# NAG Toolbox

# nag_specfun_1f1_real_scaled (s22bb)

## 1 Purpose

nag_specfun_1f1_real_scaled (s22bb) returns a value for the confluent hypergeometric function ${}_1F_1(a;b;x)$, with real parameters $a$ and $b$ and real argument $x$. The solution is returned in the scaled form ${}_1F_1(a;b;x) = m_f \times 2^{m_s}$. This function is sometimes also known as Kummer's function $M(a,b,x)$.

## 2 Syntax

```
[frm, scm, ifail] = nag_specfun_1f1_real_scaled(ani, adr, bni, bdr, x)
[frm, scm, ifail] = s22bb(ani, adr, bni, bdr, x)
```

## 3 Description

nag_specfun_1f1_real_scaled (s22bb) returns a value for the confluent hypergeometric function ${}_1F_1(a;b;x)$, with real parameters $a$ and $b$ and real argument $x$, in the scaled form ${}_1F_1(a;b;x) = m_f \times 2^{m_s}$, where $m_f$ is the real scaled component and $m_s$ is the integer power of two scaling. This function is unbounded or not uniquely defined for $b$ equal to zero or a negative integer.

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):

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

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.

For improved precision in the final result, this function accepts $a$ and $b$ split into an integral and a decimal fractional component. Specifically $a = a_i + a_r$, where $|a_r| \leq 0.5$ and $a_i = a - a_r$ is integral. $b$ is similarly deconstructed.

Additionally, an artificial bound, $arbnd$ is placed on the magnitudes of $a_i$, $b_i$ and $x$ to minimize the occurrence of overflow in internal calculations. $arbnd = 0.0001 \times I_{\mathrm{max}}$, where $I_{\mathrm{max}} =$ x02bb. It should, however, not be assumed that this function will produce an accurate result for all values of $a_i$, $b_i$ 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 hypergeometric 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: **ani** – REAL (KIND=nag_wp)

$a_i$, the nearest integer to $a$, satisfying $a_i = a - a_r$.

*Constraints*:

**ani** $= \lfloor$**ani**$\rfloor$;
$|$**ani**$| \leq arbnd$.

2: **adr** – REAL (KIND=nag_wp)

$a_r$, the signed decimal remainder satisfying $a_r = a - a_i$ and $|a_r| \leq 0.5$.

*Constraint*: $|$**adr**$| \leq 0.5$.

**Note**: if $|$**adr**$| < 100.0\epsilon$, $a_r = 0.0$ will be used, where $\epsilon$ is the ***machine precision*** as returned by nag_machine_precision (x02aj).

3: **bni** – REAL (KIND=nag_wp)

$b_i$, the nearest integer to $b$, satisfying $b_i = b - b_r$.

*Constraints*:

**bni** $= \lfloor$**bni**$\rfloor$;
$|$**bni**$| \leq arbnd$;
if **bdr** $= 0.0$, **bni** $> 0$.

4: **bdr** – REAL (KIND=nag_wp)

$b_r$, the signed decimal remainder satisfying $b_r = b - b_i$ and $|b_r| \leq 0.5$.

*Constraint*: $|$**bdr**$| \leq 0.5$.

**Note**: if $|$**bdr** $-$ **adr**$| < 100.0\epsilon$, $a_r = b_r$ will be used, where $\epsilon$ is the ***machine precision*** as returned by nag_machine_precision (x02aj).

5: **x** – REAL (KIND=nag_wp)

The argument $x$ of the function.

*Constraint*: $|$**x**$| \leq arbnd$.

## 5.2 Optional Input Parameters

None.

## 5.3 Output Parameters

1: **frm** – REAL (KIND=nag_wp)

$m_f$, the scaled real component of the solution satisfying $m_f = M(a,b,x) \times 2^{-m_s}$.

**Note**: if overflow occurs upon completion, as indicated by **ifail** $= 2$, the value of $m_f$ returned may still be correct. If overflow occurs in a subcalculation, as indicated by **ifail** $= 5$, this should not be assumed.

2: **scm** – INTEGER

$m_s$, the scaling power of two, satisfying $m_s = \log_2\left(\frac{M(a,b,x)}{m_f}\right)$.

**Note**: if overflow occurs upon completion, as indicated by **ifail** $= 2$, then $m_s \geq I_{\max}$, where $I_{\max}$ is the largest representable integer (see nag_machine_integer_max (x02bb)). If overflow occurs during a subcalculation, as indicated by **ifail** $= 5$, $m_s$ may or may not be greater than $I_{\max}$. In either case, **scm** $=$ x02bb will have been returned.

3: **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** $= 2$ (*warning*)

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

**ifail** $= 3$ (*warning*)

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: $|$**ani**$| \leq arbnd = \langle$*value*$\rangle$.

**ifail** $= 13$

Constraint: **ani** $= \lfloor$**ani**$\rfloor$.

**ifail** $= 21$

Constraint: $|$**adr**$| \leq 0.5$.

**ifail** $= 31$

Constraint: $|$**bni**$| \leq arbnd = \langle$*value*$\rangle$.

**ifail** $= 32$

On entry$b =$ **bni** $+$ **bdr** $= \langle$*value*$\rangle$.
$M(a, b, x)$ is undefined when $b$ is zero or a negative integer.

**ifail** $= 33$

Constraint: **bni** $= \lfloor$**bni**$\rfloor$.

**ifail** $= 41$

Constraint: $|$**bdr**$| \leq 0.5$.

**ifail** $= 51$

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

**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** $= 0$ $res \le 1000\epsilon$
**ifail** $= 3$ $1000\epsilon < res \le 0.1$
**ifail** $= 4$ $res > 0.1$

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

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

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

## 8 Further Comments

The values of $m_f$ and $m_s$ are implementation dependent. In most cases, if ${}_1F_1(a;b;x) = 0$, $m_f = 0$ and $m_s = 0$ will be returned, and if ${}_1F_1(a;b;x) = 0$ is finite, the fractional component will be bound by $0.5 \le |m_f| < 1$, with $m_s$ chosen accordingly.

The values returned in **frm** ($m_f$) and **scm** ($m_s$) may be used to explicitly evaluate $M(a,b,x)$, and may also be used to evaluate products and ratios of multiple values of $M$ as follows,

$$\begin{aligned} M(a,b,x) &= m_f \times 2^{m_s} \\ M(a_1,b_1,x_1) \times M(a_2,b_2,x_2) &= (m_{f1} \times m_{f2}) \times 2^{(m_{s1}+m_{s2})} \\ \frac{M(a_1,b_1,x_1)}{M(a_2,b_2,x_2)} &= \frac{m_{f1}}{m_{f2}} \times 2^{(m_{s1}-m_{s2})} \\ \ln|M(a,b,x)| &= \ln|m_f| + m_s \times \ln(2) \end{aligned}$$

## 9 Example

This example evaluates the confluent hypergeometric function at two points in scaled form using nag_specfun_1f1_real_scaled (s22bb), and subsequently calculates their product and ratio without having to explicitly construct $M$.

## 9.1 Program Text

```
function s22bb_example

fprintf('s22bb example results\n\n');

% n values of a and b
n     = 2;

% delta is perturbation on integer values for a and b
delta = 1e-4;
amod  = [-10    -10]; arem  = [ delta -delta];
bmod  = [ 30     30]; brem  = [-delta  delta];
x     = 25;

fprintf('        a         b         x         frm   scm    M(a,b,x)\n');
for j = 1:n
  [frmv(j), scmv(j), ifail] = s22bb( ...
                                     amod(j), arem(j), bmod(j), brem(j), x);
  a = amod(j) + arem(j);
  b = bmod(j) + brem(j);
  fprintf('%9.3f %9.3f %9.3f %11.3e %5d ', a, b, x, frmv(j), scmv(j));
  print_scale(frmv(j), scmv(j));
end

% Calculate the product M1*M2
frm = prod(frmv);
scm = sum(scmv);

fprintf('\nSolution product %24.3e %5d ', frm, scm);
print_scale(frm, scm);

% Calculate the ratio M1/M2
if frmv(2) ~= 0
  frm = frmv(1)/frmv(2);
  scm = scmv(1) - scmv(2);
  fprintf('\nSolution ratio   %24.3e %5d ', frm, scm);
  print_scale(frm, scm);
end

function print_scale(frm, scm)
  if scm < x02bl
    scale = frm*2^double(scm);
    fprintf('%11.3e\n', scale);
  else
    fprintf(' Not representable\n');
  end
```

## 9.2 Program Results

```
s22bb example results

       a         b         x         frm   scm    M(a,b,x)
  -10.000    30.000    25.000  -7.733e-01   -15  -2.360e-05
  -10.000    30.000    25.000  -7.732e-01   -15  -2.360e-05

Solution product                5.979e-01   -30   5.568e-10

Solution ratio                  1.000e+00     0   1.000e+00
```