NAG Toolbox

nag_specfun_1f1_real (s22ba)

1 Purpose

nag_specfun_1f1_real (s22ba) returns a value for the confluent hypergeometric function ${}_1F_1(a;b;x)$ with real parameters a and b, and real argument x. This function is sometimes also known as Kummer's function $M(a,b,x)$.

2 Syntax

```
[m, ifail] = nag_specfun_1f1_real(a, b, x)
[m, ifail] = s22ba(a, b, x)
```

3 Description

nag_specfun_1f1_real (s22ba) returns a value for the confluent hypergeometric function ${}_1F_1(a;b;x)$ with real parameters a and b, and real argument x. This function is unbounded or not uniquely defined for b equal to zero or a negative integer.

The associated function nag_specfun_1f1_real_scaled (s22bb) performs the same operations, but returns M in the scaled form $M = m_f \times 2^{m_s}$ to allow calculations to be performed when M is not representable as a single working precision number. It also accepts the parameters a and b as summations of an integer and a decimal fraction, giving higher accuracy when a or b are close to an integer. In such cases, nag_specfun_1f1_real_scaled (s22bb) should be used when high accuracy is required.

The confluent hypergeometric function is defined by the confluent series

$$ {}_1F_1(a;b;x) = M(a,b,x) = \sum_{s=0}^{\infty} \frac{(a)_s x^s}{(b)_s s!} = 1 + \frac{a}{b}x + \frac{a(a+1)}{b(b+1)2!}x^2 + \cdots $$

where $(a)_s = 1(a)(a+1)(a+2)\dots(a+s-1)$ is the rising factorial of a. $M(a,b,x)$ is a solution to the second order ODE (Kummer's Equation):

$$ x\frac{d^2M}{dx^2} + (b-x)\frac{dM}{dx} - aM = 0. \qquad (1) $$

Given the parameters and argument (a,b,x), this function determines a set of safe values $\{(\alpha_i,\beta_i,\zeta_i) \mid i \leq 2\}$ and selects an appropriate algorithm to accurately evaluate the functions $M_i(\alpha_i,\beta_i,\zeta_i)$. The result is then used to construct the solution to the original problem $M(a,b,x)$ using, where necessary, recurrence relations and/or continuation.

Additionally, an artificial bound, $arbnd$ is placed on the magnitudes of a, b and x to minimize the occurrence of overflow in internal calculations. $arbnd = 0.0001 \times I_{\max}$, where $I_{\max} = $ x02bb. It should, however, not be assumed that this function will produce an accurate result for all values of a, b and x satisfying this criterion.

Please consult the NIST Digital Library of Mathematical Functions or the companion (2010) for a detailed discussion of the confluent hypergeoemtric function including special cases, transformations, relations and asymptotic approximations.

4 References

NIST Handbook of Mathematical Functions (2010) (eds F W J Olver, D W Lozier, R F Boisvert, C W Clark) Cambridge University Press

Pearson J (2009) Computation of hypergeometric functions *MSc Dissertation, Mathematical Institute, University of Oxford*

5 Parameters

5.1 Compulsory Input Parameters

1: **a** – REAL (KIND=nag_wp)

The parameter a of the function.

Constraint: $|\mathbf{a}| \leq arbnd$.

2: **b** – REAL (KIND=nag_wp)

The parameter b of the function.

Constraint: $|\mathbf{b}| \leq arbnd$.

3: **x** – REAL (KIND=nag_wp)

The argument x of the function.

Constraint: $|\mathbf{x}| \leq arbnd$.

5.2 Optional Input Parameters

None.

5.3 Output Parameters

1: **m** – REAL (KIND=nag_wp)

The solution $M(a, b, x)$.

Note: if overflow occurs upon completion, as indicated by **ifail** $= 2$, $|M(a, b, x)|$ may be assumed to be too large to be representable. **m** will be returned as $\pm R_{\max}$, where $R_{\max}$ is the largest representable real number (see nag_machine_real_largest (x02al)). The sign of **m** should match the sign of $M(a, b, x)$. If overflow occurs during a subcalculation, as indicated by **ifail** $= 5$, the sign may be incorrect, and the true value of $M(a, b, x)$ may or may not be greater than $R_{\max}$. In either case it is advisable to subsequently use nag_specfun_1f1_real_scaled (s22bb).

2: **ifail** – INTEGER

ifail $= 0$ unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$ (*warning*)

Underflow occurred during the evaluation of $M(a, b, x)$.
The returned value may be inaccurate.

ifail $= 2$

On completion, overflow occurred in the evaluation of $M(a, b, x)$.

ifail $= 3$

All approximations have completed, and the final residual estimate indicates some precision may have been lost.

ifail $= 4$

All approximations have completed, and the final residual estimate indicates no accuracy can be guaranteed.

ifail $= 5$

Overflow occurred in a subcalculation of $M(a, b, x)$.
The answer may be completely incorrect.

ifail $= 11$

Constraint: $|\mathbf{a}| \leq arbnd = \langle value \rangle$.

ifail $= 31$

Constraint: $|\mathbf{b}| \leq arbnd = \langle value \rangle$.

ifail $= 32$

On entry.
$M(a, b, x)$ is undefined when b is zero or a negative integer.

ifail $= 51$

Constraint: $|\mathbf{x}| \leq arbnd = \langle value \rangle$.

ifail $= -99$

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

ifail $= -399$

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

ifail $= -999$

Dynamic memory allocation failed.

7 Accuracy

In general, if **ifail** $= 0$, the value of M may be assumed accurate, with the possible loss of one or two decimal places. Assuming the result does not under or overflow, an error estimate res is made internally using equation (1). If the magnitude of res is sufficiently large, a nonzero **ifail** will be returned. Specifically,

ifail $= 0$ $\quad res \leq 1000\epsilon$
ifail $= 3$ $\quad 1000\epsilon < res \leq 0.1$
ifail $= 4$ $\quad res > 0.1$

where ϵ is the ***machine precision*** as returned by nag_machine_precision (x02aj).

A further estimate of the residual can be constructed using equation (1), and the differential identity,

$$\begin{aligned} \frac{dM(a,b,x)}{dx} &= \frac{a}{b}M(a+1,b+1,x), \\ \frac{d^2M(a,b,x)}{dx^2} &= \frac{a(a+1)}{b(b+1)}M(a+2,b+2,x). \end{aligned}$$

This estimate is however dependent upon the error involved in approximating $M(a+1, b+1, x)$ and $M(a+2, b+2, x)$.

Furthermore, the accuracy of the solution, and the error estimate, can be dependent upon the accuracy of the decimal fraction of the input parameters a and b. For example, if $b = b_i + b_r = 100 + 1.0\mathrm{e}{-6}$, then on a machine with 16 decimal digits of precision, the internal calculation of b_r will only be accurate to 8 decimal places. This can subsequently pollute the final solution by several decimal places without affecting the residual estimate as greatly. Should you require higher accuracy in such regions, then you should use nag_specfun_1f1_real_scaled (s22bb), which requires you to supply the correct decimal fraction.

8 Further Comments

None.

9 Example

This example prints the results returned by nag_specfun_1f1_real (s22ba) called using parameters $a = 13.6$ and $b = 14.2$ with 11 differing values of argument x.

9.1 Program Text

```
     function s22ba_example

fprintf('s22ba example results\n\n');

a = 13.6;
b = 14.2;
x = [-4.5:1:5.5];
m = x;

for i = 1:numel(x)
  [m(i), ifail] = s22ba(a, b, x(i));
end

fprintf('       x      M(%5.2f,%5.2f,x)       \n',a,b);
fprintf('%10.2f  %14.5e\n', [x;m]);
```

9.2 Program Results

```
     s22ba example results

       x      M(13.60,14.20,x)
     -4.50     1.38786e-02
     -3.50     3.56741e-02
     -2.50     9.20723e-02
     -1.50     2.38486e-01
     -0.50     6.19691e-01
      0.50     1.61478e+00
      1.50     4.21840e+00
      2.50     1.10449e+01
      3.50     2.89776e+01
      4.50     7.61660e+01
      5.50     2.00533e+02
```