# NAG C 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

```
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 parameter $m$,

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

where $\phi$, 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 parameter $m$.

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 Parameters

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

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

*Constraints*:

abs(**u**) $\leq \sqrt{\lambda}$, where $\lambda = 1/$nag_real_safe_small_number (X02AMC);
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 parameter (see the Essential Introduction).

## 6 Error Indicators and Warnings

**NE_REAL_2**

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

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

**NE_BAD_PARAM**

On entry, parameter ⟨*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 consult NAG for assistance.

## 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 Further Comments

None.

## 9 Example

The example program reads values of the argument $u$ and parameter $m$ from a file, evaluates the function and prints the results.

## 9.1 Program Text

```
/* nag_real_jacobian_elliptic (s21cac) Example Program
 *
 * Copyright 2002 Numerical Algorithms Group.
 *
 * Mark 7, 2002.
 */

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

int main(void)
{
  double u, m, sn, cn, dn;

  /* Skip heading in data file */
  Vscanf("%*[^\n]");
  Vprintf("s21cac Example Program Results\n");
  Vprintf("      u            m            sn           cn           dn\n");
  while (scanf("%lf %lf", &u, &m) != EOF)
    {
      s21cac(u, m, &sn, &cn, &dn, NAGERR_DEFAULT);
      Vprintf("%12.3e %12.3e %12.3e %12.3e %12.3e\n", u, m, sn, cn, dn);
    }
  return EXIT_SUCCESS;
}
```

## 9.2 Program Data

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

## 9.3 Program Results

```
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
```