NAG Library Routine Document

S22AAF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

S22AAF returns a sequence of values for either the unnormalized or normalized Legendre functions of the first kind $P_n^m(x)$ or $\overline{P_n^m}(x)$ for real x of a given order m and degree $n = 0, 1, \ldots, N$.

2 Specification

```
SUBROUTINE S22AAF (MODE, X, M, NL, P, IFAIL)
INTEGER            MODE, M, NL, IFAIL
REAL (KIND=nag_wp) X, P(0:NL)
```

3 Description

S22AAF evaluates a sequence of values for either the unnormalized or normalized Legendre ($m = 0$) or associated Legendre ($m \neq 0$) functions of the first kind $P_n^m(x)$ or $\overline{P_n^m}(x)$, where x is real with $-1 \leq x \leq 1$, of order m and degree $n = 0, 1, \ldots, N$ defined by

$$P_n^m(x) = \left(1 - x^2\right)^{m/2} \frac{d^m}{dx^m} P_n(x) \qquad \text{if } m \geq 0,$$

$$P_n^m(x) = \frac{(n+m)!}{(n-m)!} P_n^{-m}(x) \qquad \text{if } m < 0 \quad \text{and}$$

$$\overline{P_n^m}(x) = \sqrt{\frac{(2n+1)}{2} \frac{(n-m)!}{(n+m)!}} P_n^m(x)$$

respectively; $P_n(x)$ is the (unassociated) Legendre polynomial of degree n given by

$$P_n(x) \equiv P_n^0(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} \left(x^2 - 1\right)^n$$

(the *Rodrigues formula*). Note that some authors (e.g., Abramowitz and Stegun (1972)) include an additional factor of $(-1)^m$ (the *Condon–Shortley Phase*) in the definitions of $P_n^m(x)$ and $\overline{P_n^m}(x)$. They use the notation $P_{mn}(x) \equiv (-1)^m P_n^m(x)$ in order to distinguish between the two cases.

S22AAF is based on a standard recurrence relation described in Section 8.5.3 of Abramowitz and Stegun (1972). Constraints are placed on the values of m and n in order to avoid the possibility of machine overflow. It also sets the appropriate elements of the array P (see Section 5) to zero whenever the required function is not defined for certain values of m and n (e.g., $m = -5$ and $n = 3$).

4 References

Abramowitz M and Stegun I A (1972) *Handbook of Mathematical Functions* (3rd Edition) Dover Publications

5 Arguments

1: MODE – INTEGER *Input*

On entry: indicates whether the sequence of function values is to be returned unnormalized or normalized.

MODE $= 1$
The sequence of function values is returned unnormalized.

MODE $= 2$
The sequence of function values is returned normalized.

Constraint: MODE $= 1$ or 2.

2: X – REAL (KIND=nag_wp) *Input*

On entry: the argument x of the function.

Constraint: abs(X) ≤ 1.0.

3: M – INTEGER *Input*

On entry: the order m of the function.

Constraint: abs(M) ≤ 27.

4: NL – INTEGER *Input*

On entry: the degree N of the last function required in the sequence.

Constraints:

NL ≥ 0;
if M $= 0$, NL ≤ 100;
if M $\neq 0$, NL $\leq 55 -$ abs(M).

5: P(0 : NL) – REAL (KIND=nag_wp) array *Output*

On exit: the required sequence of function values as follows:

if MODE $= 1$, P(n) contains $P_n^m(x)$, for $n = 0, 1, \ldots, N$;

if MODE $= 2$, P(n) contains $\overline{P_n^m}(x)$, for $n = 0, 1, \ldots, N$.

6: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this argument, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: IFAIL $= 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry IFAIL $= 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL $= 1$

On entry, abs(X) > 1.0,
or MODE $\neq 1$ or 2,
or NL < 0,
or NL > 100 when M $= 0$,
or abs(M) > 27,
or NL + abs(M) > 55 when M $\neq 0$.

IFAIL $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -399$

Your licence key may have expired or may not have been installed correctly.

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

7 Accuracy

The computed function values should be accurate to within a small multiple of the ***machine precision*** except when underflow (or overflow) occurs, in which case the true function values are within a small multiple of the underflow (or overflow) threshold of the machine.

8 Parallelism and Performance

S22AAF is not threaded in any implementation.

9 Further Comments

None.

10 Example

This example reads the values of the arguments x, m and N from a file, calculates the sequence of unnormalized associated Legendre function values $P_n^m(x), P_{n+1}^m(x), \ldots, P_{n+N}^m(x)$, and prints the results.

10.1 Program Text

```
    Program s22aafe

!     S22AAF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, s22aaf
```

```
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nlmax = 100, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: x
      Integer                          :: ifail, m, mode, n, nl
!     .. Local Arrays ..
      Real (Kind=nag_wp)               :: p(0:nlmax)
!     .. Executable Statements ..
      Write (nout,*) 'S22AAF Example Program Results'

!     Skip heading in data file
      Read (nin,*)

      Read (nin,*) mode, x, m, nl

      Write (nout,*)

      ifail = 0
      Call s22aaf(mode,x,m,nl,p,ifail)

      Write (nout,*) 'MODE      X      M    NL'
      Write (nout,*)
      Write (nout,99999) mode, x, m, nl
      Write (nout,*)

      Select Case (mode)
      Case (1)

        If (m==0) Then
          Write (nout,*) 'Unnormalized Legendre function values'
        Else
          Write (nout,*) 'Unnormalized associated Legendre function values'
        End If

      Case (2)

        If (m==0) Then
          Write (nout,*) 'Normalized Legendre function values'
        Else
          Write (nout,*) 'Normalized associated Legendre function values'
        End If

      End Select

      Write (nout,*) ' n      P(n)'

      Do n = 0, nl
        Write (nout,99998) n, p(n)
      End Do

99999 Format (1X,I3,4X,F5.1,2I6)
99998 Format (1X,I2,1X,1P,E12.4)
    End Program s22aafe
```

10.2 Program Data

```
S22AAF Example Program Data
 1   0.5   2   3  : Values of MODE, X, M and NL
```

10.3 Program Results

```
 S22AAF Example Program Results

 MODE      X      M    NL

   1      0.5     2     3

 Unnormalized associated Legendre function values
```

```
 n       P(n)
 0   0.0000E+00
 1   0.0000E+00
 2   2.2500E+00
 3   5.6250E+00
```