NAG Library Function Document

nag_real_jacobian_elliptic (s21cac)

1 Purpose

nag_real_jacobian_elliptic (s21cac) evaluates the Jacobian elliptic functions sn, cn and dn.

2 Specification

```
#include <nag.h>
#include <nags.h>
void nag_real_jacobian_elliptic (double u, double m, double *sn, double *cn,
     double *dn, NagError *fail)
```

3 Description

nag_real_jacobian_elliptic (s21cac) evaluates the Jacobian elliptic functions of argument u and argument m,

$$\begin{aligned} \mathrm{sn}(u \mid m) &= \sin\phi, \\ \mathrm{cn}(u \mid m) &= \cos\phi, \\ \mathrm{dn}(u \mid m) &= \sqrt{1 - m\sin^2\phi}, \end{aligned}$$

where ϕ, called the *amplitude* of u, is defined by the integral

$$u = \int_0^{\phi} \frac{d\theta}{\sqrt{1 - m\sin^2\theta}}.$$

The elliptic functions are sometimes written simply as $\mathrm{sn}\,u$, $\mathrm{cn}\,u$ and $\mathrm{dn}\,u$, avoiding explicit reference to the argument m.

Another nine elliptic functions may be computed via the formulae

$$\begin{aligned} \mathrm{cd}\,u &= \mathrm{cn}\,u/\,\mathrm{dn}\,u \\ \mathrm{sd}\,u &= \mathrm{sn}\,u/\,\mathrm{dn}\,u \\ \mathrm{nd}\,u &= 1/\,\mathrm{dn}\,u \\ \mathrm{dc}\,u &= \mathrm{dn}\,u/\,\mathrm{cn}\,u \\ \mathrm{nc}\,u &= 1/\,\mathrm{cn}\,u \\ \mathrm{sc}\,u &= \mathrm{sn}\,u/\,\mathrm{cn}\,u \\ \mathrm{ns}\,u &= 1/\,\mathrm{sn}\,u \\ \mathrm{ds}\,u &= \mathrm{dn}\,u/\,\mathrm{sn}\,u \\ \mathrm{cs}\,u &= \mathrm{cn}\,u/\,\mathrm{sn}\,u \end{aligned}$$

(see Abramowitz and Stegun (1972)).

nag_real_jacobian_elliptic (s21cac) is based on a procedure given by Bulirsch (1960), and uses the process of the arithmetic-geometric mean (16.9 in Abramowitz and Stegun (1972)). Constraints are placed on the values of u and m in order to avoid the possibility of machine overflow.

4 References

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

Bulirsch R (1960) Numerical calculation of elliptic integrals and elliptic functions *Numer. Math.* **7** 76–90

5 Arguments

1: **u** – double *Input*
2: **m** – double *Input*

On entry: the argument u and the argument m of the functions, respectively.

Constraints:

abs(**u**) $\leq \sqrt{\lambda}$, where $\lambda = 1/$nag_real_safe_small_number;
if abs(**u**) $< 1/\sqrt{\lambda}$, abs(**m**) $\leq \sqrt{\lambda}$.

3: **sn** – double * *Output*
4: **cn** – double * *Output*
5: **dn** – double * *Output*

On exit: the values of the functions $\mathrm{sn}\,u$, $\mathrm{cn}\,u$ and $\mathrm{dn}\,u$, respectively.

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

The NAG error argument (see Section 2.7 in How to Use the NAG Library and its Documentation).

6 Error Indicators and Warnings

NE_ALLOC_FAIL

Dynamic memory allocation failed.
See Section 2.3.1.2 in How to Use the NAG Library and its Documentation for further information.

NE_BAD_PARAM

On entry, argument ⟨*value*⟩ had an illegal value.

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 2.7.6 in How to Use the NAG Library and its Documentation for further information.

NE_NO_LICENCE

Your licence key may have expired or may not have been installed correctly.
See Section 2.7.5 in How to Use the NAG Library and its Documentation for further information.

NE_REAL_2

On entry, |**m**| is too large when used in conjunction with the supplied argument **u**: |**m**| = ⟨*value*⟩ it must be less than ⟨*value*⟩.

On entry, |**u**| is too large: |**u**| = ⟨*value*⟩ it must be less than ⟨*value*⟩.

7 Accuracy

In principle the function is capable of achieving full relative precision in the computed values. However, the accuracy obtainable in practice depends on the accuracy of the standard elementary functions such as SIN and COS.

8 Parallelism and Performance

nag_real_jacobian_elliptic (s21cac) is not threaded in any implementation.

9 Further Comments

None.

10 Example

This example reads values of the argument u and argument m from a file, evaluates the function and prints the results.

10.1 Program Text

```
/* nag_real_jacobian_elliptic (s21cac) Example Program.
 *
 * NAGPRODCODE Version.
 *
 * Copyright 2016 Numerical Algorithms Group.
 *
 * Mark 26, 2016.
 */

#include <nag.h>
#include <stdio.h>
#include <nag_stdlib.h>
#include <nags.h>

int main(void)
{
  Integer exit_status = 0;
  double u, m, sn, cn, dn;
  NagError fail;

  INIT_FAIL(fail);

  /* Skip heading in data file */
#ifdef _WIN32
  scanf_s("%*[^\n]");
#else
  scanf("%*[^\n]");
#endif
  printf("nag_real_jacobian_elliptic (s21cac) Example Program Results\n");
  printf("      u            m            sn           cn           dn\n");
#ifdef _WIN32
  while (scanf_s("%lf %lf", &u, &m) != EOF)
#else
  while (scanf("%lf %lf", &u, &m) != EOF)
#endif
  {
    /* nag_real_jacobian_elliptic (s21cac).
     * Jacobian elliptic functions sn, cn and dn of real
     * argument
     */
    nag_real_jacobian_elliptic(u, m, &sn, &cn, &dn, &fail);
    if (fail.code != NE_NOERROR) {
      printf("Error from nag_real_jacobian_elliptic (s21cac).\n%s\n",
             fail.message);
      exit_status = 1;
      goto END;
    }
    printf("%12.3e %12.3e %12.3e %12.3e %12.3e\n", u, m, sn, cn, dn);
```

```
  }

END:
  return exit_status;
}
```

10.2 Program Data

```
nag_real_jacobian_elliptic (s21cac) Example Program Data
  0.2   0.3
  5.0  -1.0
 -0.5  -0.1
 10.0  11.0
```

10.3 Program Results

```
nag_real_jacobian_elliptic (s21cac) Example Program Results
      u            m            sn           cn           dn
   2.000e-01    3.000e-01    1.983e-01    9.801e-01    9.941e-01
   5.000e+00   -1.000e+00   -2.440e-01    9.698e-01    1.029e+00
  -5.000e-01   -1.000e-01   -4.812e-01    8.766e-01    1.012e+00
   1.000e+01    1.100e+01    2.512e-01    9.679e-01    5.528e-01
```