NAG Library Function Document

nag_jacobian_elliptic (s21cbc)

1 Purpose

nag_jacobian_elliptic (s21cbc) evaluates the Jacobian elliptic functions $\mathrm{sn}\, z$, $\mathrm{cn}\, z$ and $\mathrm{dn}\, z$ for a complex argument z.

2 Specification

```
#include <nag.h>
#include <nags.h>
```

```
void nag_jacobian_elliptic (Complex z, double ak2, Complex *sn, Complex *cn,
     Complex *dn, NagError *fail)
```

3 Description

nag_jacobian_elliptic (s21cbc) evaluates the Jacobian elliptic functions $\mathrm{sn}(z \mid k)$, $\mathrm{cn}(z \mid k)$ and $\mathrm{dn}(z \mid k)$ given by

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

where z is a complex argument, k is a real argument (the *modulus*) with $k^2 \leq 1$ and ϕ (the *amplitude* of z) is defined by the integral

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

The above definitions can be extended for values of $k^2 > 1$ (see Salzer (1962)) by means of the formulae

$$\begin{aligned} \mathrm{sn}(z \mid k) &= k_1\,\mathrm{sn}(kz \mid k_1) \\ \mathrm{cn}(z \mid k) &= \mathrm{dn}(kz \mid k_1) \\ \mathrm{dn}(z \mid k) &= \mathrm{cn}(kz \mid k_1), \end{aligned}$$

where $k_1 = 1/k$.

Special values include

$$\begin{aligned} \mathrm{sn}(z \mid 0) &= \sin z \\ \mathrm{cn}(z \mid 0) &= \cos z \\ \mathrm{dn}(z \mid 0) &= 1 \\ \mathrm{sn}(z \mid 1) &= \tanh z \\ \mathrm{cn}(z \mid 1) &= \mathrm{sech}\, z \\ \mathrm{dn}(z \mid 1) &= \mathrm{sech}\, z. \end{aligned}$$

These functions are often simply written as $\mathrm{sn}\, z$, $\mathrm{cn}\, z$ and $\mathrm{dn}\, z$, thereby avoiding explicit reference to the argument k. They can also be expressed in terms of Jacobian theta functions (see nag_jacobian_theta (s21ccc)).

Another nine elliptic functions may be computed via the formulae

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

(see Abramowitz and Stegun (1972)).

The values of $\mathrm{sn}\, z$, $\mathrm{cn}\, z$ and $\mathrm{dn}\, z$ are obtained by calls to nag_real_jacobian_elliptic (s21cac). Further details can be found in Section 9.

4 References

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

Salzer H E (1962) Quick calculation of Jacobian elliptic functions *Comm. ACM* **5** 399

5 Arguments

1: **z** – Complex *Input*

On entry: the argument z of the functions.

Constraints:

abs(**z**.re) $\leq= \sqrt{\lambda}$;
abs(**z**.im) $\leq \sqrt{\lambda}$, where $\lambda = 1/$nag_real_safe_small_number.

2: **ak2** – double *Input*

On entry: the value of k^2.

Constraint: $0.0 \leq$ **ak2** ≤ 1.0.

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

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

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

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

6 Error Indicators and Warnings

NE_BAD_PARAM

On entry, argument $\langle$*value*$\rangle$ had an illegal value.

NE_COMPLEX

On entry, $|$**z**.$im|$ is too large: $|$**z**.$im| = \langle$*value*$\rangle$. It must be less than $\langle$*value*$\rangle$.

On entry, $|$**z**.$re|$ is too large: $|$**z**.$re| = \langle$*value*$\rangle$. It must be less than $\langle$*value*$\rangle$.

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.

NE_REAL

On entry, **ak2** = ⟨*value*⟩.
Constraint: **ak2** ≤ 1.0.

On entry, **ak2** = ⟨*value*⟩.
Constraint: **ak2** ≥ 0.0.

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

Not applicable.

9 Further Comments

The values of $\mathrm{sn}\, z$, $\mathrm{cn}\, z$ and $\mathrm{dn}\, z$ are computed via the formulae

$$\mathrm{sn}\, z = \frac{\mathrm{sn}(u,k)\,\mathrm{dn}(v,k')}{1-\mathrm{dn}^2(u,k)\mathrm{sn}^2(v,k')} + i\frac{\mathrm{cn}(u,k)\,\mathrm{dn}(u,k)\,\mathrm{sn}(v,k')\,\mathrm{cn}(v,k')}{1-\mathrm{dn}^2(u,k)\mathrm{sn}^2(v,k')}$$

$$\mathrm{cn}\, z = \frac{\mathrm{cn}(u,k)\,\mathrm{cn}(v,k')}{1-\mathrm{dn}^2(u,k)\mathrm{sn}^2(v,k')} - i\frac{\mathrm{sn}(u,k)\,\mathrm{dn}(u,k)\,\mathrm{sn}(v,k')\,\mathrm{dn}(v,k')}{1-\mathrm{dn}^2(u,k)\mathrm{sn}^2(v,k')}$$

$$\mathrm{dn}\, z = \frac{\mathrm{dn}(u,k)\,\mathrm{cn}(v,k')\,\mathrm{dn}(v,k')}{1-\mathrm{dn}^2(u,k)\mathrm{sn}^2(v,k')} - i\frac{k^2\,\mathrm{sn}(u,k)\,\mathrm{cn}(u,k)\,\mathrm{sn}(v,k')}{1-\mathrm{dn}^2(u,k)\mathrm{sn}^2(v,k')},$$

where $z = u + iv$ and $k' = \sqrt{1-k^2}$ (the *complementary modulus*).

10 Example

This example evaluates $\mathrm{sn}\, z$, $\mathrm{cn}\, z$ and $\mathrm{dn}\, z$ at $z = -2.0 + 3.0i$ when $k = 0.5$, and prints the results.

10.1 Program Text

```
/* nag_jacobian_elliptic (s21cbc) Example Program.
 *
 * Copyright 2000 Numerical Algorithms Group.
 *
 * NAG C Library
 *
 * Mark 6, 2000.
 */

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

int main(void)
{
```

```
  Complex  cn, dn, sn, z;
  Integer  exit_status = 0;
  NagError fail;
  double   ak2;

  INIT_FAIL(fail);

  /* Skip heading in data file */
  scanf("%*[^\n] ");
  printf("nag_jacobian_elliptic (s21cbc) Example Program Results\n");
  while (scanf(" (%lf,%lf) %lf%*[^\n] ", &z.re, &z.im, &ak2) != EOF)
    {
      /* nag_jacobian_elliptic (s21cbc).
       * Jacobian elliptic functions sn, cn and dn of complex
       * argument
       */
      nag_jacobian_elliptic(z, ak2, &sn, &cn, &dn, &fail);
      printf("          z                    ak2\n");
      printf(" (%8.4f,%8.4f)     %10.2f\n\n", z.re, z.im, ak2);
      if (fail.code == NE_NOERROR)
        {
          printf("        sn                     cn"
                 "                    dn\n");
          printf(" (%8.4f,%8.4f)    ", sn.re, sn.im);
          printf("(%8.4f,%8.4f)    ", cn.re, cn.im);
          printf("(%8.4f,%8.4f)", dn.re, dn.im);
          printf("\n");
        }
      else
        {
          printf("Error from nag_jacobian_elliptic (s21cbc).\n%s\n",
                 fail.message);
          exit_status = 1;
          goto END;
        }
    }
 END:
  return exit_status;
}
```

10.2 Program Data

```
nag_jacobian_elliptic (s21cbc) Example Program Data
 (-2.0, 3.0)  0.25 : Values of z and ak2
```

10.3 Program Results

```
nag_jacobian_elliptic (s21cbc) Example Program Results
          z                    ak2
 ( -2.0000,  3.0000)           0.25

        sn                     cn                      dn
 ( -1.5865,  0.2456)    (  0.3125,  1.2468)    ( -0.6395, -0.1523)
```