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,

\[ \begin{align*}
\text{sn}(u|m) &= \sin \phi, \\
\text{cn}(u|m) &= \cos \phi, \\
\text{dn}(u|m) &= \sqrt{1 - m \sin^2 \phi},
\end{align*} \]

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 sn \( u \), cn \( u \) and dn \( u \), avoiding explicit reference to the parameter \( m \).

Another nine elliptic functions may be computed via the formulae

\[ \begin{align*}
\text{cd} \ u &= \text{cn} \ u / \text{dn} \ u \\
\text{sd} \ u &= \text{sn} \ u / \text{dn} \ u \\
\text{nd} \ u &= 1 / \text{dn} \ u \\
\text{dc} \ u &= \text{dn} \ u / \text{cn} \ u \\
\text{nc} \ u &= 1 / \text{cn} \ u \\
\text{sc} \ u &= \text{sn} \ u / \text{cn} \ u \\
\text{ns} \ u &= 1 / \text{sn} \ u \\
\text{ds} \ u &= \text{dn} \ u / \text{sn} \ u \\
\text{cs} \ u &= \text{cn} \ u / \text{sn} \ u
\end{align*} \]

(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


5 Parameters

1: \( u \) – double \hspace{1cm} Input
2: \( m \) – double \hspace{1cm} Input

On entry: the argument \( u \) and the parameter \( m \) of the functions, respectively.

Constraints:

\[
\text{abs}(u) \leq \sqrt{\lambda}, \text{where} \quad \lambda = 1/\text{nag\_real\_safe\_small\_number} (X02AMC); \\
\text{if} \text{ abs}(u) < 1/\sqrt{\lambda}, \text{ abs}(m) \leq \sqrt{\lambda}.
\]

3: \( sn \) – double \hspace{1cm} Output
4: \( cn \) – double \hspace{1cm} Output
5: \( dn \) – double \hspace{1cm} Output

On exit: the values of the functions \( sn u \), \( cn u \) and \( dn u \), respectively.

6: \( fail \) – NagError \hspace{1cm} Input/Output

The NAG error parameter (see the Essential Introduction).

6 Error Indicators and Warnings

NE_REAL_2

On entry, \( \text{abs}(m) \) is too large when used in conjunction with the supplied argument \( u \):
\[
\text{abs}(m) = \langle\text{value}\rangle \text{ it must be less than} \langle\text{value}\rangle.
\]

On entry, \( \text{abs}(u) \) is too large: \( \text{abs}(u) = \langle\text{value}\rangle \text{ it must be less than} \langle\text{value}\rangle.\)

NE_BAD_PARAM

On entry, parameter \( \langle\text{value}\rangle \) 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

<table>
<thead>
<tr>
<th>u</th>
<th>m</th>
<th>sn</th>
<th>cn</th>
<th>dn</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.000e-01</td>
<td>3.000e-01</td>
<td>1.983e-01</td>
<td>9.801e-01</td>
<td>9.941e-01</td>
</tr>
<tr>
<td>5.000e+00</td>
<td>-1.000e+00</td>
<td>-2.440e-01</td>
<td>9.698e-01</td>
<td>1.029e+00</td>
</tr>
<tr>
<td>-5.000e-01</td>
<td>-1.000e-00</td>
<td>-4.812e-01</td>
<td>8.766e-01</td>
<td>1.012e+00</td>
</tr>
<tr>
<td>1.000e+01</td>
<td>1.100e+01</td>
<td>2.512e-01</td>
<td>9.679e-01</td>
<td>5.528e-01</td>
</tr>
</tbody>
</table>