody>
</table>
| 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
<p>|              | 21-4         | Gust yaw angle, ( \psi ), used in the Filotas lift response function; see equation (72) |</p>
<table>
<thead>
<tr>
<th><strong>Fortran Name</strong></th>
<th><strong>LOC-FNC Code</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>TJ</td>
<td>19-4</td>
<td>Array of dimension 2 which contains $T_j$, $j = 1,2$, the temporal length of an eddy; see equation (60)</td>
</tr>
<tr>
<td>TWOPI</td>
<td>1-4,9-4,12-4,</td>
<td>$2\pi$</td>
</tr>
<tr>
<td></td>
<td>17-4,19-4</td>
<td></td>
</tr>
<tr>
<td>T1</td>
<td>5-4</td>
<td>Contains a value of $Y_m^\prime (u_{mn} \cdot \eta)$, 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</td>
</tr>
<tr>
<td>T2</td>
<td>5-4</td>
<td>Contains a value of $J_m (u_{mn} \cdot S)$, the Bessel function of the first kind of order $m$ and argument the product of an eigenvalue, $u_{mn}$, times the dummy argument, $S$, of subroutine UNEGFN</td>
</tr>
<tr>
<td>T3</td>
<td>5-4</td>
<td>Contains a value of $J_m^\prime (u_{mn} \cdot \eta)$, 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</td>
</tr>
<tr>
<td>T4</td>
<td>5-4</td>
<td>Contains a value of $Y_m (u_{mn} \cdot S)$, the Bessel function of the second kind of order $m$ and argument the product of an eigenvalue, $u_{mn}$, times the dummy argument, $S$, of subroutine UNEGFN</td>
</tr>
<tr>
<td>FORTRAN name</td>
<td>LOC-FNC code</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>UNEGFRN</td>
<td>5-7,6-4,7-4, 10-4,14-4, 19-4</td>
<td>Function subprogram which computes the unnormalized duct radial eigenfunction, $R_m(\mu_{mn})$</td>
</tr>
<tr>
<td>VADBVI</td>
<td>14-4</td>
<td>Contains $V_A/V_1$ which is used in the cone model of the distortion; see equation (48)</td>
</tr>
<tr>
<td>VALUINT</td>
<td>1-4,9-4,12-4, 17-4</td>
<td>The value of the integral of the oscillatory factor over a subinterval. It is calculated by subroutine GAUSS2.</td>
</tr>
<tr>
<td></td>
<td>14-4,19-4</td>
<td>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.</td>
</tr>
<tr>
<td>WIDTHI</td>
<td>1-4,9-4,12-4, 14-4,17-4, 19-4,21-4</td>
<td>Width of a subinterval used in evaluating an integral. Every subinterval has the same width, WIDTHI.</td>
</tr>
<tr>
<td>WSLBRHO</td>
<td>14-4</td>
<td>The $L^{th}$ complex distortion coefficient at a duct radial coordinate.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For the cone model, see equation (48).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For the power model, see equation (49).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For the case where the coefficients are input, see equation (50).</td>
</tr>
<tr>
<td>X</td>
<td>1-4</td>
<td>Distance along mean streamline traveled by wake</td>
</tr>
<tr>
<td></td>
<td>4-1</td>
<td>or Dummy argument of function subprogram,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EQATION 53</td>
</tr>
<tr>
<td>FORTRAN name</td>
<td>LOC-FNC code</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Y</td>
<td>4-4</td>
<td>Contains $x \cdot n$, the product of the dummy argument, $x$, of EQUATION times the hub-to-tip ratio, $n$</td>
</tr>
<tr>
<td>YIM</td>
<td>10-4</td>
<td>Array of dimension 20 which is required in calling subroutine ROCABES</td>
</tr>
<tr>
<td>YRE</td>
<td>10-4</td>
<td>Array of dimension 20 which is required in calling subroutine ROCABES</td>
</tr>
<tr>
<td>ZERO</td>
<td>3-4</td>
<td>Contains a zero of equation (5)</td>
</tr>
<tr>
<td>ZEROS</td>
<td>3-6,13-4</td>
<td>Subroutine which calculates zeros of equation (4)</td>
</tr>
</tbody>
</table>
| 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:

\[
A_{mn}^i = \left\{ \text{CONSTANT} \right\} \sum_{j=1}^{\text{NSUB}} \left\{ \text{AVERAGE OF NON-OSCILLATORY} \right\} \int_{a_j}^{b_j} \left\{ \text{OSCILLATORY} \right\} dp
\]

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:

\[
A_{mn\sigma}^\pm = \left\{ \text{CONSTANT} \right\} \sum_{j=1}^{N_{\text{SUB}}} \left\{ \text{AVERAGE OF NON-OSCILLATORY} \right\} \int_{a_j}^{b_j} \left\{ \text{OSCILLATORY FACTOR} \right\} d\rho
\]

\[
\left\{ \text{CONSTANT} \right\} = \frac{-1}{\beta_{mn\sigma}} \frac{N_1 N_2}{8} e^{-iK_{mn\sigma}Z_2}
\]

\[
\left\{ \text{NON-OSCILLATORY FACTOR} \right\} = C_1 C_2 \left( \frac{dC_1}{d\alpha} \right)_2 M \lambda E \left( \frac{\sin\theta}{\rho \cos\psi} \right) \text{CAPLT}
\]

\[
\star \lambda_\alpha \Lambda \left\{ \frac{m \phi e}{\sigma} \right\} + K_{mn\sigma}^\pm e^z
\]

\[
\left\{ \text{OSCILLATORY FACTOR} \right\} = e^{i q N_1 \theta} R_{m} \left( \mu_{mn}^\rho \right) e^{i q \frac{d \sin\psi}{\rho \cos\psi}}
\]

See the FORTRAN dictionary (sec. 2.2) for CAPLT.
Method: The procedure is as follows:

1) Set the phase angle, $\Theta_R$, 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
ARMSLC, MAXDIM, MAXJ, AR, NDIM, NDIM, ARMUNN, NOFM, MUSE, MAXN, ALPHANN, IERROR

ENTER

CALCULATE THE PHASE ANGLE
PHISBIR

OBTAIN OR CALCULATE THE EIGEN VALUE GENERATION PARAMETERS
ILOGIC, EGNBND, ETA, MDIM, NDIM, RNOFSV, SGNRB

ARE EIGENVALUES INPUT
YES

CALCULATE THE EIGEN VALUES
NOFM, MUSE, MAXN, ARMUNN

CALL EGNVAL2

CHECK THE ERROR RETURN

COMPARE PRESENT AND PREVIOUS EIGEN VALUE GENERATION PARAMETERS

NO

A
A

IEREGNV:0

= ≠

IEREGNV:2

= ≠

A REDUCED SET OF EIGENVALUES

THERE ARE NO EIGENVALUES

= ≠

IALIZ

IEREGNV:4

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
RNUMN

CALCULATE PROPAGATION FACTORS
BETAMN, CAPKMN

B
CALCULATE EIGEN FUNCTION
NORMALIZATION FACTOR
CAPPHMN

CALCULATE THE CONSTANT TERM IN THE INTEGRAL
FACTCON

INTEGRAL = 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

IF (IAV .EQ. 3) J = 2
AVERAGE VALUE IS INPUT AV (IAV) = AR (2, J, K)

AVERAGE VALUE IS OBTAINED BY INTERPOLATION ALONG SPAN

NSPN = AR (I, J, K)

400 CONTINUE

COMPUTE FLOW ANGLES AND UPDATE NON-OscILLATORY FACTOR

FACTAV = FACTAV * FMU * FMZ * SINBETA

COMPUTE REDUCED FREQUENCY AND LIFT COEFFICIENTS

RMU, B1, B2, B3

CALL LIFLN3

CALL NONCPT

NON-COMPACT OPTION, COMPUTE NON-COMPACT TERM CAPLT

ARMISC (38): =

COMPACT OPTION, COMPUTE LIFT FUNCTION CAPLT

FACTAV = FACTAV * CAPLT

CALCULATE PROJECTION FACTOR TEMP2

D
FACTAV = FACTAV \cdot TEMP2

CALCULATE RELATIVE STREAMWISE DISTANCE TRAVELED BY WAKE

TEMP2

CALCULATE WAKE WIDTH AND DEPTH PRODUCT

TEMP2

FACTAV = FACTAV \cdot TEMP2

CALCULATE WAKE PROFILE FOURIER COEFFICIENT

TEMP2

FACTAV = FACTAV \cdot TEMP2

INTEGRATE OSCILLATORY FACTOR

CALL GAUSS2

INTEGRAL = INTEGRAL + FACTAV \cdot VALUINT

ACCUMULATE PARTIAL SUM

600 CONTINUE
INCLUDE CONSTANT FACTOR
INTEGRAL = INTEGRAL * FACTCON

STORE MODAL AMPLITUDE
ALPHAMN (N,OM) = INTEGRAL

700 CONTINUE

800 CONTINUE

SAVE THE EIGEN VALUE
GENERATION PARAMETERS

RETURN
SUBROUTINE AAAAAA(ARMS1, MAXDIM, MAXJ, ARMO, NDIM, ARMU, N, IMUSE, MAXN, ALPHAMN, IERROR)

REAL MSBT
COMPLEX ALPHAMN(NDIM, MDIM)
COMPLEX FACTAV, FACTON, FACTINT, INTEGRAL, CAPLT, VALUINT

DIMENSION ARMS(C, ARMAXOIM, MAXJ, 3), ARMUMN(NDIM, MDIM), IMUSE(MDIM, MAXN, MDIM)
DIMENSION A(11)

DATA ISORDS, EGND00, ETA0, M, MOD0, N, MOD0, RNOFS0, SIGN0, 0.0, 0.0, 0.0, 0.0 /
DATA PUTWOPI /3.14159265358979, 6.2831330717959 /

COMMON/CFACT/ M, N, ARMS(NDIM, MDIM, ETA, SIGMA, CAPKMN)
COMMON/CFACTIR/ NSBIR, SIGOL, PHIS8IR

EQUIVALENCE (AV(1), AVSP4N), (AV(2), C1), (AV(3), C2), (AV(4), C0),
(1, AV(6), DCL), (AV(7), FMI), (AV(8), FME), (AV(9), FMZ), (AV(5), FM1E)

EXTERNAL FACTINT

ISORDS = ARMS(5)
ITRACE = ARMS(6)

IF(ITRACE .GE. 1) WRITE(6, 1010)
IF(ISORDS .EQ. 1) GO TO 10

10 PHIS8IR = ARMS(13)
SIG = 1.
GO TO 30

20 PHIS8IR = ARMS(12)
SIG = 1.

30 CONTINUE

GENERATE THE EIGENVALUES

NSBIR = ARMS(10)
SIGMA = ARMS(14)
SIGN0 = SIGMA*NSBIR
MSBT = ARMS(7)
RK = SIGN0*MSBT
RKS0D = RK**2

IF(ISORDS .EQ. 1) AXIALM = AR(2, 9, 2)
IF(ISORDS .EQ. 2) AXIALM = AR(2, 9, 3)
CMACH = 1. - AXIALM**2
EGN3ND = RK/ SQRT(CMACH)

IF(ISORDS .EQ. 1) RNOFSV = ARMS(8)
IF(ISORDS .EQ. 2) RNOFSV = ARMS(9)
ETA = ARMS(3)

IF(ITRACE .GE. 1) WRITE(6, 1020) NSBIR, SIGMA, SIGN3, MSBT, RK, RKS0D, 1
AXIALM, CMACH, EGN3ND, RNOFSV, ETA

68
IF(ISOROS .NE. ISOROSQ ) GO TO 110
IF(EGN3ND .NE. EGNBNDO ) GO TO 110
IF(ETA .NE. ETAO ) GO TO 110
IF(MDIM .NE. MDMO ) GO TO 110
IF(NDIM .NE. NDMO ) GO TO 110
IF(RN0FSV .NE. RN0FSVQ ) GO TO 110
IF(SGNRB .NE. SGNR8Q ) GO TO 110
IF(TRACE .GE. 1) WRITE(6,1030) GO TO 120

CALL EGVAI2(ISORSQ,EGN3ND,ETA,MDIM,NDIM,RN0FSV,SGNR8,I0,TRACE,
N0FM,MUSEMAXN,ARMUNM,IEREGNV)

ERROR = IEREGNV
CONTINUE

ERROR RETURN
IF(IEREGNV.EQ.0) GO TO 200
IF(IEREGNV-2) 150,130,150
IF(TRACE .NE.0) WRITE(6,140)
FORMAT('(/1HO,70(1H*)//1HO,*A REDUCED SET OF EIGENVALUES IS AVAILABLE
*COMPUTATIONS WILL PROCEED*/1HO,70(1H*))')
TO 200
IF(IEREGNV-4) 200,160,200
IF(TRACE .NE.0) WRITE(6,180)
FORMAT('(/1HO,70(1H*)//1HO,*THERE ARE NO PROPAGATING RADIAL MODES*/
COMPUTATIONS CAN BE MADE*/1HO,70(1H*))')
TO 1000
CONTINUE

IF(ISOROS.EQ.1) ZSBIR = ARMISC(17)
IF(ISOROS.EQ.2) ZSBIR = ARMISC(16)
IF(ISOROS.EQ.3) ZSBIR = ARMISC(15)
IF(SIGMA.EQ.1) ZSBIR = ARMISC(13)
IF(SIGMA.EQ.2) ZSBIR = ARMISC(14)
IF(ISOROS.EQ.1 .AND. I0FM.GT.1) WRITE(6,1005)

LOOP ON M

DO 30C I0FM=1,N0FM
SET M, L, AND N0FN

M = MUSE(I0FM)
L = (M-SIGMA)/RN0FSV
SIGD = L
IF(ISORSQ.EQ.2) SIGD = SIGMA
QSBIR = QS3B3
IF(ITRACE .GT. 1) QSBIR = ARMISC(13)
MAXI = MAXI(I0FM)
IF(TRACE .GE. 1.AND.I0FM.GT.1) WRITE(6,1005)

69
IF (ITRACE .GE. 1) WRITE (6,1040) M,SIGOL,NSBIR,NOFN
LOOP ON N
DO 700 N=1,NOFN
   
   CALCULATE PROPAGATION FACTORS
   RMUMN = ARMUMN(N,IOF)
   BETAMN = SORT(RKSOO-CMACH*RMUMN**2)
   CAPKMN = (-RK*AXIALM + IFLOW*BETAMN)/CMACH
   CAPMN = EGNR1M(N,RUMMN,ETA)
   IF (ITRACE .GE. 1) WRITE (6,1050) N,RMUMN,BETAMN,CAPKMN,CAPMN
   
   CALCULATE CONSTANT FACTOR, FACTCON
   ARGEXP = -CAPKMN*ZSBIR
   FACTCON = (-.125*NSBRB*RNOSV/BETAMN)*ICMPLX(COS(ARGEXP),SIN(ARGEXP))
   
   SET THE NUMBER OF SUB-INTERVALS FOR WHICH THE LAST TWO TERMS ARE EVALUATED
   IKGGS = 2
   INTEGRAL = (0.,0.)
   
   SET NUMBER OSCILLATIONS
   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
   LOOP ON NUMBER OF SUBINTERVALS
   DO 500 ITRVL=1,NOSCE
   RLOW = ETA + (ITRVL-1)*WIDTHI
   RUP = RLOW + WIDTHI
   270 CONTINUE
   
   EVALUATE TERM TO BE AVERAGED
   SET AVERAGE SPAN
   AVSPAN = (RLOW + RUP)*.5
AV(1) = AVSPAN

SET K INDEX

K = 3
IF(ISDRS .EQ. 1) < = 2

SET AVERAGE VALUES

DO 40C IAV = 2, 11

SET J INDEX

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. 12) J = 2
IF(IAV .EQ. 5) J = 8

SET K INDEX TO BE USED

KU = K
IF(IAV .EQ. 2) KU = K - 1
IF(IAV .EQ. 4) KU = K - 1
IF(IAV .EQ. 5) KU = K - 1

SET SPAN WHEN J = 1

NSPV = AR(1, J, KU)

AVERAGE VALUE IS INPUT

IF(NSPN) 400, 330, 340
330 AV(IAV) = AR(12, J, KU)
GO TO 400

INTERPOLATE FOR AVERAGE VALUE

340 IPA = 1
CALL MLUP(AVSPAN, AV(IAV), 1, NSPN, NSPN, 1, IPA, AR(3, 1, KU), AR(3, J, KU))
CONTINUE

400 IF(TRACE .GE. 1) WRITE(6, 1070) RLO, RUP, AVSPAN, C1, C2, CD, DCL, FM1E
CALCULATE THE AVERAGE FACTOR, FACTAV

FACTAV = C1 * C2 * DCL
IF(TRACE .GE. 1) WRITE(6, 1030) FACTAV

COMPUTE MACH NUMBER RELATED VARIABLES
TEMP1 = SORT( FM1**2 - FMZ**2 )
TEMP2 = SORT( FM1**2 - FMZ**2 )
TEMP3 = .25*( TEMP1 + TEMP2 )**2
TEMP4 = SORT( FM1**2 - FMZ**2 )
FMM = SORT( FMZ**2 + TEMP3 )
COSTHS = FMZ/FMM
SINTHS = SORT( 1.-COSTHS**2 )
COSTHS = SIGN*COSTHS
SIN3ETA = FMZ* TEMP1 + TEMP4 )/( FMM*FM1E )
COSP31 = FMZ/FM1E
COT3ETA = (FMZ-TEMP1*TEMP4/FMZ)/( TEMP1 + TEMP4 )

UPDATE AVERAGE FACTOR

TEMP3 = SINEA/(AVSPAN*COSP3I)
FACTAV = FACTAV+FMM*FM1E*TEMP3
IF( ITRACE.GE.1) WRITE(6,1085)TEMP1,TEMP2,TEMP3,FMM,COSTHS,
SINETA,COSP31,FACTAV

COMPUTE THE REDUCED FREQUENCY

TEMP3 = SINEA/FMM
RNU = .5*SIGOL*NSBIR*C2*TEMP3
IF( MISC. (38).NE.0.) GO TO 410
CALL LIFTFN3(RNU,81,82,83,FACTAV)
GO TO 420

COMPUTE COMPACT OPTION - NAUMANN-YEH

IF( 4RMISC(38).NE.0.) GO TO 410
CALL LIFTFN3(RNU,81,82,83,CAPLT)
GO TO 420

COMPUTE NON-COMPACT OPTION

CALL NONCPT(81,82,83,82,CAPKMN,COSTHS,M,AVSPAN,RNU,SINTHS,CAPLT)

UPDATE AVERAGE FACTOR

FACTAV = FACTAV*CAPLT
IF( ITRACE.GE.1) WRITE(6,1090)CAPLT,FACTAV

X = (DSPAC-.25*C2*FMZ/FMU/COS3I*J1
TEMP1 = 1./ (X/C1 -0.2)
TEMP2 = 1.65*COSORT( TEMP1-0.15*TEMP1**2)
FACTAV = FACTAV*TEMP2

IF(ITRACE .GE. 1) WRITE(6,1110) TEMP1,TEMP2,FACTAV

TEMP1 = 1.36*MS21R*C1/(2*PI*AVSPAN*COSPSI)
TEMP1 = TEMP1*SQR(TD*(X/C1-0.35))*SIGOL
IF(TEMP1.NE.1.) GO TO 450
TEMP2 = .5
GO TO 450

TEMP2 = PI*TEMP1
IF(TEMP2.EQ.0.) TEMP2 = 1.
IF(TEMP2.NE.0.) TEMP2 = SIN(TEMP2)/TEMP2
TEMP2 = TEMP2/(1.-TEMP1**2)

FACTAV = FACTAV*TEMP2

IF(ITRACE .GE. 1) WRITE(6,1110) TEMP1,TEMP2,FACTAV

PERFORM GAUSSIAN INTEGRATION

CALL GAUSS2(RLOW, RUP, IQROS, VALUINT, FACTINT, ARMISC, MAXDIM, MAXJAR)

ACCUMULATE THE TERMS

INTEGRAL = INTEGRAL + FACTAV*VALUINT

END INTERVAL LOOP

IF(ITRACE .GE. 1) WRITE(6,1120) INTEGRAL, VALUINT, FACTAV

600 CONTINUE

APPLY FIRST TERM AND STORE

INTEGRAL = FACTCOM*INTEGRAL
ALP+AMN(N)/FOF = INTEGRAL
IF(ITRACE .GE. 1) WRITE(6,1130) INTEGRAL

END N AND M LOOPS

700 CONTINUE
800 CONTINUE
1000 CONTINUE

SAVE THE EIGENVALUE DETERMINING PARAMETERS

ISORO5O = ISOROS
EGN33CO = EGNAND
ETA3 = ETA
MDIMO = MDIM
NOIMO = NOIM
RMFSV = RMFSV
SIGNR8O = SIGN8B

RETURN
1005 FORMAT(1H1)
1010 FORMAT(1H1,* OPTIONAL PRINTOUT FROM SUBROUTINE AAAAA*)
1020 FORMAT(1H1,* EIGENVALUE PARAMETERS GENERATED*/ 1H ,2X,NSBRR = *,
  112X,10X,*SIGMA = *,F3.0,X,*SIGNB = *,F5.0,7X,* MS J = *,F10.4,
  22X,*RK = *,F10.4,2X,*KQSD = *,F10.4,1H ,2X,*AXI4LM = *,F10.4,
  32X,*CMACH = *,F10.4,2X,*EGNB4D = *,F10.5,2X,*RNJSFV = *,F10.4,2X,
  4*ETA = *,F10.4)
1030 FORMAT(1H1,* THE EIGENVALUES FROM PREVIOUS CALL TO SUBROUTINE *,
  1*AAAAA ARE REUSED FOR THIS CASE*)
1040 FORMAT(1H0,* M = *,I2,3X,* L = *,12,9X,* SIGOL = *,F0.2,3X,*NSBIRK = ,
  1* = *,12,6X,* NOFN = *,12)
1050 FORMAT(1H0,* V = *,12,3X,* RMMN = *,F10.4,3X,* ETA4V = *,F10.4,3X,
  1*CAPKM = *,F10.4,3X,* CAPMN = *,F10.4)
1060 FORMAT(1H0,* FACTON = *,2F10.4,5X,* NOSCE = *,12)
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,* FY1 = *,F9.4,3X,* FME = *,F9.4,6X,* FMZ = *,F9.4,
  311X,*AV(10) = *,F9.4,2X,*AV(11) = *,F9.4,2X,*FM1E = *,F9.4)
1080 FORMAT(1H0,* INTEGRAL = *,2E12.4 )
1090 FORMAT(1H0,* CAPLT = *,2E12.4,3X,*FACTAV = *,2E12.4)*
4.00 FORMAT(1H0,* TEMP1 = *,F9.4,3X,* TEMP2 = *,F9.4,3X,* TEMP3 = *,F9.4,
  1 1X,* FMY = *,F9.4,2X,* COSTHS = *,F9.4,3X,* SINBTA = *,F9.4,
  2 1X,* COSPSI = *,F9.4,3X,* FACTAV = *,2E12.4)*
1100 FORMAT(1H0,* TEMP1 = *,F10.4,3X,* TEMP2 = *,F10.4,3X,* TEMP3 = *,
  1 12E12.4)*
1110 FORMAT(1H0,* INTEGRAL = *,2E12.4,4X,* VALUANT = *,2E12.4,4X,
  1* FACTAV = *,2E12.4)*
1120 FORMAT(1H0,* INTEGRAL = *,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:

\[
\Lambda_{mn0}^+ = \left\{ \text{CONSTANT} \right\} \sum_{j=1}^{N_{SUB}} \int_{a_j}^{b_j} \left\{ \text{OSCILLATORY} \right\} d\rho
\]

with

\[
\left\{ \text{CONSTANT} \right\} = -\frac{2\pi}{2\beta_{mn0}} N_{K1} \frac{z}{K1} e^{-iK_{mn0}^+ z}
\]

\[
\left\{ \text{OSCILLATORY} \right\} = M_{M,K1(\rho)} C_{M,K2(\rho)} a_{\kappa,K1(\rho)} h_{K2(\rho)} \left( \frac{dC_L}{d\alpha} \right) \frac{K1}{2\pi}
\]

\[
\times \left( \frac{m e^0}{\rho} + \kappa_{mn0}^+ z \right) \left( e^{-i\kappa_{mn0}^+ z} \right) \left( e^{iK1(\rho)} \right) \left( \frac{d\alpha}{d\rho} \right) \left( e^{iK1(\rho)} \right)
\]

75
Method: The procedure is as follows:

1) Set the parameters to $K_1$, $K_2$, $N_{K_1}$, and $N_{K_2}$

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, ALPHANN, IERROR

ENTER

CALCULATE PARAMETERS
K1, K2, NK1, NK2

OBTAIN OR CALCULATE THE EIGENVALUE GENERATION PARAMETERS
ILOGIC, EGNBND, ETA, MDIM, NDIM, RSBNKI, SIGNKI

COMPARE PRESENT AND PREVIOUS EIGENVALUE GENERATION PARAMETERS

ARE EIGENVALUES INPUT?

YES

CALCULATE THE EIGENVALUES
CALL EGNVAL2

CHECK THE ERROR RETURN

A
A

\text{IEREGNV : 0}

\text{IEREGNV : 2}

\text{A REDUCED SET OF EIGENVALUES}

\text{1000}

\text{THERE ARE NO EIGENVALUES}

\text{IEREGNV : 4}

\text{LOOP ON M INDEX}
\text{DO 800 IOFM = 1, NOFM}

\text{CALCULATE INDICES}
\text{M, L, SIGOL, NOFN, NSBIR}

\text{LOOP ON N}
\text{DO 700 N = 1, NOFN}

\text{SET THE EIGEN VALUE}
\text{RMUNN}

\text{CALCULATE PROPAGATION FACTORS}
\text{BETANM, CAPKMN}

B
CALCULATE EIGENFUNCTION
NORMALIZATION FACTOR
CAPNNM
CALL EGNORM

CALCULATE THE CONSTANT TERM
IN THE INTEGRAL
FACTCON

INTEGRAL = 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
INTEGRATE OSCILLATORY FACTOR
VALUINT

CALL GAUSS 2

ACCUMULATE PARTIAL SUM
INTEGRAL = INTEGRAL + VALUINT

600 CONTINUE

INCLUDE CONSTANT FACTOR
INTEGRAL = INTEGRAL + FACTCON

STORE MODAL AMPLITUDE
ALPHANN (N,1OFM) = INTEGRAL

700 CONTINUE

800 CONTINUE

SAVE THE EIGEN VALUES
GENERATION PARAMETERS

RETURN

SAVE THE EIGEN VALUE
GENERATION PARAMETERS
SUBROUTINE AABA (ARMISC, MAXDIM, MAXJ, ARMUNN, INOF, PUSE, MAXN, ALPHAN, IERROR)

REAL MSBT
COMPLEX ALPHAN (NDIM, MOIM)
COMPLEX FACTCON, FACTIN2, INTEGRAL, VALUEINT

DIMENSION ARMISC (21), ARM (MAXDIM, MAXJ, 3), ARMUNN (4), IMRIM, MDIM, MOIM,

DATA ILOGICO, EGNBND, ETAO, MDIM0, MDIMO, RSBNKID, SIGNK10 /
C. -l., -l., 0., 0., 0., 0.1
DATA F1, TWOPI / 3.1415926535897926283185307179591/

EXTERNAL FACTIN2

IAERO = ARMISC (18)
ISOROS = ARMISC (5)
ILOGIC = ISOROS

IF (IAERO.EQ.-1.0 .AND. ISOROS.EQ.2.0) ILOGIC = 1
IF (IAERO.EQ.-1.0 .AND. ISOROS.EQ.1.0) ILOGIC = 2
INDEX = ILOGIC + IAERO

ITRACE = ARMISC (6)

IF (ITRACE.GE.-1.0) WRITE (6, 1010)

COMPUTE K1, K2, NK2, B

K1 = ILOGIC + 1

IF (INDEX.EQ.1) K1 = 1
K2 = K1 - IAERO
NK2 = ARMISC (10)

IF (INDEX.EQ.2) NK2 = ARMISC (8)

IF (INDEX.EQ.0) NK2 = ARMISC (9)

NK1 = ARMISC (10)

IF (INDEX.EQ.3) NK1 = ARMISC (9)

IF (INDEX.EQ.0) 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

GENERATE THE EIGENVALUES

IF (ILOGIC.EQ.1) NSBNK1 = NK1

IF (ILOGIC.EQ.2) NSB = NK1

SIGMA = ARMISC (1)

SIGNK1 = SIGMA * NSBNK1

MSBT = ARMISC (7)

ERRORT
RK = SIGNKI*MSBT
RKSQD = RK**2
AXIALM = AR(2,9,K1)
CMACH = 1.-AXIALM**2
EGN3ND = RK/SORTIC(MACH)
IF(ILOGIC.EQ.1) RS3NK1 = NK2
IF(ILOGIC.EQ.2) RS3NK1 = NK1
ETA = ARMISC(3)
IF(ITRACE .GE. 1) WRITE(6,1020) NSBNKI, SIGMA, SIGNKI, MSBT, RK, RKSQD,
NI, AXIALM, CMACH, EGN4ND, RS3NK1, ETA

IF(ILOGIC .NE. ILOGIC0) GO TO 110
IF(EGN3ND .NE. EGN3NOD) GO TO 110
IF(ETA .NE. ETA0) GO TO 110
IF(MDIM .NE. MDIM0) GO TO 110
IF(MDIM .NE. MDIM0) GO TO 110
IF(RS3NK1 .NE. RS3NK10) GO TO 110
IF(SIGNKI .NE. SIGNKI0) GO TO 110
IF(ITRACE .GE. 1) WRITE(6,1030)
GO TO 120
CALL EGNVAL2(ILOGIC, EGN3ND, ETA, MDIM, NDIM, RS3NK1, SIGNKI, ITRACE,
NOM, MUSE, MAXM, ARMUSN, IERENGV)

ERROR RETURN
IF(IERENGV .EQ. 0) GO TO 200
IF(IERENGV .LT. 2) 150,130,150
IF(ITRACE .GT. 0) WRITE(6,140)
10 FORMAT(/1HO,70,1H*1/1HO,**A REDUCED SET OF EIGENVALUES IS AVAILABLE
1LE*/1HO,**COMPUTATIONS WILL PROCEED**/1HO,70(1H*1)**)
GO TO 200
50 IF(IS3NBK1 = 4) 200,150,200
100 FORMAT(/1HO,70) WRITE(6,180)
110 FORMAT(/1HO,70(1H*1)*/1HO,**THERE ARE NO PROPAGATING RADIAL MODES**/
11HO,**NO COMPUTATIONS CAN BE MADE**/1HO,*(1H*1)**)
GO TO 100
200 CONTINUE

ZSBR1 = ARMISC(17)
IF(IINDX .EQ. 3) ZSBR1 = ARMISC(15)
IF(IINDX .EQ. 1) ZSBR1 = ARMISC(15)
IF(ILOW = ARMISC(4)

LOOP ON M
DO 305 IGF = 1,NCFM

SET M, L, AND NOFN

84
M = MUSE(IOFM)
L = (M - SIGN(I)) / RSBYKI
SIGDL = L
IF(ilogic.eq.2) SIGDL = SIGMA
NOFN = MAX(N(IOFM))
IF(ITRACE.GE.1.AND.IOFM.GT.11) WRITE(6,1005)
IF(ITRACE.GE.1) WRITE(6,1040) M, L, SIGDL, NOFN

COMPUTE ALL THE C VALUES

C7 = SIGN(I, SIGDL)
C11 = -C7*FLOAT(AERO)
C6 = 2.*FLOAT(ilogic) - 3.
C3 = (2.*FLOAT(ISOR3S) - 3.)*(-C7)
C12 = C7
C13 = C7
C9 = C3*C7
C14 = C9
C3 = -C3
IF(ITRACE.GE.1) WRITE(6,1045) C3, C6, C7, C8, C9, C11, C12, C13, C14
LOOP ON N
DO 70 C N = 1, NOFN

CALCULATE PROPAGATION FACTORS

RMUMN = ARMUMN(N, IOFM)
BETAMN = SQRT(RKSOQD*CMACH*RMMN**2)
CAPMN = (-RK*AXIALM + IFLO* BETAMN)/CMACH
CAPMN = EGNORM(M, RMUMN, ETA)
IF(ITRACE.GE.1) WRITE(6,1050) N, RMUMN, BETAMN, CAPMN, CAPMN

COMPUTE MODAL AMPLITUDES

CALCULATE CONSTANT FACTOR, FACTCON

ARGEXP = -CAPMN*ZSBIX
FACTCON = (-25*MK1* BETAMN)*CMPLX(COS(ARGEXP), SIN(ARGEXP))
1 * (TWOPI)

SET THE NUMBER OF SUB-INTERVALS FOR WHICH THE LAST TWO TERMS ARE EVALUATED

SURAGE = 2
INTEGRAL = (0, J)

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

LOOP ON NUMBER OF SUBINTERVALS

DO 500 ITRVL=1,NOSCE
RLOW = ETA + (ITRVL-1)*WIDTH
RUP = RLOW + WIDTH
CONTINUE
IF(TRACE .GE. 1) WRITE(6,1070) RLOW,RUP

PERFORM GAUSSIAN INTEGRATION

IF(TRACE.EQ.3) WRITE(6,1115)
CALL GAUSS2(RLOW,RUP,IORDGS,VALUINT,FACTIN2,ARMISC,MAXDIM,MAXJAR)

ACCUMULATE THE TERMS

INTEGR = INTEGR + VALUINT

END INTERVAL LOOP

IF(TRACE .GE. 1) WRITE(6,1120) INTEGR,VALUINT
CONTINUE

APPLY FIRST TERM AND STORE

INTEGR = FACTCON*INTEGR
ALPHAM(N,IDFM)=INTEGR
IF(TRACE .GE. 1) WRITE(6,1130) INTEGR

END N AND M LOOPS

CONTINUE
CONTINUE
1000 CONTINUE

SAVE THE EIGENVALUE DETERMINING PARAMETERS

ILGICO = ILOGIC
EGN3NDO = EGNBND
ETAD = ETA
MD10 = MDIM
NDIM = NDIM
RS3V4K1 = RS3YKI
SIGNK10 = SIGNKI
C RETURN

C
1005 FORMAT(1H1)
1010 FORMAT(1HI,* OPTIONAL PRINTOUT FROM SUBROUTINE AABA4*)
1015 FORMAT(1HG,* K = *12,3X,*K2 = *12,6X,*NK1 = *13,6X,*NK2 = *
113,3X,*3 = *F10.4)
1020 FORMAT(1HO,* EIGENVALUE PARAMETERS GENERATED*/2X,*NSBNKI = *
112,10X,*SIGMA = *F3.0,9X,*SIGNKI = *F5.7X,* MSBT = *F10.4*
22X,*PK = *F10.4,2X,*AKSOD = *F10.4,1H,2X,*XIAMB = *F10.4*
32X,*CPACH = *F10.4,2X,*EGBNBD = *F10.5,2X,*RS3NK1 = *F10.4,2X*
4*ETA = *F10.4)
1030 FORMAT(1HO,* THE EIGENVALUES FROM PREVIOUS CALL TO SUBROUTINE *)
1040 FORMAT(1HO,* L*4.4.3X,*N = *12,9X,*SIGDL = *F6.2*
1DX,*NOFN = *12)
1045 FORMAT(111X,*C3 = *F2.0,2X,*C6 = *F2.0,2X,*C9 = *F2.0,2X,*C12 = *
1F2.0,2X,*C15 = *F2.0,2X)
1050 FORMAT(1HD,* N = *14,3X,*RMJHN = *F10.4,3X,*BETAMN = *F10.4,3X*
1*CAPKNM = *F10.4,3X,*CAPNMN = *F10.4)
1055 FORMAT(1HO,* FACTCON = *F20.5,5X,*NJSCS = *F12)
1060 FORMAT(1HO,* RLOW = *F9.4,3X,*RUP = *F9.4)
1115 FORMAT(1HO,* RHO*3X,*MMK1*3X,*GAMMA*6X,*4.4,9X,*CAPHRHO*9X*
112X,*EXPRHR*12X,*CAPRHR*9X,*FACT*4X,*SCPTRMN*9X,*FACTNR*1)
1120 FORMAT(1HO,* INTEGRL = *F2E12.4,4X,*VALUINT = *F2E12.4*
1130 FORMAT(1HG,* INTEGRL = *F2E12.4)
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:

\[
A_{mn0}^{\pm} = \sum_{j=1}^{N_{SUB}} \left\{ \text{CONSTANT} \right\}_{j} \left\{ \text{AVERAGE OF NON-OSCILLATORY} \right\}_{j} \int_{a_{j}}^{b_{j}} \left\{ \text{OSCILLATORY FACTOR} \right\}_{j} d\rho
\]

with

\[
\left\{ \text{CONSTANT} \right\} = \frac{-N_{R}}{4\beta_{mn}} e^{-iK_{mn}^{\pm} Z}
\]

\[
\left\{ \text{NON-OSCILLATORY FACTOR} \right\}_{j} = C_{2} \left( \frac{dC_{j}}{d\alpha} \right) M_{m}^{M} M_{z} \sin \beta
\]

\[
* \left\{ \frac{m_{e} \phi}{p} + K_{mn}^{\pm} e_{Z} \right\} \text{CAPLT}
\]

\[
\left\{ \text{OSCILLATORY FACTOR} \right\} = W_{\kappa}(\rho) \theta_{m}^{\kappa} \left( \upsilon_{mn} \phi \right)
\]

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
AR, MDIM, NDIM, ARMUMN, NOFM, MUSE, MAXN, ALPHAMN, IERROR

ENTER

CALL 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

CALL CALCULATE THE EIGEN VALUES
NOFM, MUSE, MAXN, ARMUMN

CALL EGNVAL2

CHECK THE ERROR RETURN
CALCULATE PROPAGATION FACTORS

A

IEREGNV: 0


IEREGNV: 2


A REDUCED SET OF EIGENVALUES

THERE ARE NO EIGENVALUES

IEREGNV: 4


LOOP ON M INDEX
DO 800 IOFM = 1, NOFM

CALCULATE INDICES
ML, NSBIR, NOFN

LOOP ON N
DO 700 N = 1, NOFN

SET THE EIGEN VALUE
RMIN

CALCULATE PROPAGATION FACTORS
BETANN, CAPXNN

B
CALCULATE EIGEN FUNCTION NORMALIZATION FACTOR
CALL CAPNMN

CALCULATE THE CONSTANT TERM IN THE INTEGRAL
FACTCON

INTEGRAL = 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

KU = K

NSPN = AR (I, J, KU)

CALL MTLUP

AVERAGE VALUE IS INPUT
AV (IAV) = AR (I, J, KU)

AVERAGE VALUE IS OBTAINED
BY INTERPOLATION
ALONG SPAN

= NSPN: 0

400 CONTINUE

INITIALIZE THE NON OSCILLATORY FACTOR
FACTAV = C2 • DCL

COMPUTE FLOW ANGLES AND UPDATE
NON-OSSCILLATORY FACTOR
FACTAV = FACTAV • FMM • FMZ • SINBETA

COMPUTE REDUCED FREQUENCY
AND LIFT COEFFICIENTS
RNU, B1, B2, B3

NON-COMPCT

CALL

COMPACT OPTION,
COMPUTE NON-COMPACT TERM
CAPLT

NON-COMPCT OPTION,
COMPUTE NON-COMPACT TERM
CAPLT

LIFTFN3

COMPACT OPTION,
COMPUTE LIFT FUNCTION
CAPLT

D

CALL

95
FACTAV FACTAV CAPLT
CALCULATE PROJECTION FACTOR TEMP2
FACTAV FACTAV TEMP2
INTEGRATE OSCILLATORY FACTOR CALL GAUSS2 VALUINT
ACCUMULATE PARTIAL SUM INTEGRAL = INTEGRAL + FACTAV • VALUINT
600 CONTINUE
INCLUDE CONSTANT FACTOR INTEGRAL = INTEGRAL • FACTCON
STORE MODAL AMPLITUDE ALPHAMN (N, IOM) = INTEGRAL
700 CONTINUE
800 CONTINUE
SAVE THE EIGEN VALUE GENERATION PARAMETERS
RETURN
SUBROUTINE BCDAAURMI (ARMS, MAXD, MAXJ, AR, MDIM, NDIM, ARMN, NOFM, MUSE, IMAX, ALPHAM, IERROR)

REAL MSBT
COMPLEX ALPHAM (NDIM, MDIM)
COMPLEX FACTAV, FACTCON, FACTIN3, INTEGR, CAPLT, VALIYVT

DIMENSION ARMISC (1), AR (MAXD, MAXJ, 3), ARMN (NDIM, MDIM), IMUSE (MDIM), MAXN (MDIM)
DIMENSION AV (11)

DATA EGNBND3, ETAO, MDI KO, NOI MO, RNOFSV0, SIGNR30/
  I -1.0, -1.0, 0.0, 0.0 /
DATA PI, TRFOP / 3.14159265358979, 0.28318530717959 /

COMMON/FACTS/ MNRHUMN, CAPMN, ETA, SGN, LCAKYN

EQUIVALENCE (AV (1) » AVSPAN) / (AV (3), C2),
1 (AV (6), DCL) » (AV (7), F9) » (AV (8), FNE) » (AV (9), FMZ)

EXTERNAL FACTIN3

ITRACF = ARMS (6)
IF (ITRACF .GE. 1) WRITE (6, 1010)

GENERATE THE EIGENVALUES

NSBRB = ARMS (10)
SIGMA = ARMS (14)
SIGVRB = SIGMA * NSBRB
MSBT = ARMS (7)
RK = SIGVRB * MSBT
RK3D = RK ** 2
AXIALM = AR (2.9, 2)
CMACH = 1. - AXIALM ** 2
EGN3NC = RK / SORT (CMACH)
RNOFSV = 1
ETA = ARMS (3)
IF (ITRACE .GE. 1) WRITE (6, 1020) NSBRB, SIGMA, SIGNR3, MSBT, RK, RK3D, I
IF (EGNBN3 .NE. EGNBND3) GO TO 110
IF (ETA .NE. ETAO) GO TO 110
IF (MDIM .NE. MDI KO) GO TO 110
IF (NDIM .NE. NDI MO) GO TO 110
IF (RNOFSV .NE. RNOFSV0) GO TO 110
IF (SIGVRB .NE. SIGNR3) GO TO 110
IF (ITRACF .GE. 1) WRITE (6, 1030)
GO TO 120

110 CALL EGVALZ1 2 (EGNBN3, ETA, MDIM, NDIM, RNOFSV, SIGNR3, ITRACE, I
  NOFM, MUSE, MAXN, ARMHUMN, IERROR)
IER = IEREGNV
120 CONTINUE
ERROR RETURN
IF(IEREGNV.EQ.0) GO TO 200
IF(IEREGNV.EQ.-2) 150,130,150
130 IF(TRACE.NE.0) WRITE(6,140)
140 FORMAT(/1HO,70(1H*)//1HO,*A REDUCED SET OF EIGENVALUES IS AVAILABLE
10*1HO,*COMPUTATIONS WILL PROCEED*//1HO,70(1H*) )
GO TO 230
150 IF(IEREGNV.EQ.-1) 200,160,200
160 IF(TRACE.NE.0) WRITE(6,180)
180 FORMAT(/1HO,70(1H*)//1HO,* THERE ARE NO PROPAGATING RADIAL MODES*
11HO,* NO COMPUTATIONS CAN BE MADE*//1HO,70(1H*) )
GO TO 1000
200 CONTINUE

ZSBIR = ARMSC(17)
IFLOW=ARMSC(14)

LOOP ON M
DO 906 IOFM=1,NOFM
SET M, L, AND NOFN
M = MUSE(IOFM)
L = (M-SIGNR8)/RNDFS
NOFN = MAXN(IOFM)
IF(IOFM.GT.1.AND.ITRACE.GE.1) WRITE(6,1005)
IF(TRACE.GE.1) WRITE(6,1040) M,L,NSBR3,NJFN

LOOP ON N
DO 700 N=1,NOFN
CALCULATE PROPAGATION FACTORS
RMUMN = ARUMMN(N,IOFM)
BETAMN = SQRT(RSQD-CMACH*RINJMN**2)
CAPKN = (-RK*AXIALM + IFLOW*BETAMN)/CMACH
CAPKN = EGNN3RKM(RMUMN*ETA)
IF(TRACE.GE.1) WRITE(6,1350) N,RMUMN,BETAMN,CAPKN,CAPKNM

COMPUTE MODAL AMPLITUDES
CALCULATE CONSTANT FACTOR, FACTCON

\[
\text{TEMP1} = -\text{CAPKM} \times \text{ZSBIR} \\
\text{FACTCON} = -0.25 \times \text{NSBRB}/(\text{BETAMN}) \times (1 - \cos(\text{TEMP1}) \times \sin(\text{TEMP1})) \\
\]

SET THE NUMBER OF SUB-INTERVALS FOR WHICH THE LAST TWO TERMS ARE EVALUATED

IDRGS = 2
INTEGRAL = (0., 0.)

SET NUMBER OSCILLATIONS

NOSCE = 1
NOSCE = MAXO(NOSCE, N)
NOSCE = 1.5 * NOSCE
NOSCE = MAXO(NOSCE, 2)
WIDTHI = (1. - ETA)/NOSCE
IF(TRACE .GE. 1) WRITE(5, 1360) FACTCON, NOSCE

LOOP ON NUMBER OF SUBINTERVALS

DO 50C ITRVL=1, NOSCE
RLOW = ETA + (ITRVL-1) * WIDTHI
RUP = RLOW + WIDTHI
:70 CONTINUE

EVALUATE TERM TO BE AVERAGED

SET AVERAGE SPAN

AVSPAN = (RLOW * RUP) * .5
AV(1) = AVSPAN

SET K INDEX

K = 2

SET AVERAGE VALUES

DO 40C IAV=3, 11
SET J INDEX

J = IAV
IF(IAV.EQ.3) J=2
IF(IAV.GT.3) J = 2 + 2 * INT((19*K) + 1)
IF(J.GT.5) GO TO 400
IF(J.EQ.4) GO TO 400

99
SET K INDEX TO BE USED

KU = K

SET SPAN WHEN J=1

NSPN = AR(1,J,KU)

AVERAGE VALUE IS INPUT

IF( NSPN ) 400,330,340
330 AV(IAV) = AR(2,J,KU)
GO TO 400

INTERPOLATE FOR AVERAGE VALUE

J=0 IPA=-1
CALL MLUP(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,FM1,
               FME,FMZ,AV(10),AV(11)

CALCULATE THE AVERAGE FACTOR, FACTAV

FACTAV = C2*DCL
IF(ITRACE .GE. 1) WRITE(6,1080) FACTAV

COMPUTE MACH NUMBER RELATED VARIABLES

TEMP1 = SORT(FMI**2,FMZ**2)
TEMP2 = SORT(FME**2,FMZ**2)
TEMP3 = 0.25*(TEMP1+TEMP2)**2
FMM = SORT(FMZ**2+TEMP3)
COSTHS = FMZ/FMM
SINH3 = SORT(1.-COSTHS**2)
SIN3ETA = SINH3
COT3ETA = COSTHS/SINH3

UPDATE AVERAGE FACTOR

FACTAV = FACTAV*FMM*FMZ*SIN3ETA
IF(ITRACE .GE. 1) WRITE(6,1085) TEMP1,TEMP2,TEMP3,FMM,COSTHS,
                SIN3ETA,FACTAV

COMPUTE REDUCED FREQUENCY

RNU = .5*C2*453/f**2
B1 = ...
B2 = -AV(11)*COT3ETA
B3 = -AV(10)*COT3ETA
COMPUTE COMPACT OPTION

IF ( ARMISC(38),NE.0.) GO TO 410
CALL LIFTFN3(RNU,RL.0,82,83,CAPLT)
GO TO 420

COMPUTE NON-COMPACT OPTION

410 CONTINUE
CALL NONCP(T,B1,B2,B3,C2,CAPMN,COSTHS,M,AVSPAN,RNU,SINTHS,CAPLT)

UPDATE AVERAGE FACTOR

420 FACTAV = FACTAV*CAPLT
IF ( ITRACE.GE.1 ) WRITE(6,1690) FACTAV,CAPLT

TEMP2 = M*COSTHS/AVSPAN + CAPMN*SINTHS
FACTAV = FACTAV*TEMP2
IF ( ITRACE.GE.1 ) WRITE(6,1160) FACTAV,TEMP2

PERFORM GAUSSIAN INTEGRATION

CALL GAUSS2(LOW,RUP,ORDGS,VALUINT,FACTIN3,ARMISC,MAXDIM,MAXJ,AR)

ACCUMULATE THE TERMS

INTEGRAL = INTEGRAL + FACTAV*VALUINT

END INTERVAL LOOP

IF ( ITRACE.GE.1 ) WRITE(6,1120) FACTAV,VALUINT,INTEGRAL

600 CONTINUE

APPLY FIRST TERM AND STORE

INTEGRAL = FACTCON*INTEGRAL
ALPHA(M,N,DFM) = INTEGRAL
IF ( ITRACE.GE.1 ) WRITE(6,1130) INTEGRAL

END N AND M LOOPS

700 CONTINUE
800 CONTINUE
1000 CONTINUE

SAVE THE EIGENVALUE DETERMINING PARAMETERS

EGN3NCO = EGN3NCO
ETAJ = ETA
MDIMQ = MDIMQ
NDIMQ = NDIMQ
RNOFSVO = RNOFSV
SIGVRBO = SIGRNB

RETJRN

1005 FORMAT(IHI//)
1010 FORMAT(IHI//I1X,* OPTITIONAL PRINTOUT FROM SUBROUTINE BCDAAB*)
1020 FORMAT(I1H0,10X,*EIGENVALUE PARAMETERS GENERATED*/I1X,**NSB8 = **
113 2X,**SIGMA = **F3.2X,**SIGNRB = **F6.02X,**N3T = **F1C.4/I1X,
2*RK = **F10.4,2X,**RKSQ = **F10.4,2X,**AXIAL = **F1J.4/I1X,
3*CMNC = **F1J.4,2X,**EGNBND = **F10.4,2X,**RJFSV = **F10.4/I1X,
4*ETA = **F10.5)
1030 FORMAT(I1H0,10X,*THE EIGENVALUES FROM PREVIOUS CALL TO SUBROUTINE B
1BCAA*/11X,*ARE REUSED FOR THIS CASE*)
1040 FORMAT(I1H0,10X,*M = **I4,2X,*L = **I4,2X,
1*NS3RB = **I13,2X,*NOF = **I14)
1050 FORMAT(I1H0,10X,*Q = **I14,2X,**KMUM = **F1O.4,2X,**BETAMN = **F10.4/
120X,**CAPKM = **F1C.4,2X,**CAPKM = **F10.4)
1060 FORMAT(I1H0,10X,*FACTCN = **F2E12.4,2X,**NOSCE = **I14)
1070 FORMAT(I1H0,10X,*RLow = **F9.4,2X,**RUP = **F9.4,2X,**AVSPAN = **F9.4/
1/11X,*C2 = **F10.4,2X,*DCL = **F1J.4/I1X,
2*FM1 = **F9.4,2X,**FME = **F9.4,2X,**FMZ = **F9.4/I
31IX,*AV11C = **F9.4,2X,**AV111 = **F9.4)
1080 FORMAT(I1H0,10X,*FACTAV = **F2E12.4)
1090 FORMAT(I1H0,10X,*TEMP1 = **F9.4,2X,**TEMP2 = **F9.4,2X,**TEMP3 = **F9.4/
1 1X,* FM = **F9.4,2X,**GOSTHS = **F9.4,2X,**SINBTA = **F9.4/
2 1X,* FACTAV = **F2E12.4)
1090 FORMAT(I1H0,10X,*FACTAV = **F2E12.4,2X,**CAPLT = **F2E12.4)
1090 FORMAT(I1H0,10X,*FACTAV = **F2E12.4,2X,**TEMP2 = **F2E12.4)
120 FORMAT(I1H0,10X,*FACTAV = **F2E12.4,2X,**VALUANT = **F2E12.4/
111X,**INTEGRAL = **F2E12.4)
1130 FORMAT(I1H0,10X,*INTEGRAL = **F2E12.4)

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:

\[
A_{n \sigma}^+ = \left\{ \text{CONSTANT} \right\} \sum_{j=1}^{N_{\text{SUB}}} \left\{ \text{AVERAGE OF NON-OSCILLATORY FACTOR} \right\} \int_{a_j}^{b_j} \left\{ \text{OSCILLATORY FACTOR} \right\} d \rho
\]

with

\[
\left\{ \text{CONSTANT} \right\} = \frac{-N}{4 \beta_{mn}} e^{-iK_{mn}^z} Z \]

\[
\left\{ \text{NON-OSCILLATORY FACTOR} \right\} = C \left( \frac{dC_L}{d \alpha} \right)_j M_z \left\{ \frac{m e_{\phi}}{\rho} + K_{mn}^+ e_z \right\}
\]

\[
\left\{ \text{OSCILLATORY FACTOR} \right\} = \text{FACTIN4}
\]

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: 

IERROE (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

CALL EGNVAL2

CALCULATE THE EIGENVALUES
NOFM, MUSE, MAXN, ARMUMN

CHECK THE ERROR RETURN

A
CALCULATE PROPAGATION FACTORS

A

IEREGNV = 0

#

IEREGNV = 2

#

IEREGNV = 4

A REDUCED SET OF EIGENVALUES

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
CALCULATE EIGEN FUNCTION
NORMALIZATION FACTOR
CAPNMN

CALL EGNNORM

CALCULATE THE CONSTANT TERM IN THE INTEGRAL
FACTCON

INTEGRAL = 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
AVERAGE VALUE IS INPUT
AV(AV)+AR(2, J, KU)

AVERAGE VALUE IS OBTAINED
BY INTERPOLATION
ALONG SPAN

initialize the non oscillatory factor
FACTAV = C2 * DCL

FACTAV = FACTAV * FMZ

calculate projection factor
TEMP2

FACTAV = FACTAV * TEMP2
INTEGRATE OSCILLATORY FACTOR
VALUINT

ACCUMULATE PARTIAL SUM
INTEGRAL = INTEGRAL + FACTAV + VALUINT

600 CONTINUE

INCLUDE CONSTANT FACTOR
INTEGRAL = INTEGRAL + FACTCON

STORE MODAL AMPLITUDE
ALPHAMN (N, IOFM) = INTEGRAL

700 CONTINUE

800 CONTINUE

SAVE THE EIGEN VALUE
GENERATION PARAMETERS

RETURN
SUBROUTINE BBCAA(ARMISC, MAXDIM, MAXJ, AR, MDIM, NDIM, ARMUM, NOFS, MUSE, IMAX, ALPHAM, IERROR)

REAL MSBT
COMPLEX ALPHAM, FACT, FACTAV, FACTCON, FACTIN4, INTEGRL, VALUEN

DIMENSION ARMISC(1), AR(MAXDIM, MAXJ), ARMUM(NDIM, MDIM), IMUSE(MDIM), MAXN(MDIM, MDIM)
DIMENSION AV(I11)

DATA EGNBNO, ETAO, NDIMO, NDIMO, RNDFSVO, SIGNR8O/
1, -1., -1., 0., 0., 0., /
DATA PI, TWOPI /3.14159265358979, 6.28318530717959/

COMMON /FACT/ M, N, ARMUM, CAPNM, ETA, SIGN, L, CAPKYN

EQUIVALENCE (AV(I), AVSPAN), (AV(3), C2),
I(AV(6), DCL), (AV(7), FMI), (AV(8), FME), (AV(9), FMZ)

EXTERNAL FACTIN4

ITRACE = ARMISC(6)
IF(ITRACE .GE. 1) WRITE(6, 1010)

GENERATE THE EIGENVALUES

NSBR8 = ARMISC(10)
SIGMA = ARMISC(14)
SIGNR8 = SIGMA*NSBR8
MSBT = ARMISC(7)
RK = SIGNR8*MSBT
RKSQO = RK**2
AXIALM = AR(2, 9, 2)
CMACH = 1. -AXIALM**2
EGNBNO = RK/SORT(CMACH)
RNDFSV = 1
ETA = ARMISC(3)

IF(ITRACE .GE. 1) WRITE(6, 1020) NSBR8, SIGMA, SIGNR8, MSBT, RK, RKSQO,
AXIALM, CMACH, EGNBNO, RNDFSV, ETA

IF(EGBNO .NE. EGNBNOO) 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(RNDFSV .NE. RNDFSVO) GO TO 110
IF(SIGNR8 .NE. SIGNR8O) GO TO 110
IF(ITRACE .GE. 1) WRITE(6, 1030)
GO TO 120

110 CALL EGVVAL2I2(EGNBNO, ETA, MDIMO, NDIMO, RNDFSVO, SIGNR8, ITRACE,
NOFS, MUSE, MAXN, ARMUM, IERROR)
ERROR = IEREGNV
120 CONTINUE

ERROR RETURN

IF(IEREGNV.EQ.0) GO TO 200
IF(IEREGNV.EQ.2) 130,130,130
130 IF(ITRACE.NE.0) WRITE(6,140)
140 FORMAT(1H3,T0(1H*)//1HO,*A REDUCED SET OF EIGENVALUES IS AVAILAB
LLE*/1HO,*COMPUTATIONS WILL PROCEED*/1HO,70(1H*) )
GO TO 230
150 IF(IEREGNV.EQ.4) 200,160,200
160 IF(ITRACE.NE.0) WRITE(6,160)
170 FORMAT(1H3,T0(1H*)//1HO,*THERE ARE NO PROPAGATING RADIAL MODES*/
11HO,*NO COMPUTATIONS CAN BE MADE*/1HO,70(1H*) )
GO TO 1000
200 CONTINUE

ZSBIR = ARMISC(17)
IFLOW = ARMISC(4)

LOOP ON M
DO 300 I0FM=1,N0FM

SET M, L, AND NOFN
M = MUSE(I0FM)
L = (M-SIGNR8)/RNOFSV
NOFN = MAXN(I0FM)
IF(I0FM.GT.1.AND.ITRACE.GE.1) WRITE(6,1005)
IF(ITRACE.GE.1) WRITE(6,1345) M,L,NSB8R,N0FN

LOOP ON N
DO 700 N=1,N0FN

CALCULATE PROPAGATION FACTORS
RMURN = ARMURN(N,I0FM)
BETARN = SORT(RKSOD-CMACH*RMURN**2)
CAPKMN = (RK*AXAM + IFLOW*BETARN)/CMACH
CAPMN = EGNOXY(N,RSY4N,ETA)
IF(ITRACE.GE.1) WRITE(6,1050) N,RMURN,BETARN,CAPKMN,CAPMN

COMPUTE MODAL AMPLITUDES
CALCULATE CONSTANT FACTOR, FACTCON

TEMP1 = -CAPKMNZSBIR
FACTCON = (-NSBRB*CMPLX(COS(TEMP1), SIN(TEMP1))) / (4*BETAMN)

SET THE NUMBER OF SUB-INTERVALS FOR WHICH THE LAST TWO TERMS ARE EVALUATED

IOR3GS = 2
INTEGRAL = (0, 0.1)

SET NUMBER OSCILLATIONS

NOSCE = 1
NOSCE = MAX(NOSCE, 4)
NOSCE = 1.5 * NOSCE
NOSCE = MAX(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
270 CONTINUE

EVALUATE TERM TO BE AVERAGED

SET AVERAGE SPAN

AVSPAN = (RLOW + RUP) / 2
AV(I) = AVSPAN

SET K INDEX

K = 2

SET AVERAGE VALUES

DO 400 IAV = 3, 11

SET J INDEX

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
SET K INDEX TO BE USED

KU = K

SET SPAN WHEN J = 1

NSPN = AR(1, J, KU)

AVERAGE VALUE IS INPUT

IF (NSPN) 400, 330, 340

330 AV(1AV) = AR(2, J, KU)

GO TO 400

INTERPOLATE FOR AVERAGE VALUE

340 [IPA=1]

CALL MPLUP(AVSPAN, AV(1AV), 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, FM1, FMZ, AV(10), AV(11)

CALCULATE THE AVERAGE FACTOR, FACTAV

FACTAV = C2 * DCL

IF (ITRACE .GE. 1) WRITE (6, 1080) FACTAV

TEMP1 = SQRT( FM1**2 - FMZ**2 )

TEMP2 = SQRT( FM1**2 - FMZ**2 )

TEMP1 = 0.25 * (TEMP1 + TEMP2)**2

TEMP1 = SORT( FMZ**2 + TEMP1 )

FACTAV = FACTAV * FMZ

IF (ITRACE .GE. 1) WRITE (6, 1090) TEMP1, TEMP2, FACTAV

COSTHS = FMZ / TEMP1

SINTHS = SQRT( 1.0 - COSTHS**2 )

TEMP2 = M*COSTHS / AVSPAN + CAPIKNN*SINTHS

FACTAV = FACTAV * TEMP2

IF (ITRACE .GE. 1) WRITE (6, 1100) FACTAV, TEMP2

PERFORM GAUSSIAN INTEGRATION


call gauss2(RLOW, RUP, IQROGS, VALUINT, FACTIN, ARMISC, AXDIM, MAXJ, AR)

ACCUMLATE THE TERMS

INTEGRAL = INTEGRAL + FACTAV*VALUINT

END INTERVAL LOOP

IF (ITRACE .GE. 1) WRITE (6, 120) FACTAV, VALUINT, INTEGRAL

115
600 CONTINUE

APPLY FIRST TERM AND STORE

INTEGRAL = FACTCON*INTEGRAL
ALPHA = (M+20)*INTEGRAL
IF(TRACE .GE. 1) WRITE(6,1130) INTEGRAL

END N AND M LOOPS

700 CONTINUE

800 CONTINUE

1000 CONTINUE

SAVE THE EIGENVALUE DETERMINING PARAMETERS

EGNAND = EGNAND
ETA3 = ETA
MD1O = MD1O
ND4O = ND4O
RNDFSVO = RNDFSVO
SIG4RBO = SIG4RBO

RETURN

1005 FORMAT(10X,* OPTIONAL PRINTOUT FROM SUBROUTINE BCAAA*)
1020 FORMAT(10X,*EIGENVALUE PARAMETERS GENERATED*/13X,*NSBR = *,
113X,*SIGMA = *,F3.2,*X,*SIGNR = *,F10.4/13X,
2*RK = *,F10.4,*X,RKSO = *,F10.4,*X,AXIALM = *,F10.4/13X,
3*MACOH = *,F10.4,*X,EGNAND = *,F10.4,*X,RNDFSVO = *,F10.4/13X,
4*ETA = *,F10.4)
1030 FORMAT(10X,*THE EIGENVALUES FROM PREVIOUS CALL TO SUBROUTINE B
1BCAA*.1X.*ARE REUSED FOR THIS CASE*)
1040 FORMAT(10X,*NSBR = *,I3,*X,*NSBR = *,I3,*X,*NSBR = *,I3,*X
1M3SBR = *,I3,*X,*M3SBR = *,I3,*X,*M3SBR = *,I3)
1050 FORMAT(10X,*M = *,I3,*X,*AMUMN = *,F10.4,*X,*BETAMN = *,F10.4/13X,
120X,*CAPKM = *,F10.4,*X,*CAPKM = *,F10.4/13X)
1060 FORMAT(10X,*FACTCON = *,2E12.4,*X,*NSCE = *,I1)
1070 FORMAT(10X,*RLOW = *,F9.4,*X,*RUP = *,F9.4,*X,*AVSPAN = *,F9.4/13X,
111X,*C2 = *,F10.4,*X,*DCL = *,F10.4/11X,
120X,*FF = *,F9.4,*X,*FMZ = *,F9.4/13X,
311X,*AV(11) = *,F9.4,*X,*AV(11) = *,F9.4)
1080 FORMAT(10X,*FACTAV = *,2E12.4)
1090 FORMAT(10X,*TEMP1 = *,F10.4,*X,*TEMP2 = *,F10.4/13X)
1100 FORMAT(10X,*FACTAV = *,2E12.4,*X,*TEMP2 = *,2E12.4)
1120 FORMAT(10X,*FACTAV = *,2E12.4,*X,*VALUES = *,ZE12.4/13X,
111X,*INTEGRAL = *,2E12.4)
1130 FORMAT(10X,*INTEGRAL = *,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:

\[
P(x) = J'_m(x) - Y'_m(x) \frac{J'_m(\eta x)}{Y'_m(\eta x)}
\]

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 \( \eta \) denoting the hub-to-tip ratio.

**Method:** The procedure is as follows:

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

\[
|m_{\text{max}}| \leq \mu_{m_{\text{max}}}, n \leq E_B
\]

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, $m_{\text{max}}$, to be at most 100 (based on the restriction on the Bessel function evaluation BSSL5, sec. 3.3.6).

3) Calculate the smallest negative $\ell$, $\ell_{\text{lower}}$, 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 $\ell_{\text{lower}} = - \frac{E_B + \sigma N_R}{N_S}$

4) Determine all $m$'s according to the above equation defining $m$ and within the bounds on $m$ and $\ell$ given above by starting with the lowest $\ell$ and stepping through the $\ell$'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, $n_{\text{max}}$, 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 $\pi/(1-n)$. The bound used is

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

7) Calculate the eigenvalues for the $m$'s determined above and $n = 1$ to $n_{\text{max}}$ 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 $\mu_{mn}$'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,ARUMN,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

\[ 2 \times EGNBND \times (1. - ETA) + 2 \]

**Accuracy:**

See subroutine ZEROS for the accuracy of the eigenvalues.
CALLING LIST
LZERO, EGNBND, ETA, MDIN, NDIM, RNDFS, SIGNRB, ITRACE, NOFM, MUSE, MAXN, ARMUNN, IEREGNV

ENTER

CALCULATE THE LARGEST M
MBESEGN

CALCULATE THE LOWEST L
L

LUSE = 0

210

IS LZERO = 1 AND L = 0

YES

CALCULATE m
M

ABS (M) : MBESEGN

M : O

SAVE m
LUSE = LUSE + 1
MUSE (LUSE) = M

225

L = L + 1

LUSE : MDIM

260
SET LARGEST \( n \)

SET NUMBER OF \( m \)

\[ \text{NOFM} = \text{LUSE} \]

\[ \text{NOFM} = 0 \]

ERROR RETURN

\[ \text{IEREGNV} = 4 \]

CALL

CALCULATE THE NUMBER OF EIGEN VALUES WITHIN THE EIGEN VALUE BOUND [\( \text{MAXN (IM)} \)]

390 CONTINUE

LUSE = 0

A

SET ERROR

\[ \text{IEREGNV} = 2 \]

260

SET LARGEST \( n \)

\[ \text{NNAX} \]

CALCULATE EIGEN VALUES

\[ \text{ARNUMN} \]

CALL

ZEROS

LOOP ON \( m \) INDEX

\[ \text{DO 390 IM = 1, NOFM} \]

980
LOOP ON m INDEX
DO 490 IM=1, NOFM

NOFN = MAXN (IM)

\[
\text{IF } \text{NOFN} \leq 0 \text{ THEN}
\]

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

\[
\text{IF } \text{NOFM} \leq 0 \text{ THEN SET ERROR}
\]

IEREGNV = 4

RETURN
SUBROUTINE EGNVAL2ZERO (EGNBNO, ETA, MDIM, NDIM, RNDFSV, SIGNR8, 
   IITRACE, NOFM, MUSE, MAXN, ARNUMN, IEREGNV)

PURPOSE
OBTAINTHEAPPROPRIATE SPINNING AND RADIAL MODES.

INPUT

VARIABLE DEFINITION
EGNBNO ALL EIGENVALUESAREFOUND WHICHDOWTEXCEED 
   THIS NUMBER.
ETA HUB TO TIP RATIO, WHICH IS ZERO FOR CIRCULAR 
   DUCT
MDIM COLUMN DIMENSION OF MATRIX IN WHICH EIGEN-
   VALUES ARE PLACED, PROVIDING A MAXIMUM ON 
   THE NUMBER OF SPINNING MODES.
NDIM ROW DIMENSION OF MATRIX IN WHICH EIGENVALUES 
   ARE PLACED, PROVIDING A MAXIMUM ON THE 
   NUMBER OF RADIAL EIGENVALUES.
RNDFSV NUMBER OF STATOR VANES
SIGNR8 PRODUCT OF HARMONIC INDEX, SIGMA, 
   AND THE NUMBER OF ROTOR BLADES, NS848

OUTPUT

ARNUMN MATRIX OF EIGENVALUES WHERE 
   ARNUMN(IOFM, IOFN) 
   * IS THE EIGENVALUE FOR SPINNING MODE INDEX 
     IOFM AND RADIAL MODE INDEX IOFN WHERE 
     IOFN=1,...,MAXN(IOFM) 
   MAXN ARRAY OF THE NUMBER OF RADIAL MODES 
     WHERE MAXN(IOFM) 
     CORRESPONDS TO MUSE(IOFM), IOFM=1,...,NOFM 
   MUSE ARRAY OF THE NOFM SPINNING MODE INDICES 
   NOFM NUMBER OF SPINNING MODE INDICES

ERROR RETURN IEREGNV  
   0 ALL EIGENVALUES REQUIRED ARE RETURNED 
   2 THERE ARE MORE EIGENVALUES REQUIRED THAN 
      THERE IS SPACE FOR, AS MANY AS POSSIBLE 
      ARE RETURNED 
   4 THERE ARE NO EIGENVALUES

DIMENSION MUSE(MDIM), MAXN(MDIM), ARNUMN(NDIM, MDIM)
DIMENSION SC(40)
DATA MBES/100/

IEREGNV = 0

SET THE MAXIMUM SPINNING MODE THAT CAN POSSIBLY PROPAGATE
MEGN = EGNBND + 1
MBESEG = MINO(MEGN*MSES)

NOW COMPUTE CANDIDATE SPINNING MODES ACCORDING TO

SET THE LOWEST L THAT IS USABLE

L = ABS(1 EGNBND *SIGNRB)/RNDFS
LUSE = L

210 IF (LZERO .EQ. 1 .AND. L.EQ.0 ) GO TO 225
M = SIGNRB *L*RNDFS
IABSM = IABS(M)
IF (IABSM - MSEG ) 220, 220, 215
215 IF (M) 225, 225, 260
220 LUSE = LUSE + I
MUSE(LUSE) = M
225 L = L + 1
IF (LUSE - MDOIM ) 210, 250, 259
250 IEREGNV = 2
260 NOFM = LUSE
300 CONTINUE

CHECK TO BE SURE THERE ARE SPINNING MODES

IF (NOFM ) 310, 310, 320
310 IEREGNV = 4
GO TO 900
320 CONTINUE

OBTAIN THE EIGENVALUES

SET BOUNDS FOR EIGENVALUE CALCULATION

NBESEG = 1 - ETA1*MSES + 1.7/3.14159265 + 1.
NMAX = MINO( NBESEG , NDIM - 1)

CALL ZEROS(ETA, NOFM, MUSE, NMAX, NDIM, ITRACE, SCAR4M)

COMPUTE THE NUMBER OF RADIAL MODES FOR EACH SPINNING MODE

330 DO 390 IM = 1, NOFM
M = MUSE(IM)
M1 = IABS(M) + 1
N = 1
340 IF (ARUMDN(N, IM ) < EGBNDO ) 350, 350, 370
350 N = 'IM + 1
390 CONTINUE
IF( N - NMAX ) 340, 340, 360
360 IEREGNV = 2
370 NMI = N - 1
   MAXN(IM) = NMI
   IF( NMI - NDIM ) 390, 390, 380
380 IEREGNV = 2
   MAXN(IM) = NDIM
390 CONTINUE
C 400 CONTINUE

ELIMINATE SPINNING NODES FOR WHICH THERE ARE NO RADIAL MODES

LUSE = 0
DO 490 IM=1, NDFM
   NDFN=MAXN(IM)
   IF( NDFN ) 490, 490, 410
410 LUSE = LUSE + 1
   MUSE(LUSE) = MUSE(IM)
   MAX(LUSE) = NDFN
   DO 470 IDFN=1, NDFN
   ARM(JMNI, OFN, LUSE) = ARMUMNI(I, OFN, IM)
490 CONTINUE
   NDFN = LUSE
   IF( NDFN ) 500, 500, 1000
500 IEREGNV = 4
C
900 CONTINUE

ERROR RETURN AT THIS POINT
C
1000 RETJRN
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 \( \eta \) 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 $n \geq .2$, and by subroutine APROX2 (sec. 3.3.2) for $n < .2$. 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 $n > 0$, the second and third starting values are the first $-.1$ and $+.1$; for $n = 0$, 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.
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**

```fortran
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 EQUATION) is less than $10^{-10}$. 

129
CALLING LIST
ETA, HOFM, 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
SET THE ITERATES AND CALCULATE THE ZERO SC (IOFN)

CALL

APPROX2

CALL

APPROX1

APPROXIMATE ZERO

<

ETA : .2

APPROXIMATE ZERO

CALL

JARRATT

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

DO 600 M = 1, MMAX

DO 500 IOFN = 1, NMAX

LOOP ON m INDEX

LOOP ON n INDEX

B

131
SET THE ITERATES AND CALCULATE THE ZERO

CALL JARRATT

SC (IOFN)

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, NONM, ITACE, SC, ARMUYN)

PURPOSE

COMPUTE THE ZEROS OF THE EQUATION

\[ JP(M, X) \times YP(M, ETA \times X) - YP(M, X) \times JP(M, ETA \times X) \]

WHERE JP AND YP ARE THE DERIVATIVES OF THE BESSEL FUNCTIONS OF THE FIRST AND SECOND KINDS, RESPECTIVELY, OF ORDER M AND ARGUMENT X OR ETA*X.

METHOD

THE EQUATION IS TO BE SOLVED FOR THE FIRST M ZEROS FOR EACH ORDER, THE ORDER M = 0, 1, 2, ..., MMAX. FOR THIS PURPOSE THE ZEROS FOR M = 0 ARE FOUND BY FIRST APPLYING AN APPROXIMATION FORMULA THEN A REFINEMENT PROCEDURE USING BESSEL FUNCTION EVALUATORS. THE ZEROS FOR HIGHER ORDERS ARE FOUND BY STEPPING THROUGH ORDER USING A NONLINEAR EQUATION SOLVER AND BESSEL EVALUATORS (AS THE REFINEMENT) WITH STARTING VALUES BEING THE ZEROS FOR THE PREVIOUS ORDER.

PROGRAMS

JARRATT NONLINEAR EQUATION SOLVER
APRX1 APPROXIMATION TO EQUATION FOR ETA AT LEAST 0.2
APRX2 COMBINATION OF APPROXIMATION AND INTERPOLATION FOR ETA LESS THAN 0.2

EXTERNALS

EQUATION EVALUATES THE EQUATION

DIMENSION GUESS(3), ARMUXN(NDIM+1), SC(1), MUSE(1)
EXTERNAL EQUATION

THIS COMMON PASSES M AND ETA TO EQUATION EVALUATOR

COMMON/CEQUAT/M, ETA

EQUATION SOLVER TOLERANCES

ITLIM=30
EP1=0.
EP2=1.E-10

ETA = ETA
MMAX = MAX0(IABS(MUSE(1)), IABS(MUSE(NOFM)))
MM = MM + 1

SOLVE THE EQUATION FOR ORDER ZERO

M=0

133
GUESS(1) = SC(IOFN)
GUESS(2) = GUESS(1) - 1
IF(ETA.EQ.0.) GUESS(2) = GUESS(1) + 2
GUESS(3) = GUESS(1) + 1
CALL JARRAY(GUESS, ITLIM, EP1, EP2, EQUATION, ZERO, FT, IERJAR)
IF(ITRACE .EQ. 2) WRITE(6,20) ZERO, FT, GUESS(1), IERJAR
SC(IOFN) = ZERO
500 CONTINUE
DO 540 IOFM = 1, NDFM
 M SAVE = IABS(MUSE(IOFM))
 IF(M - M SAVE) 540, 510, 540
 10 DO 530 IOFN = 1, NMAX
 530 ARMJMN(IOFN, IOFM) = SC(IOFN)
 540 CONTINUE
500 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 \( \eta \).

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: \[ 3X + M + 12 \leq 1000 \text{ (see BSSL)} \] \[ M \geq 0 \] \[ 0 \leq \eta < 1 \]

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 \textsc{BFUF} (ref. 41) and \textsc{BSSL} (sec. 3.3.6).

Accuracy: The accuracy is of the algorithmic type and, in particular, is dominated by subroutines \textsc{BFUF} and \textsc{BSSL}. 
CALLING LIST

ENTER

EVALUATE $J_m(x)$

CALL BSSLS

FJP

ETA:0

CALL BSSLS

CALL BF4F

COMPUTE $Y_m(x)$, $J_m'(x)$, $Y_m'(x)$

FNP, FJFP, FNFP

EVALUATE THE CROSS PRODUCT

EQUATION

RETURN
FUNCTION EQUATION(X)

PURPOSE
EVALUATE THE FOLLOWING EQUATION USING BESSEL FUNCTION EVALUATORS

\[ \text{JP}(M,X) \times \text{YP}(M, \text{ETA} \times X) - \text{YP}(M,X) \times \text{JP}(M, \text{ETA} \times X) \]

WHERE JP AND YP ARE THE DERIVATIVES OF THE BESSEL FUNCTIONS OF FIRST AND SECOND KIND, RESPECTIVELY, FOR INTEGER ORDER M AND REAL ARGUMENT X AND ETA*X, ETA A GIVEN PARAMETER.

INPUT
VARIABLE DEFINITION
X REAL ARGUMENT X
M INTEGER ORDER M, OBTAINED FROM COMMON
ETA PARAMETER ETA, OBTAINED FROM COMMON

NOTATION
VARIABLE DEFINITION
FNFP YP(M, ETA*X)
FJFP JP(M, ETA*X)
FJP JP(M, X)
FNP YP(M, X)

LIBRARY ROUTINES RRC LIBRARY ROUTINES BSSL AND BF4F THAT EVALUATE BESSEL FUNCTIONS OF FIRST AND SECOND KIND, RESPECTIVELY.

RESTRICTIONS 3*X + M + 12 CAN BE AT MOST 1000

COMMON/CEQUAT/M, ETA
COMMON/SCRATCH/BES(1000)
DATA ISIGN/-1/

MP1=M+1
MP2=M+2

CALL BSSL(X, BES, MP1, IERR)
A1=BES(MP1)
A2=BES(MP2)
FJP=-(A2+(M/X)*A1)

IF(ETA.NE.0.) GO TO 10
EQUATION = FJP
GO TO 30

10 CALL BF4F(X, BES, MP1, IERR, ISIGN)
A3=BES(MP1)
A4=BES(MP2)
FNP=-(A4+(M/X)*A3)

Y = X*ETA
CALL BSSL(Y, BES, MP1, IERR)
TRACE FORMATS

IF (TRACE.EQ.2) WRITE (6,5) ETA
5 FORMAT (1X, 'TRACE THE CALCULATION FOR THE ZEROS OF THE ANNULAR ' , ', THE RATIO OF THE INNER TO OUTER RADIUS IS ', 2X, F10.5)
IF (TRACE.EQ.2) WRITE (5,10) M
IF (TRACE.EQ.2) WRITE (6,15)
15 FORMAT (1X, 'O. ', 6X, 'SET')

FIND THE N-TH ZERO BY APPLYING THE APPROXIMATION FORMULA THEN THE REFINEMENT PROCEDURE

DO 50 IOFN=2,NMAX
NTHZERO = IOFN - 1
IF (ETA .GE. .2) CALL APROX1 (RM, NTHZERO, ETA, ZERJ)
IF (ETA .LT. .2) CALL APROX2 (RM, NTHZERO, ETA, ZERJ)
GUESS(1) = ZERO
GUESS(2) = ZERO + .1
GUESS(3) = ZERO - .1
CALL JARRAY (GUESS, ITLIM, EP1, EP2, EQUATION, ZERO, FT, IERJAR)
IF (TRACE.EQ.2) WRITE (6,20) ZERO, FT, GUESS(1), IERJAR
SC(IOFN) = ZERO
20 FORMAT (1X, '3E22.14', 1X)
50 CONTINUE

DO 24C IOFM=1,NOFM
MSAVE = IABS( MUSE(IOFM) )
IF (M-MSAVE) 240,210,240
210 DO 23C IOFN=1,NMAX
230 ARMJMA(IOFN, IOFM) = SC(IOFN)
240 CONTINUE
SC(1) = 1.

IF (MMAX.EQ.0) GO TO 700

NOW STEP THROUGH THE ORDERS

DO 50C M=1,MMAX
RM=M
IF (TRACE.EQ.2) WRITE (6,10) M
MP1=M+1

FIND THE N-TH ZERO BY ITERATION

DO 50G 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

EQUATION = FJFP*FNP - FJP*FNFP

DO 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, $n$, when the argument is $\mu_{mn}\rho$, where $\mu_{mn}$ is the $n^{th}$ hardwall eigenvalue of an $m^{th}$ order duct mode and $\rho$ is the polar radial coordinate nondimensionalized on the duct outer radius:

$$R_m(\mu_{mn}\rho) = J_m(\mu_{mn}\rho) - \frac{J'_m(\mu_{mn})}{Y'_m(\mu_{mn})} Y_m(\mu_{mn}\rho),$$

where $J_m$ and $Y_m$ are the Bessel and Neumann functions, respectively; the primes denote differentiation with respect to the argument; and $n$ 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 $\rho$).

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

4) Evaluate $|J_m(\mu_{mn})|, |Y_m(\mu_{mn})|$, and $|Y_m(\mu_{mn})|$ using the recurrence relations (formula [9.1.27] of ref. 30).

5) Evaluate $R_m(\mu_{mn}\rho)$.

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

141
Usage: CALLING SEQUENCE

COMMON/SCRATCH/BES(1000)
.
.
.
CAPRMN=UNEGNFN(M,R,MUN,ETA,S)

Restrictions: \(3 \frac{\mu}{\mu_{\min}} + 12 + |m| \leq 1000\); see subroutines BSSLS (sec. 3.3.6) and BF4F (ref. 41)

\(S \geq 0\)

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.
FUNCTION UNEGNFn(M, RMUNN, ETA, S)

PURPOSE
COMPUTE THE UNNORMALIZED ANNULAR EIGEN FUNCTION

\[ \text{RMN}(\text{RMUNN}*S) = \text{JM}(\text{RMUNN}*S) - \]
\[ \text{JMP}(\text{ETA} \cdot \text{RMUNN}) \cdot \text{YM}(\text{RMUNN}*S) / \text{YMP}(\text{ETA} \cdot \text{RMUNN}) \]

WHERE S IS BETWEEN 0 AND 1
RMUNN IS THE N-TH EIGENVALUE OF THE ANNULAR EIGENVALUE EQUATION OBTAINED BY DIFFERENTIATING THE ABOVE EQUATION
ETA IS THE RATIO OF THE INNER TO OUTER RADIUS OF THE ANNULUS
JM AND YM ARE THE BESSEL FUNCTIONS OF THE FIRST AND SECOND KINDS OF INTEGER ORDER M
JMP AND YMP ARE THE DERIVATIVES OF JM AND YM

INPUT VARIABLE DEFINITION
M INTEGER ORDER OF BESSEL FUNCTION
RMUNN N-TH EIGENVALUE OF ANNULAR EIGENVALUE EQUATION
JM AND YM ARE THE BESSEL FUNCTIONS OF THE FIRST AND SECOND KINDS OF INTEGER ORDER M
JMP AND YMP ARE THE DERIVATIVES OF JM AND YM

OUTPUT VARIABLE DEFINITION
UNEGNFn THE VALUE OF RMN(RMUNN*S)

SUBPROGRAMS
BSSLs EVALUATES JM FROM LRC LIBRARY
BF4F EVALUATES YM FROM LRC LIBRARY

RESTRICTION
3*RMUNN + M + 12 CAN BE AT MOST 1000 (SEE ARRAY BJ)
ETA BETWEEN 0 AND 1, GENERALLY AT LEAST ETA
S BETWEEN 0 AND 1, IN GENERAL RMN IS ONLY MEANINGFUL FOR S GREATER THAN OR EQUAL TO ETA

COMMON/SCRATCH/BES(1000))
DATA ISIGN/-1/

COMPUTE BESSEL RELATED FUNCTIONS WITH POSITIVE ORDER AND SWITCH SIGN FOR ODD NEGATIVE ORDER
IA3SM = IABS(M)
IF(RMUNN) 100*10,20

USE LIMITING VALUE FOR RMUNN=0 WHERE M=0
10 IF(IABS+,E,0 ) UNEGNFn = 1.
GO TO 100
20 CONTINUE
MPI = IABS(M) + 1

144
MP2 = MP1+1
ARGETA = ETA*RUMN
ARGS = S*RUMN

COMPUTE JM(MUNM*S)
CALL BSSLS(ARGS,BES,IABSM,IERR)
T2 = BES(MP1)
UNEGNFN = T2
IF(ETA.EQ.0.) GO TO 90

COMPUTE YMP(ETA*MUNM)
CALL BF4F(ARGETA,BES,MP1,IERR,ISIGN)
T1 = -BES(MP2) + (IABSM/ARGETA)*BES(MP1)

COMPUTE JMP(ETA*MUNM)
CALL BSSLS(ARGETA,BES,MP1,IERR)
T3 = -BES(MP2) + (IABSM/ARGETA)*BES(MP1)

COMPUTE YM(MUNM*S)
CALL BF4F(ARGAS,BES,IABSM,IERR,ISIGN)
T4 = BES(MP1)

UNEGNFN = UNEGNF - T3*T4/T1
90 CONTINUE
IF(M.LT.0 .AND. MOD(M,2).NE.0) UNEGNF = -UNEGNF

100 RETURN
END
3.2.5 Function EGNNORM

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

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

for \( m \neq 0 \) and \( m = 0, n \neq 0 \) and

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

Method:
The procedure is as follows:

1) Set \( N_{oo} \) when \( u_{mn} = 0 \).

2) Compute the ratio \( (m/\mu_{mn})^2 \).

3) Evaluate \( R_m(u_{mn}) \) and \( R_m(u_{mn}) \) where \( R_m(u_{mn}) \) is set to
   zero when \( \eta = 0 \).

4) Evaluate \( N_{mn} \).

Usage:

**CALLING SEQUENCE**

\[
\text{CAPNMN}=\text{EGNNORM}(M,RMUMN,ETA)
\]

Timing:
The timing is dominated by function UNEGPN (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 UNEGPN.
FUNCTION EGNNORM(M,RMUMN,ETA)

PURPOSE
COMPUTE THE NORMALIZATION FACTOR TO THE ANNULAR EIGENFUNCTION

NMN**2 = 0.5*( (1.-M**2/RMUMN**2)*RMN(RMUMN)**2 - (ETA**2-M**2/RMUMN**2)*RMN(ETA*RMUMN)**2 )

WHERE RN IS THE UNNORMALIZED ANNULAR EIGENFUNCTION
M IS BESSEL FUNCTION ORDER FOR RN
Mumn IS ANNULAR EIGENVALUE
ETA IS RATIO INNER TO OUTER ANNULAR DUCT RADII

INPUT VARIABLE DEFINITION
M BESSEL FUNCTION ORDER
Mumn ANNULAR EIGENVALUE
ETA RATIO INNER TO OUTER ANNULAR DUCT RADII

OUTPUT EGNNORM VALUE OF NMN

SUBROGRAMS UNEGFN EVALUATES UNNORMALIZED ANNULAR EIGENFUNCTION

RESTRICTIONS SEE FUNCTION UNEGFN

IF(RMUMN) 100,10,20
USE LIMITING VALUE FOR RMUMN=0 WHERE M=0

10 IF(M.EQ.0) EGNNORM = SQRT(0.5*(1.-ETA**2))
GO TO 100
20 CONTINUE
RATIO = (M/RMUMN)**2
CAPRONE = UNEGFN(M,RMUMN,ETA,1.)
CAPRETA = 0.
IF(ETA.NE.0.)CAPRETA = UNEGFN(M,RMUMN,ETA,ETA)

EGNNORM = 0.5*( (1.-RATIO)*CAPRONE**2 - (ETA**2-RATIO)*CAPRETA**2 )
EGNNORM = SQRT(EGNNORM)

130 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 AAAAAA:

\[
\exp\left(-i\pi N IS R_0 \ell_m \left(\mu_{mn}\right)\right) \exp\left(i\pi R IS \frac{d \sin \psi}{\rho \cos \psi}\right)
\]

for the inlet stator-rotor, and

\[
\exp\left(i\pi N R OS \ell_m \left(\mu_{mn}\right)\right) \exp\left(i\pi R \frac{d \sin \psi}{\rho \cos \psi}\right)
\]

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 \( M_{le} \) from input or spanwise interpolation and the flow angle.

4) Compute the second exponential factor.

5) Evaluate the oscillatory factor.
**Usage:**

**CALLING SEQUENCE**

\[
\begin{align*}
\text{COMMON/CFACT/M,N,RMUMN,CAPMN,ETA,SIGN,L,CAPKNM} \\
\text{COMMON/CFACTIR/NSBIR,SIGOL,PHISBIR} \\
. \\
. \\
. \\
\text{FACTRPD = FACTINT (RHO)}
\end{align*}
\]

**Restrictions:** \( \eta \leq \rho \leq 1 \)

**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.
CALL COMPUTE MACH NUMBER BY SPANWISE INTERPOLATION

ENTER

COMPUTE \( R_m (\mu_{mn}) \)

CALL UNEGFIN

COMPUTE THE FIRST EXPONENTIAL TERM

EXPFACT

CALL MTLUP

Computes Mach numbers \( M_z (J = 9) \) and \( M_{1E} (J = 8) \)

AVERAGE VALUE IS INPUT FMZ AR (2,J,K)

COMPUTE FLOW ANGLES

COSPSI, SINPS1

Compute the second exponential term

CTEMP2

Evaluate oscillating term

FACTINT = SCPTRMN * EXPFACT * CTEMP2

RETURN
COMPLEX FUNCTION FACTINT(RHO, ARMISC, MAXDIM, MAXJ, AR)

PURPOSE EVALUATE THE INTEGRAND FACTINT TO BE CALLED BY THE INTEGRATOR GAUSS2 IN PRIMARY SUBROUTINE AAAA.

COMMON CFACT, BLOC
SUBPROGAMS UNEGNFN
CALLED

DIMENSION AR(MAXDIM, MAXJ, 3), ARMISC(1)
COMPLEX EXPFACT, CTEMP2
COMMON/CFACT/ M, N**MU, ETA, CAPNMN, ETA, SIGN, L*CAPKM
COMMON/CFACT/ NSBIR, SIGOL, PHISBR

EVALUATE NORMALIZED EIGENFUNCTION
SCPTRMN = UNEGNFN(M, RMAWN, ETA, RHO)/CAPNMN

EVALUATE FIRST EXPONENTIAL TRIGONOMETRIC FACTOR
ARGEXP2 = SIGN*SIGOL*NSBIR*PHISBR*(RHO-ETA)/(1.-ETA)
EXPFACT = CMPLX( COS(ARGEXP2), SIN(ARGEXP2) )

EVALUATE NEXT EXPONENTIAL TRIGONOMETRIC FACTOR

COMPUTE F'MZ AND F'M1E

ISRORS = ARMISC(5)
J = 9
K = ISRORS+1
NSPN = AR(1, J, K)
IF(NSPN) 40, 30, 40
30 F'MZ = AR(1, J, K)
GO TO 50
40 IPA = -1
CALL MTLUP(RH0, F'MZ, 1, NSPN, NSPN, 1, IPA, AR(3, 1, K), AR(3, J, K) )
50 J = 8
K = ISRORS
NSPN = AR(1, J, K)
IF(NSPN) 70, 60, 70
60 F'M1E = AR(2, J, K)
GO TO 80
70 IPA = -1
CALL MTLUP(RH0, F'M1E, 1, NSPN, NSPN, 1, IPA, AR(3, 1, K), AR(3, J, K) )
80 COSPSSI = F'MZ/F'M1E
SINPSI = SORT(1.-COSPSSI**2)
DSPAC = Armsci(ISORS)
TEMP1 = SIGD*NSBIR*DSPAC*SINPSI/(RHO*COSPSSI)
CTEMP2 = CMPLX( COS(TEMP1), SIN(TEMP1) )

COMBINE TO FORM FACTINT

FACTINT = EXPFACT*SCPTRMN*CTEMP2

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:

\[ M_{M,K_1}(\rho) \Gamma_{K_2}(\rho) a_{K_1}(\rho) H_{K_1}(\rho) e^{-id_{K_1}(\rho)} K_{K_1}(\rho) \]

\[ \times \left( \frac{m e^\phi}{\rho} + K_{mn} e^z \right) \left( \frac{dC_L}{d\alpha}(\rho) / 2\pi \right) \theta_m \left( \mu_{mn} \rho \right) \]

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

Method: The procedure is as follows:

1) Initialize the component index to \( K_1 \).

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 \( M_{M,K_1}(\rho) \) and \( \theta_{K_1}(\rho) \).

5) Repeat steps 2, 3, and 4 for component index \( K_2 \).

6) Compute the Glauert coefficients for component \( K_2 \) using the procedure described in step 3.

7) Compute \( r^{\circ}_{K_2}(\rho) \) and \( a_{K_1}(\rho) \).
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 $H_{K2}(\rho)$ by summation.

12) Compute $d_{K2}(\rho)$ and $e_{K1}(\rho)$.

13) Compute $\gamma$ and $\lambda$ and the corresponding value of the lift function.

14) Compute the factor $\left(\frac{me^\phi}{\rho} + k_{mn}^G e^z\right)\left(\frac{dC_L}{d\alpha}(\rho) / 2\pi\right)$.

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: \[ \eta \leq \rho \leq 1 \]

\[ 3 \leq \text{ARMISC}(18 + k_2) \leq 15 \]

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

INITIALIZE TABLE LOOK UP INDEX
IP = -1

MTLUP

AVERAGE VALUE OF INLET MACH NO. IS INPUT
MSBI (KI) = AR (2, 7, KI)

AVERAGE VALUE OF EXIT MACH NO. IS INPUT
MSBE (KI) = AR (2, 8, KI)

AVERAGE VALUE OF AXIAL MACH NO. IS INPUT
MSBZ (KI) = AR (2, 9, KI)

AVERAGE VALUE OF CHORD IS INPUT
CHORD (KI) = AR (2, 2, KI)

A

COMPUTE INLET MACH NO. BY INTERPOLATION ALONG THE SPAN
AR (1, 7, KI) =

COMPUTE EXIT MACH NO. BY INTERPOLATION ALONG THE SPAN
AR (1, 8, KI) =

COMPUTE AXIAL MACH NO. BY INTERPOLATION ALONG THE SPAN
AR (1, 9, KI) =

COMPUTE CHORD BY INTERPOLATION ALONG THE SPAN
AR (1, 2, KI) =

A
LOOP ON GLAUERT COEFFICIENT INDEX
DO 170 IND=1, MAXIND

AVERAGE VALUE OF GLAUERT COEFF. IS INPUT
CAPA (IND) = AR (2,IND+9,K2)

COMPUTE GLAUERT COEFF. BY INTERPOLATION ALONG THE SPAN

170 CONTINUE

COMPUTE $\Gamma_{K2}$ AND $\alpha_{X,K1}$
GAMMA, ARHO

COMPUTE $\delta_{K2}$
HRHO

B
AVERAGE VALUE OF SLOPE OF STEADY STATE LIFT COEFF. IS INPUT
DCSBL(K1) = AR(2,6,K1)

\[ \frac{dC_L}{d\alpha} \left( \frac{m \rho \Phi + K_{mn} \varepsilon z}{\rho_0} \right) \]

\[ = \text{AR(1,6,K1)} : 0 \]

COMPUTE SLOPE OF STEADY STATE LIFT COEFF. BY INTERPOLATION ALONG THE SPAN


1. \( \epsilon \)
2. \( \text{COMPUTE } R_m(\mu_{\text{min}}) \)
3. \( \text{SCPTRMN} \)
4. \( \text{UNEGFN} \)
5. \( \text{COMPUTE OSCILLATORY FACTOR} \)
6. \( \text{FACTIN2} \)
7. \( \text{RETURN} \)
COMPLEX FUNCTION FACTINZ(RHO, ARMISC, MAXDIM, MAXJ, AR)
REAL MSBI(3), MSBZ(3), MSBM(3), MSBZ(3)
COMPLEX CAPHRHO, CONLIFT, CTMP1, CTMP2, HRHO, LAMDA, LIFT, TERM(15)
COMPLEX CAPHRHO, EXPPHO
DIMENSION CHORD(3), ARMISC(1), BJHR(l250), BJHI(250), CAPA(15)
DIMENSION CHORD(3), ARMISC(1), BJHR(l250), BJHI(250), CAPA(15)
COMMON/SCRATCH/ES(i:CO)
COMMON/CFIT2/BAPXMN, CAPNMN, C, C6, C7, C8, C9, C10, C11, C12, C13, C14, K, K2, L, M, N, NK2, RMUN, SIGOM
EQUIVALENCE (BJhr(l250), BJHI(250), ES(1250))
K1 = K1
IP = -1
IF (AR(l, 7, KI) .GT. 0.) GO TO 20
MSBI(KI) = AR(l, 7, KI)
GO TO 40
NPTS. = AR(l, 7, KI)
CALL MTLUPPHO, MSBI(KI), l, NPTS, NPTS, l, IP, AR(l, 3, KI), AR(3, 7, KI))
IF (AR(l, 3, KI) .GT. 0.) GO TO 50
MSBE(KI) = AR(l, 3, KI)
GO TO 70
NPTS. = AR(1, 8, KI)
CALL MTLUPPHO, MSBE(KI), l, NPTS, NPTS, l, IP, AR(3, l, KI), AR(3, 8, KI))
IF (AR(l, 9, KI) .GT. 0.) GO TO 80
MSBI(KI) = AR(l, 9, KI)
GO TO 100
NPTS. = AR(l, 9, KI)
CALL MTLUPPHO, MSBI(KI), l, NPTS, NPTS, l, IP, AR(l, 3, KI), AR(3, 9, KI))
IF (AR(l, 2, KI) .GT. 0.) GO TO 110
CHORD(KI) = AR(l, 2, KI)
GO TO 130
NPTS. = AR(l, 2, KI)
CALL MTLUPPHO, CHORD(KI), l, NPTS, NPTS, l, IP, AR(3, l, KI), AR(3, 2, KI))
IF (MSBI(KI) .LT. MSZ2(KI)) TEMP1 = 0.
IF (MSBI(KI) .GE. MSZ2(KI)) TEMP1 = SQRT(MSBI(KI)**2 - MSZ2(KI)**2)
1
IF (MSBE(KI) .LT. MSZ2(KI)) TEMP2 = 0.
IF (MSBE(KI) .GE. MSZ2(KI)) TEMP2 = SQRT(MSBE(KI)**2 - MSZ2(KI)**2)
1
TEMP3 = TEMPI + TEMP2
TEMP4 = (TEMP3**2)/4. + MSZ2(KI)**2
MSBI(KI) = SQRT(TEMP4)
TEMP1 = ACOS(MSBI(KI)/MSBM(KI))
THETA(KI) = ABS(TEMP1)
IF (KI .EQ. K2) GO TO 140
K1 = K2
GO TO 10
MAXIND = ARMISC(K2+1B)
GO TO 170
140 IF (K1 .LT. MAXIND) GO TO 150
CAPA(K1) = AR(l, 10, 1, K1, K2)
GO TO 170
150 NPTS = AR(l, 10, 3, K2)
NPTS = AR(1,6,K1)
IP = -1
CALL PTLUP(RHO, DCSBL(K1), 1, NPTS, NPTS, 1, IP, AR(3, 1, K1), AR(3, 6, K1))

225 TEMP4 = (DCSBL(K1)/6.2831853071796) * TEMP4

SCPTRMN = UNEGFN(M, RMJMN, ARMISC(3), RHO) / CAP4YN
FACTIN2 = MSBC(K1) * GAMMA * RHO * CAPRHO * EXPDROH * CAPKRHO * TEMP4 * SCPTRMN

IF (ARVISC(6) .EQ. 3.3) THEN
  RHO = MSB(K1) * GAMMA * RHO
  CAPRHO * EXPDROH * CAPKRHO * TEMP4 * SCPTRMN * FACTIN2
END IF

230 FORMAT(1H,F7.4,F5.3,2(1X,E9.2),3(1X,E9.2),2(1X,E9.2),1X,2E10.3)

RETJRN
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 $\ell$ is zero, proceed to step 16.

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

5) Calculate $D = (1 - \frac{V_A}{V_1})/(A^2 - 1), \delta_{\ell,0}$ 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 $a_\ell$ is input, use it and proceed to step 11.

10) Compute an average value of $a_\ell.$

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 $|l|$ 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 $l < 0$, conjugate the coefficient.

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

**Usage:**

**CALLING SEQUENCE**

```plaintext
COMPLEX PACTIN3, VFACTIN
COMMON/CFACT/M,N,RMUMN,CAPNMIN,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 UNEGIGN plus, for the cone model, $2 \times |l|$ 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 UNEGIGN and, for the cone model, GAUSS, and for the power model or direct Fourier coefficient input, MTLUP.
CALLING LIST
RHO, AR, MISC, MAXDIM, MAXJ, AR

START

EVALUATE THE EIGEN FUNCTION
SCPTRM

INITIALIZE THE DISTORTION COEFFICIENT TO ZERO
WSBLRHO=0.0.

L = 0

MISC22 = AR MISC (22)

MISC 22:1

= 0

= 20

compute D, d_j, o, d_j, 1 + s_j, -1
DVALUF, DELTALO, DELTALI

CALCULATE NUMBER OF SUBINTERVALS AND SUB INTERVAL WIDTH
NOSCE, WIDTHI

A
DO 10 ITERVL=1, NOSCE

COMPUTE Subinterval
Upper and lower bounds
RUP, RLOW

COMPUTE the integral
On the subinterval
VALUINT

Accumulate integral
10 WSBLRHO=WSBLRHO+VALUINT

WSBLRHO = -WSBLRHO

A

GAUSS

B

20

MISC 22.2
/=>

Q = ARMISC (23)

Compute the J index of A1
Used in power model
J

100

60

167
COMPUTE THE VALUE OF $a_i$ BY INTERPOLATION ALONG SPANWISE INPUT DATA

COMPUTE $w(x)$ IN POWER MODEL

USE THE AVERAGE VALUE INPUT

COMPUTE THE FOURIER DISTORTION COEFFICIENT FOR $|x|$;

COMPUTE THE FOURIER INDEX CORRESPONDING TO $|x|$ IN THE INPUT, IF ANY

DO 50 $i = 1, 2$

Compute the J location of the input data corresponding to ICOEF for the distortion cosine and sine series \(i=1,2\) respectively.

Compute the average value by spanwise interpolation

\[ ABN (i) = \text{average input value} \]

Use the average input value

\[ ABN (i) = AR (2, J, 2) \]

50 CONTINUE

Compute the complex distortion coefficient for \(J\)

\[ \text{WSBLRHO} = \text{CMPLX}(\text{ABN}(1), \text{ABN}(2)) \]

Compute the integrand of oscillatory factor

\[ \text{FACT} \text{IN}3 = \text{SCPTRMN} \times \text{WSBLRHO} \]

RETURN
COMPLEX FUNCTION FACTIN3(RHO, ARMISC, MAXDIM, MAXJ, AR)

PURPOSE
EVALUATE THE INTEGRAND FACTIN3 TO BE CALLED BY THE INTEGRATOR GAUSS2

DIMENSION AR(MAXDIM, MAXJ, 3), ARMISC(1), ABN(2)
COMMON/FACT/ M, N, RMUNN, CAPMN, ETA, SIGN, L, CAPMN
COMPLEX WSBLRHO, VSBLRHO3, DISINT, VALUINT
COMMON/COINT/ CAPADIS, RHOINC
EXTERNAL DISINT
DATA PI/3.14159265358979/

EVALUATE NORMALIZED EIGENFUNCTION
SCPTRHN = UNEGNFNM(RMJN, ETA, RHO)/CAPMN

COMPUTE THE DISTORTION FACTOR
WS6LRHO = (.0, 0.)
IF(L.EQ.0) GO TO 100
IF( ARMISC(22) .EQ. 0. ) GO TO 100

COMPUTE THE FOURIER COEFFICIENTS FROM CONE MODE
IF( ARMISC(22) .NE. 1. ) GO TO 20
RHOINC = RHO
VA0 = ARMISC(23)
CAPADIS = ARMISC(24)
OVALUE = (1. - VA0)/(CAPADIS**2 - 1.)
DELTAL0 = 0.
IF(L.EQ.0)DELTAL0 = 1.
DELTAL1 = 0.
IABSL = IABSL(L)
IF(IABSL.EQ.1)DELTAL1 = L.
IORDGS = 2
NOSCE = MAXO(2, IABSL)
WIDTHI = 2.*PI/NOSCE
DO 10 ITRL = 1, NOSCE
   RLOW = (ITRL - 1)*WIDTHI
   RUP = RLOW + WIDTHI
   CALL GAUSSJRLOW, RUP, VALUINT, DISINT, IORDGS
       10 WSBLRHO = WSBLRHO + VALUINT
WSBLRHO = (1.+OVALUE)*DELTAL0 - .5*CAPADIS*RHO*DELTAL1*OVALUE
         - .5*OVALUE**2*SLRHO/Pi
WSBLRHO = -2.*PI*WSBLRHO
GO TO 100
COMPUTE THE FOURIER COEFFICIENT FROM POWER MODEL

IF (ARMISC(22) .NE. 2) GO TO 50
Q = ARMISC(23)
IA3SL = IABS(L)
J = 9 + ARMISC(20) * 2 + 1
NSPN = AR(1,J,2)
IF (NSPN • 40,30,40
30 COEFAJ = AR(2,J,2)
GO TO 50
40 IPA = -1
CALL MTLUP(RH0,COEFAJ,1,NSPN,NSPN,1,IPA,AR(3,1,2),AR(3,J,2))
50 WSLRHO = .5*COEFAJ/FLOAT(IA3SL)**Q
GO TO 100

COMPUTE THE FOURIER COEFFICIENT FROM INPUT VALUES

IF (ARMISC(22) .NE. 3) GO TO 100
IA3SL = IABS(L)
MAXCOEF = ARMISC(23)/2
MULTFCT = ARMISC(24)
GO 65 NCOEF = MAXCOEF
IF (IA3SL .NE. NCOEF*MULTFCT) GO TO 65
IC0EF = NCOEF
GO TO 67
65 CONTINUE
GO TO 100
67 DO 90 IAB = 1,2
J = 9 + ARMISC(20) + 2 + 2*(IC0EF-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(RH0,ABN(IAB),1,NSPN,NSPN,1,IPA,AR(3,1,2),AR(3,J,2))
90 CONTINUE
WSBLRHO = .5*COMPLEX(ABN(1),-ABN(2))
IF (L.LT.0) WSLRHO = CONJG(WSBLRHO)
COMBINE TO FORM FACTIN3
100 FACTIN3 = SCPTRMN*WSBLRHO
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):

\[ \text{FACTIN4} = R_m \left( u_{mn} \right) \left( g_1(\rho) F_1(\rho) - g_2(\rho) F_2(\rho) \right) \]

where:

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

\[ P_j = \begin{cases} U_j & \text{if } BT_j < 1 \\ E_j & \text{if } 1 \leq BT_j \leq 10 \\ -\frac{1}{2} \left( \frac{T}{T_j} \right)^2 e & \text{if } 10 < BT_j \end{cases} \]

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

If ARMISC(38) = 0, then:

\[ F_1(\rho) = M_{M,2}(\rho) \sin(\theta_2(\rho)) S(\varphi(\rho)) - \alpha(\rho) M_{Z,2}(\rho) F(\varphi(\rho)) - f(\rho) M_{Z,2}(\rho) F(\varphi(\rho)) \]

\[ F_2(\rho) = M_{Z,2}(\rho) S(\varphi(\rho)) + \alpha(\rho) M_{M,2}(\rho) \sin(\theta_2(\rho)) F(\varphi(\rho)) + f(\rho) M_{M,2}(\rho) \sin(\theta_2(\rho)) F(\varphi(\rho)) \]
If ARMISC(38) ≠ 0, then:

\[ F_1(\rho) = M_{M,2}(\rho) \sin(\Theta_2(\rho)) S\left(\nu_\ell(\rho)\right) J\left(\kappa_{mn\sigma}^{\pm}\right) - \alpha(\rho) M_{Z,2}(\rho) J\left(\nu_\ell + \kappa_{mn\sigma}^{\pm}\right) \]

\[ - f(\rho) M_{Z,2}(\rho) \left\{ J_1\left(\kappa_{mn\sigma}^{\pm}\right) F\left(\nu_\ell\right) + \frac{2J_1\left(\nu_\ell + \kappa_{mn\sigma}^{\pm}\right)}{\nu_\ell + \kappa_{mn\sigma}^{\pm}} \right\} \]

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

\[ F_2(\rho) = M_{Z,2}(\rho) S\left(\nu_\ell(\rho)\right) + \alpha(\rho) M_{M,2}(\rho) \sin(\Theta_2(\rho)) J\left(\nu_\ell + \kappa_{mn\sigma}^{\pm}\right) \]

\[ + f(\rho) M_{M,2}(\rho) \sin(\Theta_2(\rho)) \left\{ J_1\left(\kappa_{mn\sigma}^{\pm}\right) F\left(\nu_\ell\right) + \frac{2J_1\left(\nu_\ell + \kappa_{mn\sigma}^{\pm}\right)}{\nu_\ell + \kappa_{mn\sigma}^{\pm}} \right\} \]

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

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

\[ \sin(\Theta_2(\rho)) = \frac{\sqrt{M_{I,2}^2(\rho) - M_{Z,2}^2(\rho)} + \sqrt{M_{E,2}^2(\rho) - M_{Z,2}^2(\rho)}}{2 \cdot M_{M,2}(\rho)} \]

\[ \nu_\ell = \frac{C_2(\rho)}{2 \rho} \ell \cdot \sin(\Theta_2(\rho)) \]
If \text{ARMISC(25)} = 4 \text{ (i.e., LIFTFN4 is specified)}:

\[
\text{FACTORN4} = 2\sqrt{2\pi} \frac{\text{m}}{\alpha} \left(\frac{\mu}{\text{mm}}\right) e^{-l\phi}
\]

\[
\left\{ M_{M,2}(\rho) \sin(2(\rho)) P_1 I_\ell \left(\frac{\rho R}{a_1^2}\right) e^{-\rho R/a_1^2} I_1 
- M_{2,2}(\rho) P_2 I_\ell \left(\frac{\rho R}{a_2^2}\right) e^{-\rho R/a_2^2} I_2 \right\}
\]

where:

\[
I_j = a_j \Re \left\{ \int e^{-l(p-R)K} T^* \left( h(\rho, K), \psi(\rho, K) \right) e^{-K^2a_j^2/2} dK \right\}
\]

LIFTFN4 calculates \( T(h, \psi) \)

\[
h(\rho, K) = \frac{C_2(\rho)}{2} \sqrt{K^2 + \left[ \frac{\xi}{\rho} \sin \vartheta_2(\rho) \right]^2}
\]

\[
\psi(\rho, K) = \cos^{-1} \left( \frac{C_2(\rho)K}{2h(\rho, K)} \right) = \cos^{-1} \left( \frac{1}{\sqrt{1 + \frac{\xi^2}{(K\rho \sin \vartheta_2(\rho))^2}}} \right)
\]

\[
= \tan^{-1} \left( \frac{\xi}{K\rho \sin \vartheta_2(\rho)} \right)
\]
Method:

The procedure is as follows:

1) Evaluate normalized eigenfunction, \( R_m(u_{mn}) \).

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 \( [2\pi B/\pi] + 1 \) 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 \( I_j (pR/a_j^2)^{-pR/a_j^2} \).

7) Calculate \( e^{-i\phi} \).

8) Obtain \( M_{\alpha,2}(p), M_{\beta,2}(p), M_{\gamma,2}(p), C_{\alpha}(p) \) from array AR using linear interpolation, if necessary.

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

10) Calculate \( M_{\alpha,2}(p) \) and \( \sin \theta_{\alpha}(p) \).

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:
\[ I_j = a_j \Re \left\{ \int_0^\infty e^{-i(\rho-R)K} T^*(h,\psi) e^{-K^2a_j^2/2} \, dk \right\} \]

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

(Let \( x = K\alpha \))

\[ = \Re \left\{ \int_0^\infty e^{-i(\rho-R)x/a_j} T^*(h_{\rho,x/a_j},\psi_{\rho,x/a_j}) \, e^{-x^2/2} \, dx \right\} \]

\[ \approx \sum_{K=0}^{K_{\text{MAX}}} \Re \left\{ e^{-i(\rho-R)(K\Delta/a_j)} e^{(K\Delta)^2/2} T^*(h_{\rho, K\Delta/a_j}, \psi_{\rho, K\Delta/a_j}) \right\} \]

where:

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

calculate \textsc{factin} and return.

\textbf{Usage:}

\textsc{calling sequence}

\begin{verbatim}
COMPLEX \textsc{factin} \[4, Z \]
DIMENSION \textsc{armisc}(40), \textsc{ar}(\textsc{maxdim}, \textsc{maxj}, 3)
COMMON/\textsc{cfact}/M,N,RMUMN,CAPNMN,ETA,SIGN,L,CAPKMN
COMMON/\textsc{scratch}/BES(1000)
\end{verbatim}

\begin{verbatim}
\textsc{z} = \textsc{factin4}(\textsc{rho}, \textsc{armisc}, \textsc{maxdim}, \textsc{maxj}, \textsc{ar})
\end{verbatim}
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 $I_j$. 
NPTS = AR (1,8,2)

YES

AR (1,8,2) > 0

NO

MSBE = AR (2,8,2)

MTLUP

CALCULATE MSBE

MSBZ = AR (2,9,2)

NO

AR (1,9,2) > 0

YES

NPTS = AR (1,9,2)

MTLUP

CALCULATE MSBZ

AR (1,2,2) > 0

YES

NPTS = AR (1,2,2)

MTLUP

CALCULATE MSBZ

NO

CHORD = AR (2,2,2)

MTLUP

CALCULATE CHORD

ARMISC (25) = 4 ?

NO

AR (1,10,2) > 0

YES

NPTS = AR (2,10,2)

FRHO = AR (2,10,2)

MTLUP

CALCULATE FRHO

NO

B

C
$\text{INTEG}(J) = \text{INTEG}(J) + \text{REAL} (\text{CTEMP2} \times \text{CLIFT4})$

$K = K + 1$

$K > K_{\text{MAX}}$?

$J = J + 1$

$J > 2$?

$\text{CALCULATE FACTIN4}$

$\text{RETURN}$
COMPLEX FUNCTION FACTIN4(RHO, ARMISC, MAXDIM, MAXJ, AR)

PURPOSE: EVALUATE THE INTEGRAND FACTIN4 TO BE CALLED BY THE
INTEGRATOR GAUSS2

REAL INTEGJ, MSBE, MSBI, MSBM, MSBT, MSB2
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/CFACT1, MN, RMUN, CAPNMN, ETA, SIGN, L, CKM1N
COMMON/CFUNIN4/CTJ, TAU
COMMON/SCRATCH/BES(100)
EXTERNAL FUNIN4
DATA PI, TWOPI, 3.14159265358979, 6.28318530717958/
DATA SQRT2PI, 2.5066282746310/
DATA PI1, TAU, 1.0
EVALUATE NORMALIZED EIGENFUNCTION
SCPTRMN = UNEGNF(N, RMJMN, ETA, RHO)/CAPNMN

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
TJGS = TAU
TAU = ARMISC(37)
CTJ = TJ(J)
EJ(J) = 0.
NOSCE = 2*I*FIX(TAU*ARMISC(36)/PI) + 1
WIDTH = ARMISC(36)/NOSCE
DO 25 IN = 1, NOSCE
RLOW = FLOAT(IN-1)*WIDTH
RUP = RLOW + WIDTH
CALL GAUSS(RLOW, RUP, VALUINT, FUNIN4, IORDGS)
EJ(J) = EJ(J) + REAL(VALUINT)
CONTINUE

EJ(J) = (2./SQRT2PI)*TJ(J)*EJ(J)
PJ(J) = ARMISC(29+J)*EJ(J)
GO TO 40

30 BTJ = ARMISC(36)*ARMISC(37)
IF (BTJ .EQ. 0.) TEMP1 = 1.
IF (BTJ .NE. 0.) TEMP1 = SIN(BTAU)/BTAU
PJ(J) = ARMISC(29+J)*TJ(J)*TEMP1/SQRT2PI

50 CONTINUE
CMPLX(0.,-L*ARMISC**2))
CTEMP1 = CEXP(CTEMP1)
IP = -1
IF(AR(1,7,2) .GT. 9.) GO TO 60
MSBI = AR(1,7,2)
GO TO 70
NPTS = AR(1,7,2)
CALL MTLUP(RHO,MSBI,1,NPTS,NPTS,1,IP,AR(3,1,2),AR(3,7,2))
IF(AR(1,8,2) .GT. 0.) GO TO 80
MSBE = AR(2,8,2)
GO TO 90
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
MSBA = AR(2,9,2)
GO TO 110
NPTS = AR(1,9,2)
CALL MTLUP(RHO,MSBA,1,NPTS,NPTS,1,IP,AR(3,1,2),AR(3,9,2))
IF(AR(1,10,2) .GT. 0.) GO TO 120
CHORD = AR(2,10,2)
GO TO 130
NPTS = AR(1,12,2)
CALL MTLUP(RHO,CHORD,1,NPTS,NPTS,1,IP,AR(2,2),AR(3,1,2),AR(3,12,2))
IF(ARMISC(25) .LE. 4.) GO TO 140
FRHO = AR(2,12,2)
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
NPTS = AR(1,11,2)
CALL MTLUP(RHO,ALFA,1,NPTS,NPTS,1,IP,AR(3,1,2),AR(3,11,2))
IF(MSBI .LE. MSBZ) TEMPI = 0.
IF(MSBI .GT. MSBZ) TEMPI = SQRT(MSBI-MSBI-MSBZ*MSBZ)
IF(MSBE .LE. MSBZ) TEMP2 = 0.
IF(MSBE .GT. MSBZ) TEMP2 = SQRT(MSBE-MSBZ-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(23))*(RHO-ARMISC(23))/
1. (2.*ARMISC(31)*J)*ARMISC(31)+J)
180 G(J) = (2.*J)**(1/2)
B1 = 1.
B2 = -ALFA*COTTHS
83 = -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,CAPM,K,COSTH,M,RHO,RNU,SINTH${}^3$,CAPF1)
$B1 = COTTHS$
$B2 = ALFA$
$B3 = FR40$
IF(ARMISC(38).EQ.0.) CALL LIFTFN3(RNU,B1,B2,B3,CAPF2)
IF(ARMISC(38).NE.0.) CALL NONCPT(B1,B2,B3,CHORD,CAPM,K,COSTH,M,RHO,RNU,SINTH${}^3$,CAPF2)
FACTIN4 = SCPTRMN*(GJ(1)*CAPF1 - GJ(2)*CAPF2)*MSBN*SINTH$
RETURN
150 IFORM = 2
FKMAX = 20.
DO 250 J = 1,2
DELT = ARMISC(31+J) / ARMISC(28)
INTEGJ(J) = .5
DO 240 K = 1,KMAX
TEMP1 = DELT*K / ARMISC(31+J)
TEMP2 = -(RHO - ARMISC(28))*TEMP1
CTE4P2 = CMPLX(0.,TEMP2)
CTE4P2 = CEXP(CTE4P2)*EXP(-.5*(DELT*K)**2)
COMPUTE FILOTAS L. R. F.$
TEMP2 = TEMP1**2 + (L*SINTH / RHO)**2
HRHOK = .5*CHORD*SORT(TEMP2)
PSI = ATAN(L*SINTH / (RHO*TEMP1))
CALL LIFTFN4(HRHOK,PSI,IFQRM,LIFT*,IERLFT4)
ACCUMULATE INTEGRAL
CLIFT4 = CONJG(LIFT4)
INTEGJ(J) = INTEGJ(J) + REAL(CTE4P2*CLIFT4)
CONTINUE
INTEGJ(J) = DELT*K*INTEGJ(J)
CONTINUE
FACTIN4 = 2.*SORT2PI*SCPTRMN*CTE4P1
1 = MSBN*SINTH*PJ(1)*BESIEJ(1)*INTEGJ(1)
2 = -MSBN*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 \( v \) and \( \lambda \), where:

\[
\begin{align*}
v &= \text{reduced temporal frequency} \\
\lambda &= \text{complex reduced spatial frequency} \\
LIFT &= \begin{cases} 
K_L(v, \lambda) & v \geq 0 \\
K_L(-v, -\lambda) & v < 0
\end{cases}
\]

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

\[
K_L(v, \lambda) = \begin{cases} 
\left[ J_0(\lambda) - i J_1(\lambda) \right] \left( \frac{H_1^{(2)}(\nu \lambda)}{H_1^{(2)}(\nu) + i H_0^{(2)}(\nu)} \right) + i \nu / \lambda J_1(\lambda) & \text{if } \nu \lambda \neq 0 \\
\frac{H_1^{(2)}(\nu)}{H_1^{(2)}(\nu) + i H_0^{(2)}(\nu)} + i \nu / 2 & \text{if } \nu \neq 0, \lambda = 0 \\
J_0(\lambda) - i J_1(\lambda) & \text{if } \nu = 0, \lambda \neq 0 \\
1 & \text{if } \nu = 0, \lambda = 0
\end{cases}
\]

\( -100 \leq \nu \leq 100 \) and \( \lambda \) is complex with \( |\lambda| \leq 100 \).
Method: The procedure is as follows:

1) If \( v > 0 \), go to step 3 to calculate \( K_L(v, \lambda) \).

2) If \( v < 0 \), go to step 3 to calculate \( K_L(-v, -\lambda) \).

3) If \( v \neq 0 \), calculate

\[
\text{HANKEL} = \frac{H_1^{(2)}(v)}{H_1^{(2)}(v)} + i \frac{H_0^{(2)}(v)}{H_1^{(2)}(v)}.
\]

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

5) If \( v \neq 0 \) and \( \lambda \neq 0 \), calculate \( \text{CAPKL} = \text{BESJLAM} \times \text{HANKEL} + i \frac{v}{\lambda} J_1(\lambda) / \lambda \) and go to step 9.

6) If \( v \neq 0 \) and \( \lambda = 0 \), calculate \( \text{CAPKL} = \text{HANKEL} + i \frac{v}{2} \) and go to step 9.

7) If \( v = 0 \) and \( \lambda \neq 0 \), calculate \( \text{CAPKL} = \text{BESJLAM} \) and go to step 9.

8) If \( v = 0 \) and \( \lambda = 0 \), let \( \text{CAPKL} = 1 \) 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 \( \text{CONLIFT} = \text{CAPKL} = K_L(v, \lambda) \), \( \text{LIFT} = \overline{\text{CONLIFT}} = \overline{K_L(v, \lambda)} \), and return.

11) Calculate \( \text{LIFT} = \text{CAPKL} = K_L(-v, -\lambda) \) and calculate \( \text{CONLIFT} = \overline{\text{LIFT}} = \overline{K_L(-v, -\lambda)} \).
Usage: CALLING SEQUENCE

COMPLEX CONLIFT,LAMDA,LIFT
.
.
.
CALL LIFTFN2(RNU,LAMDA,LIFT,CONLIFT)

Common Blocks: SCRATCH

Restrictions: -100 ≤ RNU ≤ 100 and |LAMDA| ≤ 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.

<table>
<thead>
<tr>
<th>n</th>
<th>λ</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0; e^{i\theta}, \theta = 10^\circ, 20^\circ, \ldots, 90^\circ; 5e^{i\theta}, \theta = 10^\circ, 20^\circ, \ldots, 90^\circ; 10e^{i\theta}, \theta = 10^\circ, 30^\circ, 60^\circ, 90^\circ</td>
</tr>
<tr>
<td>1</td>
<td>0; e^{i\theta}, \theta = 10^\circ, 30^\circ, 60^\circ, 90^\circ; 5e^{i\theta}, \theta = 10^\circ, 30^\circ, 60^\circ, 90^\circ</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0; e^{i\theta}, \theta = 10^\circ, 30^\circ, 60^\circ, 90^\circ; 5e^{i\theta}, \theta = 10^\circ, 30^\circ, 60^\circ, 90^\circ</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>100</td>
<td>0</td>
</tr>
</tbody>
</table>
START

\( n > 0 \)

10

IG0 = 1

SAVE \( n \) AND \( \Lambda \)
SET \( n = -n \)
AND \( \Lambda = -\Lambda \)
IG0 = 2

30

\( n < 0 \)

50

INITIALIZE

\( n = 0 \)

\( n \neq 0 \)

BSSLS AND BF4F
CALCULATE
HANKEL = \( \frac{H^2_n(n)}{H^2_n(n) + iH^2_0(n)} \)

60

\( \Lambda \neq 0 \)

ROCABES
CALCULATE
BESJLAM
= \( J_0(\Lambda) - i J_1(\Lambda) \)

70

\( \Lambda = 0 \)
\[ \text{CALCULATE } \text{CAPKL} = BESJLAM \times \text{HANKEL} + i\nu J_1(\nu) / \lambda \]

\[ \text{CALCULATE } \text{CAPKL} = \text{HANKEL} + i\nu / 2 \]

\[ \text{CALCULATE } \text{CAPKL} = \text{BESJLAM} \]

\[ \text{CALCULATE } \text{CAPKL} = 1 \]

\[ \lambda \neq 0 \]

\[ \lambda = 0 \]

\[ \nu \neq 0 \]

\[ \nu = 0 \]

\[ \text{IGO} = 1 \]

\[ \text{IGO} = 2 \]

\[ \text{CALCULATE } \text{CONLIFT} = \text{LIFT} = \text{CAPKL} = K_L(\nu, \lambda) \]

\[ \text{CALCULATE } \text{LIFT} = \text{CONLIFT} = K_L(\nu, \lambda) \]

\[ \text{RESET } \nu \text{ AND } \lambda \]

\[ \text{RETURN} \]
SUBROUTINE L I P T F N 2 ( R N U , L A M D A , L I F T , C O N L I F T )
REAL L A M D A , L A M D A I
COMPLEX L A M D A , L I F T , S A V E L A M
D I M E N S I O N B J L A M R ( 2 5 0 ) , B J L A M I ( 2 5 0 ) , B J 2 L A M R ( 5 0 ) , B J 2 L A M I ( 5 0 )
COMMON / S C R A T C H / B E S ( 1 0 0 0 )
EQUIVALENCE ( B E S ( 1 ) , B J L A M R ( 1 ) ) , ( B E S ( 2 5 1 ) , B J L A M I ( 1 ) )
EQUIVALENCE ( B E S ( 5 0 1 ) , B J L A M R ( 1 ) ) , ( B E S ( 5 5 1 ) , B J L A M I ( 1 ) )
I F ( I G O . G E . 3 . ) G O T O 3 0
I G O = 1
G O T O 5 0
C O N L I F T = C A P K L
L I F T = C O N J G ( C O N L I F T )
R E T J R N
S A V E R N U = R N U 
S A V E L A M = L A M D A
R N U = - R N U
L A M D A = - C O N J G ( L A M D A )
I G O = 2
G O T O 5 0
L I F T = C A P K L
C O N L I F T = C O N J G ( L I F T )
R N U = S A V E R N U
L A M D A = S A V E L A M
F T J R N
S I G N = - 1
N G = 1
A B S L A M = C A B S ( L A M D A )
I F ( I G O . E Q . 0 . ) G O T O 6 0
C A L L B S S L S ( R N U , B E S , N B , I E R R )
B J 1 R N U = B E S ( 1 )
B J 2 R N U = B E S ( 2 )
C A L L B F ( F R N U , B E S , N B , I E R R , I S I G N )
B Y R N U = - B E S ( 1 )
B Y 2 R N U = - B E S ( 2 )
H 1 R N U = C M P L X ( B J 1 R N U * B Y 1 R N U )
H 2 R N U = C M P L X ( B J 2 R N U * B Y 2 R N U )
H A N K E L = H 2 R N U / ( H 2 R N U + ( 0 . , 1 . ) * H 1 R N U )
2 I F ( A B S L A M . E Q . 0 . ) G O T O 7 0
L A M D A I = R E A L ( L A M D A )
B J 1 L A M = C M P L X ( B J L A M R ( 1 ) , B J L A M I ( 1 ) )
B J 2 L A M = C M P L X ( B J L A M R ( 2 ) , B J L A M I ( 2 ) )
B E S J L A M = B J 1 L A M - ( 0 . , 1 . ) * B J 2 L A M
1 0 I F ( I G O . E Q . 0 . ) G O T O 9 0
I F ( A B S L A M . E Q . 0 . ) G O T O 8 0
C A P K L = B E S J L A M * H A N K E L + ( 0 . , 1 . ) * R N U * B J 2 L A M / L A M D A
J O T O ( 2 0 , + 3 ) I G O
8 0 C A P K L = H A N K E L + ( 0 . , 1 . ) * R N U / 2.
G O T O ( 2 0 , + 4 0 ) I G O
9 0 I F ( A B S L A M . E Q . 0 . ) G O T O 1 0 0
C A P K L = B E S J L A M
G O T O ( 2 0 , + 4 0 ) I G O
1 0 0 C A P K L = ( 1 . , 0 . )
G O T O ( 2 0 , + 4 0 ) I G O
E N D
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):

\[ L(v) = S(v) - \text{COT} \beta \left[ F_f(v) + a F_\alpha(v) \right] \]

with \( v \) the reduced frequency, \( \beta \) the angle made by the velocity perturbation and the mean flow through the cascade, \( S \) the Sears function, \( f \) and \( a \) the ratio of maximum camber to the half-chord and the angle of attack, respectively, and \( F_f, F_\alpha \) the camber and angle of attack responses to the longitudinal component of the velocity perturbation.

\[ F_f(v) = \frac{H_o^2(v) + i H_1^2(v)}{-H_o^2(v) + i H_1^2(v)} \left[ J_0(v) - \frac{J_1(v)}{v} - i J_1(v) \right] \]

\[ \left[ J_0(v) - \frac{J_1(v)}{v} + i J_1(v) \right] + \frac{4}{v} J_1(v) \]

\[ F_\alpha(v) = J_0(v) + i J_1(v) \quad S(v) = \frac{-i}{2\pi} \frac{v}{H_o^2(v) + i H_1^2(v)} \]

191
It should be noted that these response functions, including the Sears function, are spectral functions under the convention:

\[
\tilde{g}(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \tilde{g}(\omega) e^{i\omega \tau} d\omega.
\]

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 \(\nu\), \(b_1\), \(b_2\), \(b_3\) 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 \(J_0(|\nu|), J_1(|\nu|), Y_0(|\nu|), Y_1(|\nu|)\).

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

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

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

7) Compute \(F_f(\nu) = \begin{cases} F_f(|\nu|) & \text{if } \nu > 0 \\ F_f(|\nu|) & \text{if } \nu < 0 \end{cases}\)
8) Compute $L(v) = b_1 S(v) + b_2 F_d(v) + b_3 F_f(v)$.

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 BF^4F.
SUBROUTINE LIFTFN3(RNU,B1,B2,B3,CAPLT)

C REFERENCE
H. NAUMANN AND H. YEN, LIFT AND PRESSURE FLUCTUATIONS OF
A CAMBERED AIRFOIL UNDER PERIODIC GUSTS AND APPLICATIONS
IN TURBOMACHINERY, JOURNAL OF ENGINEERING FOR POWER,

COMPLEX CAPLT,CTEMP1,FALPHNU,FFNU,H1RNU,H2RNU,SNU
COMMON/SCRATCH/8ES(100)
DATA PI,ISIGN/3.14159265358979,-1/
FOR RNU IN ABSOLUTE VALUE LESS THAN 1.E-10, SET THE
FUNCTIONS TO 1 AND RETURN

SNU = (1.,0.)
FALPHNU = (1.,0.)
FFNU = (2.,0.)
ABSNU = ABS(RNU)
IF( ABSNU .LE. 1.E-10 ) GO TO 30

COMPUTE FUNCTIONS FOR ABSNU AND THEN RNU

COMPUTE REQUIRED BESSEL FUNCTIONS

20 CALL BSLSL(ABSNU,BES,1.,IERR)
BJ1 = BES(1)
BJ2 = BES(2)
CALL BF4F(ABSNU,BES,1.,IERR,ISIGN)
BY1 = BES(1)
BY2 = BES(2)

COMPUTE HANKEL FUNCTIONS AN SNU

H1RNU = CMPLX(BJ1-BY1)
H2RNU = CMPLX(BJ2-BY2)
CTEMP1 = CMPLX(BJ1-BY2,-BJ2-BY1)
SNU = 1./(0.5*PI*ABSNU*CTEMP1)
IF( RNU.LT.0. ) SNU = CONJG(SNU )

COMPUTE FALPHNU

FALPHNU = CMPLX(BJ1+B2)
IF( RNU.LT.0. ) FALPHNU = CONJG(FALPHNU )

COMPUTE T, F, AND FALPHNU

CTEMP1 = (H1RNU + (0.,1.)*H2RNU) / (-H1RNU + (0.,1.)*H2RNU)
TEMP1 = BJ1 - BJ2/ABSNU
FFNU = CTEMP1*CMPLX(TEMP1,BJ2) - CMPLX(TEMP1,BJ2) + 4.*BJ2/ABSNU
IF( RNU.LT.0. ) FFNU = CONJG(FFNU)

30 CAPLT = B1*SNU + B2*FALPHNU + B3*FFNU
RETURN
END
Subroutine LIFTFN

**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 \( \nu \) and \( \beta \), where:

- \( \nu \) = reduced frequency
- \( \beta \) = gust yaw angle

and

\[
T(\nu, \beta) = \frac{1}{\frac{\pi}{2} \nu} \cdot \frac{I_0(\nu_2) + I_1(\nu_2)}{J_0(\nu_1 - 1 \cdot \nu_2) + I_1(\nu_1 - 1 \cdot \nu_2)}
\]

\[
F(\nu, \beta) = \frac{\pi}{2} \nu + e^{-i\nu_1} \left\{ \nu_2 K_1(\nu_2) \right\} - \nu \sec \beta \cdot G(\nu_2, \tan \beta)
\]

\[
G(x, \alpha) = \int_x^\infty e^{-i\alpha z} K_0(z) \, dz
\]

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:

\[
T(\nu, \beta) \approx e^{-i\nu} \left[ \frac{\pi \beta (1 + \frac{1}{2} \cos \beta)}{1 + 2\pi \nu (1 + \frac{1}{2} \cos \beta)} \right] \left[ 1 + \pi \nu (1 - \sin^2 \beta + \pi \nu \cos \beta) \right]^{\frac{1}{2}}
\]

The subroutine evaluates either the response function or the approximation in the return variable LIFT.4.
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 $|\nu| \leq \text{EPS} = 10^{-10}$, $T(|\nu|, \beta) = (1.,0.)$ and return.

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

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

6) When $|\beta - \pi/2| > \text{EPS} = 10^{-10}$, 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 $\nu_2$.

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 $\text{EPS} = 10^{-10}$ 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 $\nu_2$. 

197
12) Evaluate $F(|v|, \beta)$ and proceed to step 13.

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

$$F\left(|v|, \frac{\pi}{2}\right) = \cos v + v \sinh(v) + i \left[ v \cosh(v) - \sinh v \right]$$

where $\sinh$ and $\cosh$ are the hyperbolic sine and cosine integrals.

14) Compute the complex Bessel functions:

$$J_0(v, -iv)$$ and $$J_1(v, -iv).$$

15) Compute $T(v, \beta)$ using the appropriate formula, depending on whether $v$ 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: $\text{RNU} \geq 0$

Timing: The timing is dependent upon the input variables $v$ and $\beta$, 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.
CALLING LIST
RMU, THETA, IFORM
LIFT4, IERLFT4

ENTER

INITIALIZE ERROR RETURN CODE
IERLFT4 = 0

EPS = 10**(-10)

SET [\psi]', \psi/2 AND \psi/4
ABSNU, TEMP1, TEMP2

ABSNU : EPS

IFORM : 2

ABS (THETA - HALFPI) : EPS

SET SUB-INTERVAL WIDTH WIDTHI

INITIALIZE INTEGRAL VALUE, LOWER AND UPPER BOUNDS
CTEMP2 = (\sigma, \sigma)
RLOW, RUP

A
SET GAUSSIAN ORDER
FORMULA INDEX
IORDGS

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
BES10, BES11, BESK0, BESK1

CALCULATE F (|y|, |p|)
FRTH
COMPUTE SINE AND COSINE INTEGRAL

\[ S_1, C_1 \]

COMPUTE \( F(\lfloor \alpha \rfloor, 90) \)

\[ F_{RTH} \]

BES1 = 1, BES11 = 0.

COMPUTE BESSEL FUNCTIONS \( J_0(\lambda_1 - L/2) \) AND \( J_1(\lambda_1 - L/2) \)

COMPLETE T(\lfloor 0 \rfloor, 0)

\[ \text{LIFT4} \]

\[ \text{RETURN} \]
SUBROUTINE LIFTFN4(RNU, THETA, IFORM, LIFT4, IERLFT4)

COMMON/SCRATCH/BES(1000)
COMMON/ALPHA/ALPHA
COMPLEX CTEMP1, CTEMP2, CTEMP3, GRTHFCN, LIFT4
EXTERNAL GRTHFCN
DATA PI, HALFPI /3.14159265358975, 1.57079632679493/

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 FILTAS 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 TRIGONOMETRIC FACTOR UNTIL A CONTRIBUTION IS SMALL

ALPHA = TEMP3/TEMP4
WIDTHI = PI/AMAX1(1., ALPHA)
CTEMP2 = (0., 0.)
RLQW = 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 GAUS(RLOW, RUP, CTEMP3, GRTHFCN, IORDGS)
CTEMP2 = CTEMP3 + CTEMP3
IF(CABS(CTEMP3).LE.EPS*CABS(CTEMP2)) GO TO 130
RLQW = RUP
RUP = RUP + WIDTHI

GO TO 500
IF (RLQW = 1000.) 110, 110, 120
20 IERLFT4 = 1
120 CALL BESIK(TEMP2, ABSNU, BESK0, BESK1, IERBES)
CTEMP1 = CMPLEX( COS(TEMP1), -SIN(TEMP1) )
FRTH = HALFPI*ABSNU*CTEMP1*CMPLEX( 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 SICT(SI, CI, ABSNU)
SI = SI + HALFPI
FRTH = CMPLEX( 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(RNU) 410, 1000, 420
LIFT4 = (1./(-HALFPI*ABSNU + CONJG(FRTH))) * 
1 ((BESIO - BESI1)/CMPLEX(BES(1), BES(450)) + CMPLEX(BES(+51), -BES(2))))
GO TO 1000

LIFT4 = (1./HALFPI*ABSNU + FRTH)) * 
1 ((BESIO + BESI1)/CMPLEX(BES(1), BES(450)) - CMPLEX(BES(+51), -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 = CMPLEX( COS(TEMP1), -SIN(TEMP1) )*TEMP2

600 CONTINUE

RETURN
END
3.2.13 Function DISINT

**Purpose:** This routine evaluates the function:

\[
\left[ (A_0 \cos \phi - 1)^2 - (A^2 - 1)(p^2 - 1) \right]^{1/2} e^{-i \ell \phi}
\]

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

```plaintext
COMPLEX DISINT, VDISINT
COMMON/CDISINT/CAPADIS, RHOINC
COMMON/CFACT/M, N, RMUM, CAPMN, ETA, SIGN, L, CAPMN
.
.
VDISINT = DISINT(\phi)
```

**Accuracy:** The accuracy is of the algorithmic type and is dominated by the system routines SQRT, SIN, and COS.
EVALUATE THE EXPONENTIAL ARGUMENT
ARG = \(-L \times \Phi\)

EVALUATE THE EQUATION
DISINT

RETURN
COMPLEX FUNCTION DISINT(PHI)

PURPOSE: EVALUATE THE INTEGRAND OF THE INTEGRAL IN THE DISTORTION COEFFICIENT

COMMON/COSINT/CAPADIS*RHOMIC
COMMON/CFACT/ M, N, RMUNN, RAPHN, ETA, SIGN, L, CAPKYN

FUNPHI = CAPADIS*RHOMIC*COS(PHI) - 1.
FUNPHI = FUNPHI**2 - (CAPADIS**2 - 1.)*(RHOMIC**2 - 1.)
FUNPHI = SQRT(FUNPHI)
ARG = -L*PHI
DISINT = FUNPHI*CMPLX(COS(ARG), SIN(ARG))

RETURN
END
**3.2.14 Function FUNIN4**

*Purpose:* Complex Function FUNIN4 computes:

\[
\frac{(\omega T)^2}{2} e^{-\omega T} \cos(\omega T)
\]

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

```fortran
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:

\[ L_n'(\nu) = b_1 \cdot S(\nu) \cdot J(\kappa_{mn\sigma}^+) + b_2 \cdot J(\nu + \kappa_{mn\sigma}^-) + b_3 \cdot \left\{ J(\kappa_{mn\sigma}^+) F(\nu) + \frac{2J_1(\nu + \kappa_{mn\sigma}^+)}{\nu + \kappa_{mn\sigma}^+} \right\} \]

\[ - \frac{2}{\nu} \sum_{j=1}^{J_{\text{MAX}}} (-1)^j \cdot J_j(\nu) \left[ J_{j+1}(\kappa_{mn\sigma}^+) + J_{j-1}(\kappa_{mn\sigma}^-) \right] \]

where \( J(X) = J_0(X) + i J_1(X) \)

\[ S(X) = \frac{-1}{\frac{\pi X}{2} (-H_0^{(2)}(X) + i H_1^{(2)}(X))} \]

\[ T(X) = \frac{H_0^{(2)}(X) + i H_1^{(2)}(X)}{-H_0^{(2)}(X) + i H_1^{(2)}(X)} \]

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

\[ \kappa_{mn\sigma}^+ = \frac{C_2}{2} \left[ \kappa_{mn}^+ e_\phi - \frac{m}{\rho} e_z \right], \text{ and} \]

\[ b_1, b_2, b_3, C_2, \kappa_{mn}^+, e_\phi, m, \rho, \nu, e_z \text{ are input.} \]
Method:

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

2) Compute $H^2_0(|v|)$, $H^2_1(|v|)$ and $S(|v|)$.

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

4) Compute $F(|v|)$ and

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

5) Compute $\kappa^\pm_{\text{mn} \sigma}$.

6) Compute $J_0(\nu + \kappa^\pm_{\text{mn} \sigma})$, $J_1(\nu + \kappa^\pm_{\text{mn} \sigma})$,

and $\text{TEMP3} = \frac{2J_1(\nu + \kappa^\pm_{\text{mn} \sigma})}{\nu + \kappa^\pm_{\text{mn} \sigma}}$ where

if $\nu + \kappa^\pm_{\text{mn} \sigma} = 0$, then $J_0(0) = 1$, $J_1(0) = 0$, and $\text{TEMP3} = 1$.

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

8) Set $\text{JMAX} = \max\left(\left|\kappa^\pm_{\text{mn} \sigma}\right|, |v|\right) + 1$.

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

$$J_i(|\kappa^\pm_{\text{mn} \sigma}|), i = 0, 1, \ldots, \text{JMAX} + 1$$

using subroutine

$$\begin{cases} \text{BSSLS} & \text{if } \text{JMAX} + 1 \leq 100 \\ \text{BESNX} & \text{if } 100 < \text{JMAX} + 1 \end{cases}$$
10) Compute \( J_{j}(X) = (-1)^{j} J_{j}(|X|) \) where \( X = v, \kappa_{mn}^{\pm} \).

11) Compute 
\[
\text{SUM} = \frac{2}{\nu} \sum_{j=1}^{J_{\text{MAX}}} (-1)^{j} J_{j}(\nu) \left[ J_{j+1}(\kappa_{mn}^{+}) + J_{j-1}(\kappa_{mn}^{-}) \right]
\]
and go to step 14.

12) Compute necessary Bessel functions for steps 13 and 14, i.e., compute \( J_{0}(\kappa_{mn}^{+}), J_{1}(\kappa_{mn}^{+}) \) for step 14,

compute \( J_{2}(\kappa_{mn}^{+}) \) if \( \kappa_{mn}^{+} \neq 0 \) and

compute \( J_{1}(\nu) \) if \( \nu \neq 0 \).

13) Compute 
\[
\text{SUM} = \begin{cases} 
-1 & \text{if } \nu = 0 \text{ and } \kappa_{mn}^{+} = 0 \\
\frac{1}{\nu} \left[ J_{2}(\kappa_{mn}^{+}) + J_{0}(\kappa_{mn}^{+}) \right] & \text{if } \nu = 0 \text{ and } \kappa_{mn}^{+} \neq 0 \\
\frac{2}{\nu} J_{1}(\nu) & \text{if } \nu \neq 0 \text{ and } \kappa_{mn}^{+} = 0 
\end{cases}
\]

14) Compute \( L_{\nu}^{i}(\nu) \) 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.

\[ F(1 \nu 1) = 0 \]
\[ S(1 \nu 1) = 1 \]

FNU, SNU

\[ ISIGN = -1 \]

\[ \nu = 0? \]

RNU

Yes

NO

\[ \nu < 0 \]

RNU

Yes

Yes

\[ S(\nu) = S(1 \nu 1) \]

SNU

\[ S(\nu) = S^*(1 \nu 1) \]

SNU

Calculate \( F(1 \nu 1) \)

FNU

Calculate \( H_0^{(2)}(1 \nu 1), H_1^{(2)}(1 \nu 1) \)

BY1, BY2

Calculate \( J_0(1 \nu 1), J_1(1 \nu 1) \)

BJ1, BJ2

Calculate \( J_0(\nu + \kappa \pm \frac{m n \sigma}{\nu + \kappa \pm mn \sigma}) = 1 \)

\[ J_1(\nu + \kappa \pm \frac{m n \sigma}{\nu + \kappa \pm mn \sigma}) = 0 \]

BJ1, BJ2

Calculate \( 2J_1(\nu + \kappa \pm \frac{m n \sigma}{\nu + \kappa \pm mn \sigma}) \)

Temp 3

A

\[ F(\nu) = F(1 \nu 1) \]

FNU

\[ F(\nu) = F^*(1 \nu 1) \]

FNU

10

Calculate \( \kappa \pm \frac{m n \sigma}{\nu + \kappa \pm mn \sigma} \)

RKAPA

5
SUBROUTINE NONCPT(B1,B2,B3,C2,CAPMN,COSTHS,M,RHO,NU,SINTHS)
COMPLEX CAPLT,CTEMP1,FNU,H1RNU,H2RNU,SNU
COMMON/SCRATCH/BES(1003)
DATA ISIGN,P1/-1.314159265358979/
ABSNU = ABS(NU)
FNU = (0.,0.)
SNU = (1.,0.)
IF(NU.EQ.0.) GO TO 10
CALL BSSLST(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(BJ1-3Y2,-BJ2-3Y1)
SNU = 1./(-S1*ABSNU*CTEMP1)
IF(NU.LT.0.) SNU = CONJG(SNU)
CTEMP1 = (H1RNU + (0.,1.)*H2RNU) / (-H1RNU + (0.,1.)*H2RNU)
TEMP1 = BJ1 - BJ2/ABSNU
FNU = CTEMP1*CMPLX(TEMP1,-BY2) - CMPLX(TEMP1,BJ2)
IF(NU.LT.0.) FNU = CONJG(FNU)
RKAPA = (C2/2.)*CAPMN*COSTHS - (M*SINTHS/RHO)
TEMP1 = RNU + RKAPA
BJ1 = 1.
BJ2 = 0.
TEMP3 = 1.
IF(TEMP1.EQ.0.) GO TO 20
TEMP2 = ABS(TEMP1)
CALL BSSLST(TEMP2,BES,1,IERR)
BJ1 = BES(1)
BJ2 = BES(2)
IF(TEMP1.LT.0.) BJ2 = -BJ2
TEMP3 = (2.*BJ2)/TEMP1
20 IF(NU+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 BSSLST(ABSNU,BES,JMAX1,IERR)
CALL BSSLST(ABSKAPA,BES,JMAX2,IERR)
GO TO 40
30 CALL BESNX(JMAX1,ABSNU,BES)
CALL BESNX(JMAX2,ABSKAPA,BES)
GO TO 40
40 IF(NU.GT.0.) GO TO 50
30 5G = 1.2*A2.*
BES(1) = -BES(1)
50 BES(500+1) = -BES(500+1)
60 CTEMP1 = (0.,0.)
DO 70 J=1,JMAX
SIGN = 1.
IF( MOD(J,2) .NE. 2) SIGN = -1.
70 CTEMP1 = CTEMP1 + SIGN*BJ(J+1)*(BJ(502+J) + BJ(500+J))
SUM = (2./RNUI)*CTEMP1
GO TO 90
80 IF(RKAPA .EQ. 0.) BES(501) = 1.
IF(RKAPA .EQ. 0.) BES(502) = 0.
ABS(RKAPA) = ABS(RKAPA)
IF(RKAPA .NE. 0.) CALL BSSLABS(RKAPA,BES(501)+2,IERR)
IF(RKAPA .LT. 0.) BES(502) = -BES(502)
IF(RNU .NE. 0.) CALL BSSLAPS(RNU,BES(501)+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(5C1))
IF(RNU .NE. 0. AND. RKAPA .EQ. 0.) SUM = -(2./RNUI)*BES(2)
90 CTEMP1 = CMPLX(BES(501),BES(5G2))
CAPLHT = B1*RNU*CTEMP1 + B2*CMPLX(BJ1,BJ2)
1 = B3*(CTEMP1*RNU + 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:

\[ J'_ν (Z) Y'_ν (λZ) - J'_ν (λZ) Y'_ν (Z) \]

for \( λ \leq 5 \), where \( Z \sim β + \frac{p}{β} + q \frac{-p^2}{β^3} + \frac{r - 4pq + 2p^3}{β^5} \)

\[ μ = 4ν^2, \quad β = \frac{Sπ}{λ-1} \]

with \( S \) equal to the ordinal number of zero when \( ν = 0 \)

\[ p = \frac{μ + 3}{8λ}, \quad q = (μ^2 + 46μ - 63)(λ^3 - 1) \]

\[ r = \frac{(μ^3 + 185μ^2 - 2053μ + 1899)(λ^5 - 1)}{5(4λ)^5 (λ - 1)^5} \]

**Method:** The procedure is as follows:

1) Given \( η \), \( 0.2 ≤ η < 1 \), calculate \( λ = \frac{1}{η}, \quad λ - 1, \quad μ, \quad μ^2 \), and \( β \).

2) Calculate \( p, q, r \).

3) Calculate \( Z \).

4) Multiply \( Z \) by \( λ \) and output \( Zλ \).
Usage: CALLING SEQUENCE

CALL APROX1 (RM,NS,ETA,RZ)

INPUT

RM the value of v
NS the value of S, a positive integer
ETA the value of η, where λ = 1/η

OUTPUT

RZ the computed value of Zλ

Restrictions: .2 ≤ η < 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 \lambda, \lambda - 1
\mu, \mu ^2, \text{ AND } /3
FL, FL, U, UZ, B

CALCULATE P, Q, AND R
P, Q, R

CALCULATE Z
RZ

MULTIPLY BY \lambda
RZ

RETURN
SUBROUTINE APROXI(RM, NS, ETA, RZ)

PURPOSE APPLY APPROXIMATION FORMULA TO THE ZEROS OF THE EQUATION

\[ JP(M, X) \cdot YP(M, ETA \cdot X) - YP(M, X) \cdot JP(M, ETA \cdot X) = R(M, X) \]


REFERENCE HANDBOOK OF MATHEMATICAL FUNCTIONS EDITED BY M. ABRAMOWITZ AND I. STEGUM, NATIONAL BUREAU OF STANDARDS APPLIED MATHEMATICS SERIES 55 ISSUED JUNE 1964

VARIABLE DEFINITION
RM REAL ORDER M, SHOULD BE SMALL
ETA AT LEAST .2 (TO FIND NS-TH ZERO, ESP. FOR NS SMALL) AND LESS THAN 1 (THIS IS RATIO OF INNER TO OUTER RADII IN ANNULAR DUCT)
RZ THE APPROXIMATION TO THE ZERO

RESTRICTIONS THE RESTRICTION PLACED ON THE INPUT ABOVE GUARANTEES A VALID APPROX. TO NS-TH ZERO IS FOUND (PLACING ETA < .15, RM < .1, NS = 1 WILL GIVE ESTIMATE NOT OF FIRST BUT MUCH HIGHER ZERO).

DATA PI/01721622077325042051/

FL=1./ETA
FL1=FL-1.
U=4.*RM*RM
U2 = U*U
B = NS*PI/FL1
P = (U+3.)/(8.*FL)
Q = (U2+46.*U-63.)*(FL**3-1.)/(10.*((4.*FL)**3)*FL1)
R = (U**3+185.*U2-2053.*U+1899.)*(FL**5-1.)/(5.*((4.*FL)**5)*FL1)
RZ = B + P/B + (Q-6.*P**2)/8**3 + (R-4.*P*Q+2.*P**3)/3**5
RZ = RZ*FL

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:

\[ J'_0(X) Y'_0(\eta X) - Y'_0(X) J'_0(\eta X), \]

when \( \eta < 0.2 \), where \( J'_0 \) and \( Y'_0 \) are, respectively, the Bessel and Neumann functions of the zero \( \eta \)th 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, \( 0 \leq \eta < 0.2 \)

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.
CALL MTLUP
CALL
SET THE ZEROS
Y(1), Y(2), Y(3)

CALCULATE ZEROS
FOR η = .2 AND .3
Y(2), Y(3)

CALCULATE ZEROS
FOR η = .0
Y(1)

ETA: 0
RZ = Y(1)

MTLUP CALL

RETURN

CALL

NS: 1

APPROX 1 CALL
SUBROUTINE APROX2(RM, NS, ETA, RZ)

PURPOSE
FIND AN APPROXIMATION TO THE ZEROS OF

\[ JP(M, X) \times \left( YP(M, ETA \times X) - YP(M, X) \times JP(M, ETA \times X) \right) \]

IN THE REGION WHERE APROX1 FAILS
WHERE JP AND YP ARE THE DERIVATIVES OF THE BESSEL
FUNCTIONS OF THE FIRST AND SECOND KIND, RESPECTIVELY,
OF ARGUMENT X AND ETA \times X. ETA BETWEEN 0 AND 1.

REFERENCE
HANDBOOK OF MATHEMATICAL FUNCTIONS EDITED BY
M. ABRAMOWITZ AND I. STEGUM, NATIONAL BUREAU OF
STANDARDS APPLIED MATHEMATICS SERIES 55 ISSUED JUNE 1964

METHOD
SUBROUTINE APROX1 USES FORMULA 9.5.31 FROM THE REFERENCE
TO APPROXIMATE THE NS-TH ZERO OF THE ABOVE EQUATION
BUT THIS FAILS FOR ETA BELOW 0.2. TO CORRECT THIS PROBLEM
THIS ROUTINE IS PROVIDED. THE APPROXIMATION TO THE ZERO
IS FOUND BY QUADRATIC INTERPOLATION USING THE
APPROXIMATIONS FOR ETA \geq 0.2 AND 0.3 AND THE APPROXIMATION
FORMULA 9.5.13 OF THE REFERENCE FOR JP(M, X) = 0 WHICH
CORRESPONDS TO ETA = 0. FOR THE FIRST POSITIVE ZERO, THE
CORRESPONDING ZEROS ARE TABULATED BECAUSE THE
APPROXIMATION FORMULA ARE POOR FOR THE FIRST ZERO.

INPUT
VARIABLE DEFINITION
RM  BESSEL ORDER M AND IS 0 HERE
  +NS  THE NS-TH POSITIVE ZERO IS TO BE APPROXIMATED
  ETA  HUB TO TIP RATIO, 0 < ETA < 0.2

OUTPUT
RZ  APPROXIMATION TO ZERO

SUBPROGRAMS
APROX1  APPROXIMATION FOR ETA \geq 0.2
MTLUP  LRC LIBRARY INTERPOLATOR

DIMENSION X(3), Y(3)
DATA X/0, 0.2, 0.3/
NOETAT = NS
IF (NOETAT = 1) 10, 10, 20

10 Y(1) = 3.8317 $ Y(2) = 4.2357 $ Y(3) = 4.7058
GO TO 30

USE EXACT ZEROS FOR NS = 1. THE APPROXIMATION IS TOO POOR

10 Y(1) = 3.8317 $ Y(2) = 4.2357 $ Y(3) = 4.7058
GO TO 30

USE APROX1 FOR ZERO APPROXIMATION AT ETA = 0.2 AND 0.3

20 IF (ETA .EQ. 0.1) GO TO 25
CALL APROX1(RM, NOETAT, X(2), Y(2))
CALL APROX1(RM, NOETAT, X(3), Y(3))
APPLY REFERENCE FORMULA 9.5.13 FOR ETA 0 AND M=0

25  BETAP = (NOETAT +.25)*3.14159265
    BETAP8 = 8.*BETAP
    TERM1 = -3./BETAP8
    TERM2 = 36./(3.*BETAP 9**3)
    TERM3 = -113184./ (15.*BETAP8**5)
    TERM4 = 374532128./(105.*BETAP8**7)
    Y(1) = BETAP*TERM1+TERM2+TERM3+TERM4

30  IF(ETA) 40,40,50
40   RZ = Y(1)
   GO TO 60
50  IPA = -1
   CALL MTLUP(ETA,RZ,3,3,1,IPA,X,Y)
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 \( f(X) < \epsilon \), with \( \epsilon \) 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 \( f(X_n) \).

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 1.

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) ≠ START(2) ≠ START(3)

Timing: The timing is proportional to the number of iterations multi-
        plied 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 $3 \times 7.11 \times 10^{-15}$
USESML

INITIALIZE COUNTERS
IERJAR, IFLAT, ITER

SET INITIAL ITERATES AND FUNCTION VALUES
$X_1, X_2, X_3, F_1, F_2, F_3$

CHECK EACH PAIR $F_1, F_2,$ AND $F_3$ FOR EQUALITY

ARE TWO EQUAL?

NO

FUN CALL
PERTURB AND RECOMPUTE FUNCTION

YES

ERROR RETURN IERJAR

CONSTANT?

YES

NO

1000

A
CALL APPLY ITERATION FORMULA ROOT CALCULATE NEW FUNCTION VALUE FUNROOT

\[ |x_n - x_{n-1}| \leq \delta |x_n| \]

YES

\[ f_n \leq \epsilon \]

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, IERJAR)

PURPOSE
TO FIND THE ZERO OF A SINGLE REAL VALUED FUNCTION OF
A REAL VARIABLE BASED UPON JARRATT METHOD

INPUT
VARIABLE DEFINITION
START AN ARRAY OF THREE STARTING VALUES FOR X0
MAXITER MAXIMUM NUMBER OF ITERATIONS USED
TOLITER RELATIVE TOLERANCE ON THE CLOSENESS OF TWO
SUCCESSIVE ITERATES
TOLFUN TOLERANCE ON SMALLNESS OF FUNCTION VALUE
FUN EXTERNAL FUNCTION EVALUATOR, WHERE Y = FUN(X)

OUTPUT
ROOT ZERO CALCULATED
FUNROOT FUNCTION VALUE CORRESPONDING TO ROOT

ERROR RETURN
IERJAR 0 SUCCESSFUL EXECUTION
1 FAIL TO CONVERGE IN MAXITER ITERATIONS
2 FUNCTION APPEARS CONSTANT

REFERENCE
P. JARRATT AND D. NUOOS, THE USE OF RATIONAL FUNCTIONS
IN THE ITERATIVE SOLUTION OF EQUATIONS ON A DIGITAL
COMPUTER, THE COMPUTER JOURNAL, APRIL 1965, VOL. 8, NO. 1

DIMENSION START(3)
C THIS DATA STATEMENT DEFINES SMALLEST NUMBER
SIGNIFICANTLY ADDING TO 1.0

DATA SMALL/7.11E-15/
USESML = 3. * SMALL

INITIALIZE ERROR RETURN AND CONSTANT FUNCTION INDICATOR
AND ITERATION COUNTER
IERJAR = 0
IFLAT = 0
ITER = 0

SET STARTING FUNCTION VALUES
X1 = START(1)
X2 = START(2)
X3 = START(3)
F1 = FUN(X1)
F2 = FUN(X2)
F3 = FUN(X3)

CHECK FOR EQUAL FUNCTION VALUES, WHEN TWO ARE EQUAL
PERTURB THE STARTING VALUE AND RE-EVALUATE THE FUNCTION,
AND DO THIS AT MOST 3 TIMES

10 IF(F1.EQ.F2) GO TO 2
IF(F1.EQ.F3) GO TO 20
IF(F2.EQ.F3) GO TO 20
GO TO 50
20 IF(LAT = 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
   PERFORM JARRATT ITERATION
   X12 = X1-X2
   X13 = X1-X3
   ROOT = X1*(X12+X13+F1+F2-F3)/(X12+F3-F1)*F2*X13+F3
   FUNROOT = FUN(ROOT)
   CHECK FOR CONVERGENCE
   IF(ABS(ROOT-X1).LE.TOLITER*ABS(ROOT)) GO TO 1000
   IF(ABS(FUNROOT).LE.TOLFUN) GO TO 1000
   CHECK MAX ITERATION
   ITER = ITER+1
   IF(ITER-MAXITER).LE.1 GO TO 1000
55 IERJAR = 1
   GO TO 1000
   UPDATE LIST OF VARIABLE AND FUNCTION VALUES
   60 X3 = X2
   X2 = X1
   X1 = ROOT
   F3 = F2
   F2 = F1
   F1 = FUNROOT
   GO TO 10
1000 RETURN
   END
3.3.1 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:

\[
\frac{(B-A)2^N + 1}{(2N+1) [2N]!^3} \cdot \frac{(N!)^4 2^N + 1}{f(2N)(\xi)}
\]
CALLING LIST
A, B, ANS, FACTINT, INT

ENTER

SET INTERNAL HALF-WIDTH AND MIDPOINT:
H, X

SET THE 4, 8, AND 12-POINT ABCISSAS
Z4(), Z8(), Z12()

INT = 1
INT = 2
INT = 3

APPLY GAUSSIAN 4-POINT FORMULA
G8

APPLY GAUSSIAN 8-POINT FORMULA
G8

APPLY GAUSSIAN 12-POINT FORMULA
G8

SET THE ANSWER AND CORRECT FOR SIGN
ANS

RETURN
SUBROUTINE GAUSS(A,B,ANS,FACTINT,INT)

COMPLEX ANS,FACTINT,G4,G8,G12,Z1,Z2

4-, 8-, AND 12-POINT GAUSSIAN WEIGHTING COEFFICIENTS

DIMENSION W4(2),W8(4),W12(6), Z4(2),Z8(4), Z12(6)
DATA W4(1),W4(2),W8(1:1:1=1),W8(1:1:1=6),/52145158862546,
L,34785484537354,.36283738337362,.313706645877387,.2223810345337
L,1.10122836290376,.249147045813493,.233425355333956,
L.20316726723066,.6007328543346,.106939325995318,
L.647175336386512/

Y = A
H = (B-Y)/2.
SGN=SIGN(1..H)
H=A35(H)
X = Y + H*SGN

4-POINT ABSCISSAE
Z4(1)=.33981043384856#H
Z4(2)=.861136311594053#H

8-POINT ABSCISSAE
Z8(1)=.183434642495650#H
Z8(2)=.525532409916329#H
Z8(3)=.79866447413627#H
Z8(4)=.960298956497536#H

12-POINT ABSCISSAE
Z12(1)=.125233408511469#H
Z12(2)=.367831498998180#H
Z12(3)=.587317954286617#H
Z12(4)=.76902674194303#H
Z12(5)=.904117256370475#H
Z12(6)=.981560634246719#H

EVALUATE FUNCTION AND PERFORM WEIGHTED SUM

GO TO (10,20,40) INT

10 CONTINUE
G4+4*(W4(1)*FACTINT(X+Z4(1))*FACTINT(X-Z4(1)))+
W4(2)*(FACTINT(X+Z4(2))*FACTINT(X-Z4(2)))+
G8*8
GO TO 60

20 CONTINUE
G8 = C
DO 30 I=1,4
Z1=FACTINT(X+Z8(I))
Z2=FACTINT(X-Z8(I))
30 GB=GB+W3(I)+(Z1+Z2)
   GB=GB*H
   GO TO 60
40 CONTINUE
   GB=0
   DO 50 I=1,b
50   GB=GB+W12(I)*FACTINT(X+Z12(I))+FACTINT(X-Z12(I))
   G12=G12+H
   GB=G12
   CONTINUE
   ANS=GB
   IF(GB.ALT.0.) 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)

COMPLEX ANS,FACTIN2,G4,G8,G12,Z1,Z2
DIMENSION AR(MAXDIM,MAXJ,3),ARMISC(1)

4-, 8-, AND 12-POINT GAUSSIAN WEIGHTING COEFFICIENTS

DIMENSION W4(2),W8(4),W12(6),Z4(2),Z8(4),Z12(6)


C 4-POINT ABSCISSAE
Z4(1) = 3.399981043584856*H
Z4(2) = 8.61363111594053*H

C 8-POINT ABSCISSAE
Z8(1) = 1.83434642495650*H
Z8(2) = 5.255532409916329*H
Z8(3) = 7.96666477413627*H
Z8(4) = 9.60289856497536*H

C 12-POINT ABSCISSAE
Z12(1) = 1.25233408511469*H
Z12(2) = 3.67831498998180*H
Z12(3) = 5.87317954286617*H
Z12(4) = 7.69902674194305*H
Z12(5) = 9.04117256370475*H
Z12(6) = 9.81506534246719*H

EVALUATE FUNCTION AND PERFORM WEIGHTED SUM

GO TO (10,20,40) INT
10 CONTINUE
G4 = H*(W4(1)*(FACTIN2(X+Z4(1),ARMISC,MAXDIM,MAXJ,AR)
   + FACTIN2(X-Z4(1),ARMISC,MAXDIM,MAXJ,AR))
   + W4(2)*(FACTIN2(X+Z4(2),ARMISC,MAXDIM,MAXJ,AR))
   + FACTIN2(X-Z4(2),ARMISC,MAXDIM,MAXJ,AR))
G8 = G4
GO TO 60
20 CONTINUE
G8 = 0.
DO 30 I=1,4
Z1 = FACTIN2(X+Z1(I),ARMISC,MAXDIM,MAXJ,AR)
Z2 = FACTIN2(X-Z8(I),ARMISC,MAXDIM,MAXJ,AR)
30 G8 = G8 + W3(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)
    + FACTIN2(X-Z12(I),ARMISC,MAXDIM,MAXJ,AR))
GO TO 60
G12 = G12 + H
G8 = G12
G8 = 0.
CONTINUE
ANS = G8
IF(8-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 \( x \): \( J_0(x), J_1(x), \ldots, J_n(x) \).

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)

COMPUTES BESSEL FUNCTIONS OF THE FIRST KIND FOR POSITIVE
REAL ARGUMENT AND INTEGER ORDER

THIS IS A MODIFIED VERSION OF SUBROUTINE BSSLS. THE
RESTRICTION ON THE ORDER N, HAS BEEN REMOVED FROM THE
STANDARD LRC LIBRARY VERSION. THE STATEMENTS REMOVED
FROM THE LRC LIBRARY VERSION HAVE BEEN MADE INTO COMMENT
STATEMENTS DELIMITED BY **. THIS WAS DONE ON
MAY 8, 1973 BY GEORGE A. GRAF OF BCS.

DIMENSION F(1)
COMMON/FIX/NPR,NP,NPP
IERR=0
** NMAX=30 **
** IF(N.LE.NMAX)GO TO 701 **
** IER=1 **
** RETJRN **
701 MX=X
NO. OF FUNCTIONS COMPUTED
NPP=3*MX*12+10*ABS(N-1)/10
IF(X.GT.N)NPP=3.0*X*12.
IF(MOD(NPP,2).EQ.0)NPP=NPP+1
CLEAR COMPUTING AREA
DO 702 I=1,NPP
702 F(1)=0.0
IF(X.NE.0.0)GO TO 703
X=0
F(1)=1.0
RETJRN
703 IF(X.GE.1E-6)GO TO 703
SMALL VALUES OF X
J=Z**N/FACTORIAL M
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))
RETJRN
BACKWARD RECURSION
703 NP=NPP+1
NPR=NPP-1
F(NP-1)=1.0
F(NP)=0.0
DO 11 I=1,NPR
NP=NPP-I
XN=1P
11 F(NP)=2.0*XN/X*F(NP+1)-F(NP+2)
XN=F(1)
DO 7 I=1,NPP+2
7 XN=2.0*F(1)+XN
XN=1P/XN
DO 3 I=1,NPP
F(I)=XN*F(I)
3 CONTINUE
RETJRN
END
3.3.7 Subroutine BESN\(X\)

**Purpose:** BESN\(X\) 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, BESN\(X\) will compute:

\[
J_0(X), J_1(X), \ldots, J_N(X) \quad \text{(if } N > 0) \text{, or}
\]
\[
J_0(X), J_{-1}(X), \ldots, J_{-N}(X) \quad \text{(if } N < 0).
\]

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

**Step 1:** Determine index, \(NMAX\), to start backward recursion from the equations:

\[
IX = \max \left(5, \left|X\right|^{1/3}, 10\right)
\]

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

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:

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

**Step 3:** Calculate uncorrected \(J_K(X)\), \(K = NMAX, NMAX - 1, \ldots, 1, 0\) 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:

\[
REL = J_0(X) + 2 \sum_{j=1}^{N_o} J_{2j}(X)
\]

where \( N_o = \lfloor \text{NMAX}/2 \rfloor \), the largest integer, which is less than or equal to \( \text{NMAX}/2 \). 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 \( K = 1, 2, \ldots, N \) by dividing the uncorrected values by REL.

Step 6: In case the argument \( X \) is zero, let \( J_0(X) = 1, J_1(X) = J_2(X) = \ldots = J_{|N|}(X) = 0 \).

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

Usage:

**CALLING SEQUENCE**

```plaintext
REAL JNX
DIMENSION JNX ( \( \geq |N| + 1 \) )
.
.
.
CALL BESNX (N, X, JNX)
```

**INPUT**

- \( N \) integer order of the Bessel function
- \( X \) real argument of the Bessel function
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).

<table>
<thead>
<tr>
<th>Range of argument x and index n</th>
<th>Least number of significant figures</th>
</tr>
</thead>
<tbody>
<tr>
<td>x = .001(.001).009 n = 0(1)50</td>
<td>9</td>
</tr>
<tr>
<td>x = .01(.01).09 n = 0(1)50</td>
<td>12</td>
</tr>
<tr>
<td>x = .1(.1).9 n = 0(1)50</td>
<td>12</td>
</tr>
<tr>
<td>x = 1.(1.9) n = 0(1)50</td>
<td>12</td>
</tr>
<tr>
<td>x = 10.(10.)90 n = 0(1)50</td>
<td>5</td>
</tr>
<tr>
<td>x = 100.(100.)900 n = 0(1)50</td>
<td>5</td>
</tr>
</tbody>
</table>
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 BESNXT(N, X, JNX)

C PURPOSE EVALUATE THE BESSEL FUNCTION OF THE FIRST KIND FOR
C INTEGER ORDER AND REAL ARGUMENT USING THE RECURRENCE
C ALGORITHM OF MILLER AND 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

REAL JN, JNM, JNP, JNX, JX
DIMENSION JNX(1)
DATA SML,BIG/00001400030000000000000377677777777777777777/
IFIX,.EQ. 0. ) GO TO 130

C DETERMINE INDEX TO START BACKWARD RECURSION
ABSX • ABS(X)
IABSX • IFIX(ABSX)
IABSN1 • IABS(N) + 1
IFIABSX .LT. 3. ) 10*20
10 IX • 10
GO TO 30
20 IX • 5.*ABSX**.333333333333333
30 IF( IABSX .GE. IABSN1 ) GO TO 40
40 NMAX • IABSN1 + IX + 1
GO TO 50

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.
IABSN1 • IABS(N)
DO 14 G = I + NMAX
INDEX • NMAX - I + 1
IF( INDEX .GT. IABSN1) 60,70
60 NDEK • IABSN1
GO TO 80
70 NDEK • INDEX
80 JNM • (2.*FLOAT(INDEX) / X)*JN - JNP
JNX(INDEX) • JNM

C PREVENT FROM OVERFLOW AND UNDERFLOW
ABSJNM • ABS(JNM)
IF(ABSJNM .LE. OVER) GO TO 110
JN • JN / OVER
JNM • JNM / OVER

248
P.EL = REL / OVER
IF(INDEX .GT. IABSNI) GO TO 110
DO 90 II=INDEX,IIABSNI
J = II
JX = JNX(II)
ABSJNX = ABS(JX)
IF(ABSJNX .LE. UNDER) GO TO 100
JNX(II) = JNX(II) / OVER
GO TO 110
90 JABSNI = J - 1
110 IF(INDEX .NE. 1) GO TO 120
REL = REL + JNM
GO TO 130
110 L = MCD(INDEX*2)
IF(L .EQ. 0) GO TO 130
REL = REL + 2.*JNM
JNP = JN
JN = JNM
CALCULATE CORRECTED JNX
SMLREL = SML * REL
DO 15C I=1,IIABSNI
MAX = I
JX = JNX(I)
ABSJNX = ABS(JX)
IF(ABSJNX .LE. SMLREL) GO TO 160
JNX(I) = JNX(I) / REL
IF(IABSNI .EQ. IABSNI) GO TO 200
MAX = IABSNI + 1
DO 17C I=MAX,IABSNI
JNX(I) = 0.
GO TO 230
15C JNX(I) = 1.
DO 19C I=2,IIABSNI
19C JNX(I) = 0.
2 IF(N .GE. 0) RETURN
DO 210 I=2,IABSNI+2
JNX(I) = -JNX(I)
RETURN
END
3.3.8 Subroutine BESJLA

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

$$ \begin{align*} J_v(z) &= \sqrt{\frac{2}{\pi z}} \left\{ P(v,z) \cos X - Q(v,z) \sin X \right\} \\ P(v,z) &\approx \sum_{K=0}^{K_{\text{MAX}}} (-1)^K \frac{(v,2K)}{(2z)^{2K}} \\ &= 1 - \frac{(v-1)(v-9)}{2!(8z)^2} + \frac{(v-1)(v-9)(v-25)(v-49)}{4!(8z)^4} - \ldots \\ Q(v,z) &\approx \sum_{K=0}^{K_{\text{MAX}}} (-1)^K \frac{(v,2K+1)}{(2z)^{2K+1}} \\ &= \frac{v-1}{8z} - \frac{(v-1)(v-9)(v-25)}{3!(8z)^3} + \ldots \\ K_{\text{MAX}} &= \text{MAX}\left\{ \frac{1}{2} v + 1, 3 \right\} \\ v &= 4\nu^2 \\ x &= z - \left(\frac{1}{2} \nu + \frac{1}{4}\right)\pi \end{align*} \]
**Method:**
The procedure is as follows:

1) Evaluate $w$, $X$, $8Z$, and $(8Z)^{2}$.

2) Set $K_{\text{max}}$.

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

4) For each $K$, $K = 1, \ldots, K_{\text{max}}$, 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 $8 + 13 K_{\text{max}}$ multiplications, $2 + 2 K_{\text{max}}$ divisions, $2 + 2 K_{\text{max}}$ additions, and $3 + 6 K_{\text{max}}$ 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 \(20 \times e^{0.025} \times \text{ORD} \).
START

INPUT
N,X

X ≠ 0

CALCULATE
NMAX

50
CALCULATE
OVER,
UNDER

K = NMAX

180
SET J_0(X) = 1,
J_1(X) = J_2(X) =
... J_m(X) = 0

X = 0

CALCULATE
UNCORRECTED
J_{K-1}(X)

80

K

K = K - 1

K > 1

CALCULATE
REL

K = 1

200
CORRECT FOR
SIGN

CALCULATE
CORRECTED
J_K(X)
FOR ALL K

RETURN
SUBROUTINE BESJLA(ORD, ARG, BESVAL)

PURPOSE  
COMPUTE HANKELS ASYMPTOTIC EXPANSION FOR LARGE
ARGUMENT TO THE BESSEL FUNCTION

INPUT   
ORD    NON-NEGATIVE ORDER
ARG    POSITIVE ARGUMENT

OUTPUT  
BESVAL EVALUATION OF HANKELS EXPANSION

REFERENCE   
HANDBOOK OF MATHEMATICAL FUNCTIONS,
EDITED BY M. ABRAMOWITZ AND I. A. STIGUM
NATIONAL BUREAU OF STANDARDS APPLIED MATHEMATICS SERIES
NUMBER 55, ISSUED 1964,
SECTION 9.2, FORMULAS 9.2.5, 9.2.9 AND 9.2.10

BASIC FORMULA VARIABLES

TEMP1 = 4. * ORD * ORD
TEMP2 = ARG - (.5 * ORD + .25) * 3.14159265358979
TEMP3 = 8. * ARG
TEMP4 = TEMP3 * TEMP3

COMPUTE MAXIMUM SUM INDEX

KMAX = .5 * ORD + 1.
IF( KMAX.LT.3 ) KMAX=3

COMPUTE FORMULAS 9.2.9 AND 9.2.10

INITIALIZE SUM TO K=0 TERM

TEMP5 = 1.
TEMP6 = TEMP5
TEMP7 = (TEMP1-1.)/TEMP3
TEMP8 = TEMP7

ACCUMULATE THE SUM IN 9.2.9 AND 9.2.10 GENERATING EACH
ELEMENT IN THE SUM BY RECURSION

DO 100 K = 1, KMAX
TEMP9 = 4.*K
TEMP10 = 2.*K
TEMP6 = -TEMP6*(TEMP1-(TEMP9-3.1)**2)*(TEMP1-(TEMP9-1.1)**2)/
       1   ( TEMP10*(TEMP10-1.*)**2 )
TEMP5 = TEMP5 + TEMP6

TEMP8 = -TEMP3*(TEMP1-(TEMP9-1.1)**2)*(TEMP1-(TEMP9+1.1)**2)/
       1   ( (TEMP1+1.)**2 )
TEMP7 = TEMP7 + TEMP8
100 CONTINUE

COMPUTE HANKELS APPROXIMATION

BESVAL = SORT( 2./(3.14159265358979*ARG)) *
        1   ( TEMP5*COS(TEMP2) - TEMP7*SIN(TEMP2) )

RETURN
END
3.3.9 Subroutine BESIE

Purpose: This subroutine evaluates approximation formulas for $I_\nu(x)e^{-x}$. The formulas are given in reference 30. For $x < 1$, formula (9.6.7) is used. For $1 \leq x \leq A$, formula (9.6.52) where $A = \max (20|\ell|, 5)$, is used. For $A < x$, formula (9.6) is approximated. Formula (9.6.6), $I_\nu(x) = I_0(x)$, is indirectly used.

Method: The procedure is as follows:

1) If $\text{ARG} < 1$, calculate:

$$
\text{BESIEX} = \frac{1}{(\mid \ell \mid)_2} \frac{\text{ARG}}{\mid \ell \mid} e^{-\text{ARG}}
$$

and return.

2) If $\text{ARG} > \max (20|\ell|, 5)$, to to step 6.

3) Set $N\text{MAX} = \begin{cases} 
(2.5)(\text{ARG}) + 1 & \text{if } \text{ARG} < 50 \\
(1.25)(\text{ARG}) + 1 & \text{if } \text{ARG} \geq 50
\end{cases}$

Calculate $J_{\mid \ell \mid + n}(\text{ARG})$, $n = 0, 1, \ldots, N\text{MAX}$.

a) If $\mid \ell \mid + N\text{MAX} \leq 100$, use BSSLS.

b) If $\mid \ell \mid + N\text{MAX} > 100$ and $\text{ARG} > 20 e^{0.025(\mid \ell \mid + N\text{MAX})}$, use BESJLA.

c) If $\mid \ell \mid + N\text{MAX} > 100$ and $\text{ARG} > 20 e^{0.025(\mid \ell \mid + N\text{MAX})}$, use BESNX.

5) Calculate:

$$
\text{BESIEX} = e^{-\text{ARG}} \sum_{n=0}^{N\text{MAX}} \frac{(\text{ARG})^n}{n!} J_{\mid \ell \mid + n}(\text{ARG})
$$

and return.
6) Calculate:

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

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 \( \text{ARG} \leq \max (20|L|,5) \), then array \( \text{BES} \) having dimension 1000 implies \( |L| + 1.25(\text{ARG}) + 2 \leq 1000 \).

**Accuracy:** The accuracy is of the computer type.
where \( A = \max(20|L|, 5) \)

\[
\text{CALCULATE BESIX} \\
\text{RETURN}
\]

\[
\text{CALCULATE NMAX} \\
\text{RETURN}
\]

\[
\text{YES} \\
\text{LNMAX} = |L|, \quad \text{NMAX} \leq 100 \\
\text{RETURN}
\]

\[
\text{NO} \\
\text{ARG} \geq 20 \alpha \\
\text{NO} \\
\text{ARG} \geq 20 \alpha \\
\text{YES} \\
\text{BSSL} \\
\text{CALCULATE} \\
\text{BESJL} \\
\text{BESN}
\]

\[
\text{NO} \\
\text{ARG} \geq 20 \alpha \\
\text{NO} \\
\text{ARG} \geq 20 \alpha \\
\text{YES} \\
\text{CALCULATE PSUM} \\
\text{CALCULATE BESIX} \\
\text{RETURN}
\]
SUBROUTINE BESIE(L, ARG, BESIEX)
COMMON/SCRATCH/BESI1000/ 
DATA SQRT2PI/2.506628274631/ 
LABS = IABS(L)
IF( ARG .GE. 1.) GO TO 30 
IF( L .EQ. 0.) GO TO 10 
BESIEX = 1. 
BESIEX = BESIEX*EXP(-ARG) 
RETURN 
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(MAXO(20*LABS,5)) 
IF( ARG .GT. TEMPl ) GO TO 90 
IF( ARG .LE. 50.) NMAX = IFIX(2.5*ARG) + 1 
IF( ARG .GT. 50.) NMAX = IFIX(1.25*ARG) + 1 
NMAX = MAXO(NMAX,10) 
LNMAX = LABS + NMAX 
IF( LNMAX .GT. 100 ) GO TO 40 
CALL BSSL(N,ARG,BES,LABS,NMAX,IERR) 
GO TO 70 
TEMP1 = 20.*EXP(.025*LNMAX) 
IF( ARG .LT. TEMPl ) GO TO 60 
DO 90 N=LABS,LNMAX 
ORDER = FLOAT(N) 
CALL BESJ(N,ORDER,ARG,BES(N+1)) 
GO TO 70 
CALL BESNX(LNMAX,ARG,BES) 
TEMP1 = 1. 
PSUM = BES(LABS + 1) 
DO 30 N=1,NMAX 
TEMP1 = TEMPl*(ARG/FLOAT(N)) 
PSUM = PSUM + TEMP1*BES(LABS+N + 1) 
BESIEX = PSUM*EXP(-ARG) 
RETURN 
TEMP1 = SQRT2PI/SQRT(ARG) 
TEMP2 = ( 4.0*FLOAT(LABS)**2 - 1.0 ) / (B.*ARG) 
TEMP3 = TEMPl*( 4.0*( FLOAT(LABS)**2 ) - 9.0 ) / (16.*ARG) 
TEMP4 = TEMPl*( 4.0*( FLOAT(LABS)**2 ) - 25.0 ) / (24.*ARG) 
BESIEX = TEMPl*(1.0 - 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 $x \leq 3.75$ and $x > 3.75$, respectively.

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

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

5) Compute $K_1$ using formulas (9.8.7) and (9.8.8) of reference 30 when $x \leq 2$ 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

\[
\begin{align*}
\text{BESIO} & \quad I_0 \\
\text{BESI1} & \quad I_1 \\
\text{BESK0} & \quad K_0 \\
\text{BESK1} & \quad K_1
\end{align*}
\]
 depending on IFCN above

Error Return: IERR = 0 unless \( x \leq 0 \) 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 \( 2.2 \times 10^{-7} \).
Compute $K_0(x)$

Compute $K_0(x)$

Compute $K_1(x)$

Compute $K_1(x)$

Return J
SUBROUTINE BESIK(X,IFCN,BESI0,BESI1,BESK0,BESK1,IERR)

PURPOSE
COMPUTE THE MODIFIED BESSEL FUNCTIONS I AND K OF ORDERS 0 AND 1 USING POLYNOMIAL APPROXIMATIONS FROM REF.

REFERENCE
M. ABRAMOWITZ AND I. A. STEGUM, HANDBOOK OF MATHEMATICAL FUNCTIONS, NATIONAL BUREAU OF STANDARDS APPLIED MATHEMATICS SERIES 55

CHECK THE ARGUMENT

IERR = 0
IF( X ) 10,10,20
10 IERR = 1 GO TO 1200
20 IF( IFCN.EQ.2 ) GO TO 100
IF( IFCN.EQ.5 ) GO TO 100
IF( X.GT.3.75 ) GO TO 50

COMPUTE I FOR ORDER 0 AND ARGUMENT AT MOST 3.75 USING FORMULA 9.6.1 OF REF.

T = X/3.75
T2 = T*T
BESIO = 1. + T2*(3.5156229 + T2*(3.0899424 + T2*(1.23179 + T2*( 0.0360763 + T2*( 0.0058131))))
GO TO 100

COMPUTE I FOR ORDER 0 AND ARGUMENT AT LEAST 3.75 USING FORMULA 9.6.2 OF REF.

50 T1 = 3.75/X
BESIO = 1. + T1*(0.1328592 + T1*(-0.0225319 + T1*(-0.0015365
1. - T1*(0.00916281 + T1*(-0.0205776 + T1*(-0.0239537 + T1*(-0.014763)
2. + T1*(-0.03923771)))))))
BESIO = BESIO*EXP(X)/SQRT(X)

100 IF( IFCN.EQ.1 ) GO TO 200
IF( IFCN.EQ.4 ) GO TO 200
IF( X.GT.3.75 ) GO TO 150

COMPUTE I FOR ORDER 1 AND ARGUMENT AT MOST 3.75 USING FORMULA 9.9.3 OF REF.

T = X/3.75
T2 = T*T
BESI1 = 1. + T2*(0.13790374 + T2*(0.1498323 + T2*(0.0539933
1. + T2*(0.0258733 + T2*(0.03031532 + T2*(-0.032411))))))
BESI1 = BESI1*X
GO TO 200

COMPUTE I FOR ORDER 1 AND ARGUMENT AT LEAST 3.75
USING FORMULA 9.8.4 OF REF

150 T1 = 3.75/X
BESI1 = .39894228 + T1 * (.03936024 + T1 * (.00362018 + T1 * (.00163801
1 + T1 * (.01031555 + T1 * (.0095312 + 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

COMPUTE K FOR ORDER 1 AND ARGUMENT AT MOST 2.
USING REF. FORMULA 9.8.5

X2 = .25*X*X
BES<0 = -.57721566 + X2* (.42278420 + X2* (.2369756 + X2* (.03488590
1 + X2* (.00262698 + X2* (.03013750 + X2* .0000740 ))))
BES<0 = -ALOG(X*.5)*BES<0 + BES<0
GO TO 300

C

250 X2 = 2./X
BES<0 = 1.25331414 + X2* (.07832358 + X2* (.0219568 + X2* (.0106646
1 + X2* (.00262698 + X2* (.021251 + X2* .0000740 )))
BES<0 = BES<0 * EXP(-X) / SQRT(X)

C

300 IF (IFCN.EQ.4) GO TO 1000
IF (X.GT.2.) GO TO 353

COMPUTE K FOR ORDER 1 AND ARGUMENT AT MOST 2
USING REF FORMULA 9.8.7

X2 = .25*X*X
BES<1 = 1 + X2* (.15443144 + X2* (.67278579 + X2* (.18156897
1 + X2* (.01919402 + X2* (.301104 + X2* (.00034855))))
BES<1 = ALOG(X*.5)*BES<1 + BES<1/X
GO TO 1000

C

353 X2 = 2./X
BES<1 = 1 + X2* (.23498019 + X2* (.0365620 + X2* (.01530468
1 + X2* (.00703953 + X2* (.032561 + X2* (.00368245)))
BES<1 = BES<1 * 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 \(z = x + iy\), and the order \(W\) is real. For \(W > 0\), define \(N = \lfloor W \rfloor\) (the greatest integer less than or equal to \(W\)), \(a = W - N\), and the orders

\[
W = a + n \quad , \quad n = 0,1,\ldots, N
\]

and for \(W < 0\), define \(N = \lfloor W \rfloor + 1\), \(a = W - N\), and the orders

\[
W = a + n \quad , \quad n = 0,-1,\ldots, -|N|
\]

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:
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

\[
\begin{align*}
\text{BJRE(1)} & \quad \text{Re} J_{(\alpha + 0)}(z) & \quad \text{Re} J_{(\alpha + 0)}(z) \\
\text{BJRE(2)} & \quad \text{Re} J_{(\alpha + 1)}(z) & \quad \text{Re} J_{(\alpha + 1)}(z) \\
\vdots & \quad \vdots & \quad \vdots \\
\text{BJRE(N+1)} & \quad \text{Re} J_{(\alpha + N)}(z) & \quad \text{Re} J_{(\alpha + |N|)}(z)
\end{align*}
\]

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

ALGAMP; see reference 5

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: 2^55 octal, which includes all subroutines listed above under SUBPROGRAMS CALLED except ALGAMP
DIMENSION BJRE(1), BJIM(1), YRE(1), YIM(1)
CALL BEGIN(X, Y, N, K, R)
CALL MJRECUR(X, Y, ALPHA, K, R, BJRE, BJIM)
CALL MJSUM(ALPHA, K, BJRE, BJIM, SUMRA, SUMA)
CALL MFACTOR(X, Y, ALPHA, Q, R)
CALL MJNORM(K, Q, R, SUMRA, SUMA, BJRE, BJIM)
CALL MYSUM(X, Y, ALPHA, K, BJRE, BJIM, ASUMRA, ASUMA)
CALL MYGUP(X, Y, ALPHA, Q, R, ASUMRA, ASUMA, BJRE, BJIM, YRE, YIM)
CALL MYNORM(K, R, ASUMRA, ASUMA, BJRE, BJIM)
CALL MYNORM2(K, Q, R, ASUMRA, ASUMA, BJRE, BJIM, YRE, YIM)
CALL MYNORM2(K, Q, R, ASUMRA, ASUMA, BJRE, BJIM, YRE, YIM)
If (3*SQRT(1.5) - .5*E+14) 14*14*15
CALL MJNORM(X, Y, ALPHA, N, BJRE, BJIM, YRE, YIM)
GO TO 12
CALL MYNORM2(X, Y, ALPHA, N, BJRE, BJIM, YRE, YIM)
GO TO 12
RETURN
END

SUBROUTINE BEGIN(X, Y, N, K, R)
SSQ = X*X*Y*Y
XTEN = 2.0*SQRT(SSQ) + 20.0
NTEN = INTABS(IN)*10
M = MAXC(XTEN, NTEN)/2
K = 2*M + 1
R = K + 1
RETURN
END

SUBROUTINE MJRECUR(X, Y, ALPHA, K, R, BJRE, BJIM)
DIMENSION BJRE(1), BJIM(1)
R ALPHA = R + ALPHA
SSQ = X*X*Y*Y
BJRE(K+2) = 0.
BJIM(K+2) = 0.
BJRE(K+1) = 1.0*E-37
BJIM(K+1) = 0.0
TX = X/SSQ
TY = Y/SSQ
DO 1 = 1, K
L = K + 1 - I
R ALPHA = R ALPHA - 1.0
A = R ALPHA*TX
B = -2.0 R ALPHA*TY
1 CONTINUE
RETURN
END
BJRE(L1) = (A * BJRE(L1+1)) - (B * BJIM(L1+1)) - BJRE(L1+2)
BJIM(L1) = (B * BJRE(L1+1)) + (A * BJIM(L1+1)) - BJIM(L1+2)
RETURN
END

SUBROUTINE JSUM(ALPHA, BJR, BJIM, SUMRA, SUMIA)
DIMENSION BJRE(1), BJIM(1)

BO1 SUMRA = BJRE(3) * (ALPHA + 2, 0)
SUMIA = (ALPHA + 2, 0) * BJIM(3)
GRE = 1.0
GIM = 0.0
S = 1.0
DD61 = 5 * X, 2
S = S + 1.0
GRE = GRE * (ALPHA + S - 1.0) / S
GIM = GIM * (ALPHA + S - 1.0) / S
GRE = GRE
ALPTS = ALPHA + 2, 0 * S
GJR = GRE * BJRE(I)
GJI = GIM * BJIM(I)
GJR = GIM * BJRE(I)
SUMB = ALPTS * (GJR - GJI) * SUMRA
SUMB = ALPTS * (GJR + GJI) * SUMIA

THE FOLLOWING STATEMENT IS ADDED TO COMPENSATE THE DEFFICIENCY FOUND IN THE PURE IMAGINARY CASE

19 IF(ABS((SUMB / SUMRA) - 1.0) - .5E-14) 21, 21, 10
21 IF(SUMIA / SUMRA) 19, 21, 10
22 RETURN
END

SUBROUTINE FACTOR(X, Y, ALPHA, J, RI
CALL ALGAMF(ALPHA + 1, 0, 0, U, V)
CALL MC04LOG(X, Y, A1, B1)
A2 = ALPHA * A1
B2 = ALPHA * B1
A2 = A2
B2 = B2
CALL MC04EXP(A2, B2, A3, B3)
A4 = .5 * 3, 471 CO * ALPHA
CALL MC04EXP(A4, 0, A5, B5)
A6 = 3 * A5 - B3 * B5
B6 = 3 * A5 + A3 * B5
CALL MC04EXP(U, V, A7, B7)
Q = A5*A7 - B6*B7
R = B5*A7 + A6*B7
RETJRN
END

SUBROUTINE MCOMLOG(X, Y, A, B)
PI = 3.141592654
A = 5*LOG(X*X + Y*Y)
IF(X < 1*PI)
B = 5*PI
IF(Y < 12*3*8)
B = 3
GO TO 8
B = 0
GO TO 8
B = ATAN(Y/X)
GO TO 8
5 B = ATAN(Y/X)
IF(Y > 16*7*7)
B = 8 - PI
GO TO 8
B = 8 + PI
RETJRN
END

SUBROUTINE MCOMEXP(X, Y, A, B)
C = EXP(X)
A = C*SIN(Y)
B = C*SIN(Y)
RETJRN
END

SUBROUTINE JNORM(K, Q, R, SUMRA, SUMIA, BJRE, BJIM)
DIMENSION BJRE(1), BJIM(1)
S = ((SUMRA + BJRE(1)) + Q) - ((SUMIA + BJIM(1)) + R)
T = (((SUMRA + BJIM(1)) - Q) + ((SUMIA + BJRE(1)) + R)
IF(Abs(S) - Abs(T)) 
101 TS = S
TSSQ = S*(1 + TS*TS)
12 DO 13 I = 1, K
BJRE(I) = (BJRE(I) + BJIM(I))*TS/TSSQ
BJIM(I) = (BJIM(I) - BJRE(I)*TS)/TSSQ
13 BJRE(I) = BJRE(I)
GO TO 14
100 ST = S/T
STSQ = T*((ST*ST) + 1).
102 DO 13 I = 1, K
BJRE(I) = (BJRE(I)*ST + BJIM(I))/STSQ
BJIM(I) = (BJIM(I)*ST - BJRE(I))/STSQ
101 CONTINUE
SUBROUTINE SUM(X,Y,ALPHA,K,BJRE,BJIM,ASUMR,ASUMI)
DIMENSION BJRE(K),BJIM(1),
A1=ALPHA-1.0
A2=ALPHA-1.0
A3=ALPHA
GAMRE=-(2.0+ALPHA)/A1
GAMIM=0.
ASUMR=GAMRE*BJRE(3)
ASUMI=GAMIM*BJIM(3)
T=1.0
DO 500 I=5,K,2
 T=T+1.0
 B1=2.0*T
 F1=3.1+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
 BSUMR=GAMRE*BJRE(I)+GAMIM*BJIM(I)+ASUMR
 BSUMI=GAMIM*BJRE(I)+GAMRE*BJIM(I)+ASUMI
 IF(ABS((BSUMR/ASUMR)-1.0)-5.0E-14) 521,521,510
 521 IF(ASUMI)520,511,520
 520 IF(ABS((BSUMI/ASUMI)-1.0)-5.0E-14) 511,511,510
 510 ASUMR=ASUMR
 500 ASUMI=BSUMI
 511 RETURN
END

SUBROUTINE GNU(X,Y,ALPHA,K,R,ASUMR,ASUMI,BJRE,BJIM,YRE,YIM)
DIMENSION BJRE(K),BJIM(1),YRE(1),YIM(1)
P1=3.141592654
TP1=2.0/P1
QRE=TP1*(Q+R-R*R)
QIM=TP1*2.0*R
QRE=QRE+ASUMR-QIM*ASUMI
QIM=QIM+ASUMR+QRE+ASUMI
IF(ALPHA)1,3,1
3 CALL MYZERU(X,Y,ALPHA,ALPIM)
GO TO 720

720 CLEAR
RETURN
END
PALPHA = PI * ALPHA
COX = COS(PALPHA)
SIX = SIN(PALPHA)
ERE = COX / SIX
ABS3 = 2.0 * PALPHA * ALPHA
ALPRE = ERE - (ERE * PALPHA / ABS3)
ALPH = -QIM * ALPHA / ABS3

ONE = ALPRE * BJSRE(1) - ALPRE * BJIM(1) * DRE
YIM(1) = ALPRE * BJSRE(1) + ALPRE * BJIM(1) * DIM
RETN
END

SUBROUTINE MYZERO(X, Y, ALPRE, ALPH)
    TPI = 2.0 / 3.1 * 159265
    CALL MCOMLOG(X, Y, A, B)
    ALPRE = TPI * (-1.159315157 + A)
    ALPH = TPI * B
    RETRN
END

SUBROUTINE MYRONS(X, Y, BJSRE, BJIM, YRE, YIM)
    DIMENSION BJSRE(1), BJIM(1), YRE(1), YIM(1)
    SSQ = X * X + Y * Y
    TPI = 2.0 / 3.1 * 159265
    AZRE = TPI * X / SSQ
    AZIM = TPI * Y / SSQ
    ZRE = BJRE(2) * YRE(1) - BJIM(2) * YIM(1)
    ZIM = BJIM(2) * YRE(1) + BJRE(2) * YIM(1)
    BZRE = ZRE - AZRE
    BZIM = ZIM - AZIM
    BJS = BJRE(1) * BJIM(1) * BJIM(1)
    CZRE = BJRE(1) / BJS
    CZIM = BJIM(1) / BJS
    YRE(2) = BZRE * CZRE - BZIM * CZIM
    YIM(1) = BZIM * CZRE + BZRE * CZIM
    RETRN
END

SUBROUTINE MYNEGN(X, Y, ALPH, BJRE, BJIM, YRE, YIM)
    DIMENSION BJRE(1), BJIM(1), YRE(1), YIM(1)
    L = 1 + ABS(N) + 1
    SSQ = X * X + Y * Y
    TX = 2.0 * X / SSQ
    TY = 2.0 * Y / SSQ
    RALPHA = ALPH
    A = RALPHA * TX
    B = RALPHA * TY
    BJRE(2) = A * BJRE(1) - 8 * BJIM(1) - BJRE(2)
    BJIM(2) = A * BJRE(1) + 8 * BJIM(1) - BJIM(2)
YRE(1) = A*YIM(1) - B*YRE(1)
YIM(1) = B*YRE(1) + A*YIM(1)
DO 1 I = 1, L
ALPHA = ALPHA - 1.0
A = ALPHA * TX
B = - ALPHA * TY
BLRE(I) = A*BLRE(I-1) - B*BJM(I-1) - BLRE(I-2)
BJM(I) = B*BJRE(I-1) + A*BJM(I-1) - BJIM(I-2)
YRE(I) = A*YRE(I-1) - B*YIM(I-1) - YRE(I-2)
YIM(I) = B*YRE(I-1) + A*YIM(I-1) - YIM(I-2)
RETURN
END

SUBROUTINE MYRECUR(X, Y, N, BJRE, BJIM, YRE, YIM)
DIMENSION BJRE(1), BJIM(1), YRE(1), YIM(1)
SSQ = X*X + Y*Y
TPI = 2.0 / 3.1592654
AZRE = TPI * X / SSQ
AZIM = TPI * Y / SSQ
L = IABS(N) + 1
DO 1 I = 1, L
ZRE = BLRE(I) - BJM(I) - YRE(I-1)
ZIM = BJM(I) + YIM(I-1)
BZRE = ZRE - AZRE
BZIM = ZIM - AZIM
BJSO = BJRE(I-1) + BJM(I-1) + BJIM(I-1)
CZRE = BJRE(I-1) / BJSO
CZIM = BJM(I-1) / BJSO
YRE(I) = BZRE * CZRE - BZIM * CZIM
YIM(I) = BZIM * CZRE + BZRE * CZIM
RETURN
END

SUBROUTINE MYGNUP(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 .LE. 1) GO TO 720
CALL MYZERO(X, Y, ALPRE, ALPIM)
GO TO 720
ALPHA = PI * ALPHA
COS = COS(PALPHA)
SIN = SIN(PALPHA)
ERE = COS / SIX
ABSQ = 2.0 * ALPHA * ALPHA
ALPRE = ERE - (QRE * ALPHA / ABSQ)

BES42560
BES42570
BES42580
BES42590
BES42610
BES42620
BES42630
BES42640
BES42650
BES42660
BES42670
BES42710
BES42720
BES42730
BES42740
BES42750
BES42760
BES42770
BES42780
BES42790
BES42800
BES42810
BES42820
BES42830
BES42840
BES42850
BES42860
BES42870
BES42890
BES42900
BES42910
BES42920
BES42930
BES42940
BES42950
BES42960
BES42970
BES42980
BES42990
BES43000
BES43010
273
ALPIM = QIM*ALPHA/ABSQ
TRE = ALPRE*BJRE(1) - ALPIM*BJIM(2) + DRE
TIM = ALPIM*BJRE(1) + ALPRE*BJIM(2) + DIM
SSQ = X*X + Y*Y
ALPRE = -(Q*X*R*Y)/SSQ
ALPIM = -(X*R - Q*Y)/SSQ
YRE(2) = ALPRE*BJRE(1) - ALPIM*BJIM(2) + TRE
YIM(2) = ALPIM*BJRE(1) + ALPRE*BJIM(2) + TIM
RETURN
END

SUBROUTINE MYSUM(X,Y, ALPHA, BJRE, BJIM, ASUMR, ASUMI)
DIMENSION BJRE(1), BJIM(1)
A1 = ALPHA - 1.0
A2 = A1 - 1.0
A3 = A1 + ALPHA
ABSQ = A1*A1
ROLDRE = A1*(2.0*ALPHA)/ABSQ
RODIM = 0.
RES1 = -ROLDRE/2.0
VMS1 = 0.
SSQ = X*X + Y*Y
STORE = 3.*ALPHA*X/SSQ
STOI M = 3.*ALPHA*Y/SSQ
RES2 = (ROLDRE*STORE - ROLDIM*STOIM)
VMS2 = (ROLDRE*STOIM + ROLDIM*STORE)
ASUMR = RES1*BJRE(2)
ASUMR = ASUMR + RES2*BJIM(3)
ASUMI = RES1*BJIM(2)
ASUMI = ASUMI + RES2*BJRE(3) + VMS2*BJIM(3)
T = 1.0
DO 500 I = 3, K + 2
T = T + 1.0
BL = 2.0/T
FL = BL + 1.0 + ALPHA
F2 = A3*T
F3 = A1 + T
F5 = T - ALPHA
FB = A2*B1
G1 = F1*F2
H1 = G1*F3
P1 = F5*FB
CRE = H1/(P1*T)
TEMP = -CRE*ROLDRE
RNEWIM = CRE*RODIM
RNEWRE = TEMPP
RES1 = (RNEWRE - RNEWRE)/2.0
VMS1 = (ROLDIM - RNEWIM)/2.0
RES2 = (RNEWRE*STORE - RNEWIM*STOIM)
VMS2 = (RNEWRE*STOIM + RNEWIM*STORE)
BSUMR = RES1*BJRE(I+1) - VMS1*BJIM(I+1) + ASUMR
END

274
BSUMI = VMS1 * BJRE (I+1) * RES1 * BJIM (I+1) * ASUMI
BSUMR = RES2 * BJRE (I+2) - VMS2 * BJIM (I+2) * ASUMR
BSUMI = VMS2 * BJRE (I+2) * RES2 * BJIM (I+2) * ASUMI

521 IF (ABS((BSUMR / ASUMR) - 1.0) - 5E-14) 521, 521, 510
520 IF (ABS((BSUMI / ASUMI) - 1.0) - 5E-14) 511, 511, 510
510 ASUMR = BSUMR
ASUMI = BSUMI
RELDIM = RNEWIM
500 ROLDRE = RNEWRE
511 RETURN
END
3.3.12 Subroutine SICI

Purpose: This subroutine evaluates the sine and cosine integrals

\[ \text{Si}(X) = \int_{\infty}^{X} \frac{\sin(t)}{t} \, dt \quad X > 0 \]

\[ \text{Ci}(X) = \int_{\infty}^{X} \frac{\cos(t)}{t} \, dt \quad X > 0 \]

as taken from reference 55.
SUBROUTINE SICI(SI, CI, X)

PURPOSE
COMPUTES THE SINE AND COSINE INTEGRAL

USAGE
CALL SICI(SI, CI, X)

DESCRIPTION OF PARAMETERS
SI - THE RESULTANT VALUE SI(X)
CI - THE RESULTANT VALUE CI(X)
X - THE ARGUMENT OF SI(X) AND CI(X)

REMARKS
THE ARGUMENT VALUE REMAINS UNCHANGED

SUBROUTINES AND FUNCTION SUBPROGRAMS CALLED
NONE

METHOD
DEFINITION
SI(X) = INTEGRAL(SIN(T)/T)
CI(X) = INTEGRAL(COS(T)/T)
EVALUATION
REDUCTION OF RANGE USING SYMMETRY
DIFFERENT APPROXIMATIONS ARE USED FOR ABS(X) GREATER THAN 4 AND FOR ABS(X) LESS THAN 4.
REFERENCE
LUKE AND WiLM. *POLYNOMIAL APPROXIMATIONS TO INTEGRAL TRANSFORMS*, MATHEMATICAL TABLES AND OTHER AIDS TO COMPUTATIONS, VOL. 15, 1961, ISSUE 74, PP. 174-178.

***ROUTINE SICI(SI, CI, X)***

TEST ARGUMENT RANGE

\[ Z = A S ( X ) \]
\[ I F ( X < 1 ) \]
\[ Z = 1 - 1.4 \]
\[ Y = ( 1 - 2 ) * ( 1 + 2 ) \]
\[ S I = 1.570796326 \]
\[ I F ( Z > 3.2 ) \]
\[ C I = 1.675 \]
\[ R E T U R N \]
\[ S I = S I * ( ( 1.75314 E - 9 + 1.55892 E - 7 ) + 1.37416 E - 9 ) + 1.31939 E - 9 ) * Y + 1.37139 E - 9 ) * Y - 1.39192 E - 9 ) * Y - 1.31939 E - 9 ) * Z \]
\[ R E T U R N \]
\[ S I = S I + ( X ) \]
\[ Y = C I S ( Z ) \]
\begin{verbatim}
Z=4.*2
1*Z+4.987716E-2)*Z=3.332519E-3)*Z+2.314617E-2)*Z=1.134933E-5)*Z
2*Z+6.250111E-2)*Z+2.535393E-10
V=II((Z-5.108699E-3)*Z+2.319179E-2)*Z-6.537233E-2)*Z
CZ=Z*(S1+Y+V)
S1=-Z*(S1+U+Y+V)
IF(X>=5.05)
5 S1=3.141593E0-S1
6 RETJRN
END
\end{verbatim}
3.3.13 Function GRTHFCN

Purpose: This function evaluates:

\[ e^{i\alpha z} K_0(z) \]

where \( K_0 \) is the modified Bessel function.

Method: The procedure is as follows:

1) Compute \( e^{i\alpha z} \).

2) Compute \( K_0(z) \).

3) Compute the function value.

Usage: CALLING SEQUENCE

\[
\text{COMPLEX GRTHFCN,VALFCN} \\
\text{COMMON/ALPHA/ALPHA} \\
\vdots \\
\text{VALFCN = GRTHFCN(Z)}
\]

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
Calling List
Z

Start

Evaluate the exponential
CEXP

BESIK

Evaluate the modified Bessel function
BESKO

Complete the function value
GRTHFCN

Return
COMPLEX FUNCTION GRTHFCN(Z)
COMMON/ALPHA/ALPHA
COMPLEX CEXP

ARG = ALPHA*Z
CEXP = CMPLX( COS(ARG) , -SIN(ARG) )

CALL BESIK(Z,4,BESIO,BES1,BESKO,BESK1,IERBES)

GRTHFCN = CEXP*BESK0

RETURN
END
REFERENCES


