NAG Library Function Document

nag_elliptic_integral_F (s21bec)

1 Purpose

nag_elliptic_integral_F (s21bec) returns a value of the classical (Legendre) form of the incomplete elliptic integral of the first kind.

2 Specification

```
#include <nag.h>
#include <nags.h>
double nag_elliptic_integral_F (double phi, double dm, NagError *fail)
```

3 Description

nag_elliptic_integral_F (s21bec) calculates an approximation to the integral

$$F(\phi \mid m) = \int_0^{\phi} \left(1 - m\sin^2\theta\right)^{-\frac{1}{2}} d\theta,$$

where $0 \leq \phi \leq \frac{\pi}{2}$, $m\sin^2\phi \leq 1$ and m and $\sin\phi$ may not both equal one.

The integral is computed using the symmetrised elliptic integrals of Carlson (Carlson (1979) and Carlson (1988)). The relevant identity is

$$F(\phi \mid m) = R_F(q, r, 1)\sin\phi,$$

where $q = \cos^2\phi$, $r = 1 - m\sin^2\phi$ and R_F is the Carlson symmetrised incomplete elliptic integral of the first kind (see nag_elliptic_integral_rf (s21bbc)).

4 References

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

Carlson B C (1979) Computing elliptic integrals by duplication *Numerische Mathematik* **33** 1–16

Carlson B C (1988) A table of elliptic integrals of the third kind *Math. Comput.* **51** 267–280

5 Arguments

1: **phi** – double *Input*

2: **dm** – double *Input*

On entry: the arguments ϕ and m of the function.

Constraints:

$0.0 \leq$ **phi** $\leq \frac{\pi}{2}$;
dm $\times \sin^2($**phi**$) \leq 1.0$;
Only one of $\sin($**phi**$)$ and **dm** may be 1.0.

Note that **dm** $\times \sin^2($**phi**$) = 1.0$ is allowable, as long as **dm** $\neq 1.0$.

3: **fail** – NagError * *Input/Output*

The NAG error argument (see Section 3.6 in the Essential Introduction).

6 Error Indicators and Warnings

NE_ALLOC_FAIL

Dynamic memory allocation failed.
See Section 3.2.1.2 in the Essential Introduction for further information.

NE_INTERNAL_ERROR

An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please contact NAG for assistance.

An unexpected error has been triggered by this function. Please contact NAG.
See Section 3.6.6 in the Essential Introduction for further information.

NE_NO_LICENCE

Your licence key may have expired or may not have been installed correctly.
See Section 3.6.5 in the Essential Introduction for further information.

NE_REAL

On entry, **phi** $= \langle value \rangle$.
Constraint: $0 \leq$ **phi** $\leq \frac{\pi}{2}$.
On failure, the function returns zero.

NE_REAL_2

On entry, **phi** $= \langle value \rangle$ and **dm** $= \langle value \rangle$; the integral is undefined.
Constraint: **dm** $\times \sin^2($**phi**$) \leq 1.0$.
On failure, the function returns zero.

NW_INTEGRAL_INFINITE

On entry, $\sin($**phi**$) = 1$ and **dm** $= 1.0$; the integral is infinite.
On failure, the function returns the largest machine number (see nag_real_largest_number (X02ALC)).

7 Accuracy

In principle nag_elliptic_integral_F (s21bec) is capable of producing full ***machine precision***. However round-off errors in internal arithmetic will result in slight loss of accuracy. This loss should never be excessive as the algorithm does not involve any significant amplification of round-off error. It is reasonable to assume that the result is accurate to within a small multiple of the ***machine precision***.

8 Parallelism and Performance

Not applicable.

9 Further Comments

You should consult the s Chapter Introduction, which shows the relationship between this function and the Carlson definitions of the elliptic integrals. In particular, the relationship between the argument-constraints for both forms becomes clear.

For more information on the algorithm used to compute R_F, see the function document for nag_elliptic_integral_rf (s21bbc).

If you wish to input a value of **phi** outside the range allowed by this function you should refer to Section 17.4 of Abramowitz and Stegun (1972) for useful identities. For example, $F(-\phi|m) = -F(\phi|m)$ and $F(s\pi \pm \phi|m) = 2sK(m) \pm F(\phi|m)$ where s is an integer and $K(m)$ is the complete elliptic integral given by nag_elliptic_integral_complete_K (s21bhc).

A parameter $m > 1$ can be replaced by one less than unity using $F(\phi|m) = \frac{1}{\sqrt{m}} F\left(\theta|\frac{1}{m}\right)$, $\sin\theta = \sqrt{m}\sin\phi$.

10 Example

This example simply generates a small set of nonextreme arguments that are used with the function to produce the table of results.

10.1 Program Text

```
/* nag_elliptic_integral_F (s21bec) Example Program.
 *
 * Copyright 2014 Numerical Algorithms Group.
 *
 * Mark 9, 2009.
 */
/* Pre-processor includes */
#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nags.h>
#include <nagx01.h>

int main(void)
{
  /*Integer scalar and array declarations */
  Integer  exit_status = 0;
  Integer  ix;
  /*Double scalar and array declarations */
  double   dm, f, phi, pi;
  NagError fail;

  INIT_FAIL(fail);

  printf("%s\n",
         "nag_elliptic_integral_F (s21bec) Example Program Results");
  printf("\n");
  printf("%s\n", "    phi    dm       nag_elliptic_integral_F");
  printf("\n");
  pi = nag_pi;
  for (ix = 1; ix <= 3; ix++)
    {
      phi = ix*pi/6.00e0;
      dm = ix*0.250e0;
      /*
       * nag_elliptic_integral_F (s21bec)
       * Elliptic integral of 1st kind, Legendre form, F( phi |m)
       */
      f = nag_elliptic_integral_F(phi, dm, &fail);
      if (fail.code != NE_NOERROR)
        {
          printf("Error from nag_elliptic_integral_F (s21bec).\n%s\n",
                  fail.message);
          exit_status = 1;
          goto END;
        }
      printf("%7.2f%7.2f%12.4f\n", phi, dm, f);
    }

 END:

  return exit_status;
}
```

10.2 Program Data

None.

10.3 Program Results

```
nag_elliptic_integral_F (s21bec) Example Program Results

    phi    dm       nag_elliptic_integral_F

    0.52   0.25      0.5294
    1.05   0.50      1.1424
    1.57   0.75      2.1565
```

Example Program
Classical (Legendre) Form of the Incomplete Elliptic Integral of the First Kind