

NAG C Library Function Document

nag_jacobian_elliptic (s21cbc)

1 Purpose

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

2 Specification

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

3 Description

This routine evaluates the Jacobian elliptic functions $\operatorname{sn}(z|k)$, $\operatorname{cn}(z|k)$ and $\operatorname{dn}(z|k)$ given by

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

where z is a complex argument, k is a real parameter (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

$$\begin{aligned}\operatorname{sn}(z|k) &= k_1 \operatorname{sn}(kz|k_1) \\ \operatorname{cn}(z|k) &= \operatorname{dn}(kz|k_1) \\ \operatorname{dn}(z|k) &= \operatorname{cn}(kz|k_1)\end{aligned}$$

where $k_1 = 1/k$.

Special values include

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

These functions are often simply written as $\operatorname{sn}z$, $\operatorname{cn}z$ and $\operatorname{dn}z$, thereby avoiding explicit reference to the parameter 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

$$\begin{aligned}\operatorname{cd}z &= \operatorname{cn}z/\operatorname{dn}z \\ \operatorname{sd}z &= \operatorname{sn}z/\operatorname{dn}z \\ \operatorname{nd}z &= 1/\operatorname{dn}z\end{aligned}$$

$$\begin{aligned}
 dcz &= dnz/cnz \\
 ncz &= 1/cnz \\
 scz &= snz/cnz \\
 nsz &= 1/snz \\
 dsz &= dnz/snz \\
 csz &= cnz/snz
 \end{aligned}$$

(see Abramowitz and Stegun (1972)).

4 Parameters

- | | | |
|----|--|---------------------|
| 1: | z – Complex | <i>Input</i> |
| | <i>On entry:</i> the argument z of the functions. | |
| | <i>Constraints:</i> | |
| | $\text{abs}(\mathbf{z.re}) \leq \sqrt{\lambda};$ | |
| | $\text{abs}(\mathbf{z.im}) \leq \sqrt{\lambda},$ where $\lambda = 1/\mathbf{X02AMC}$. | |
| 2: | ak2 – double | <i>Input</i> |
| | <i>On entry:</i> the value of k^2 . | |
| | <i>Constraint:</i> $0.0 \leq \mathbf{ak2} \leq 1.0$. | |
| 3: | sn – Complex * | <i>Output</i> |
| 4: | cn – Complex * | <i>Output</i> |
| 5: | dn – Complex * | <i>Output</i> |
| | <i>On exit:</i> the values of the functions snz , cnz and dnz , respectively. | |
| 6: | fail – NagError * | <i>Input/Output</i> |
| | The NAG error parameter (see the Essential Introduction). | |

5 Error Indicators and Warnings

NE_REAL

On entry, **ak2** = *<value>*.
 Constraint: $0.0 \leq \mathbf{ak2} \leq 1.0$.

NE_COMPLEX

On entry, **z** = (*<value>*, *<value>*).
 Constraint: $\text{abs}(\mathbf{z.re}) \leq \lambda$ and $\text{abs}(\mathbf{z.im}) \leq \lambda$, where $\lambda = \frac{1}{\sqrt{\mathbf{X02AMC}}}$.

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.

6 Further Comments

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

$$\begin{aligned}\text{sn}z &= \frac{\text{sn}(u, k)\text{dn}(v, k')}{1 - \text{dn}^2(u, k)\text{sn}^2(v, k')} + i \frac{\text{cn}(u, k)\text{dn}(u, k)\text{sn}(v, k')\text{cn}(v, k')}{1 - \text{dn}^2(u, k)\text{sn}^2(v, k')} \\ \text{cn}z &= \frac{\text{cn}(u, k)\text{cn}(v, k')}{1 - \text{dn}^2(u, k)\text{sn}^2(v, k')} - i \frac{\text{sn}(u, k)\text{dn}(u, k)\text{sn}(v, k')\text{dn}(v, k')}{1 - \text{dn}^2(u, k)\text{sn}^2(v, k')} \\ \text{dn}z &= \frac{\text{dn}(u, k)\text{cn}(v, k')\text{dn}(v, k')}{1 - \text{dn}^2(u, k)\text{sn}^2(v, k')} - i \frac{k^2 \text{sn}(u, k)\text{cn}(u, k)\text{sn}(v, k')}{1 - \text{dn}^2(u, k)\text{sn}^2(v, k')},\end{aligned}$$

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

6.1 Accuracy

In principle the routine is capable of achieving full relative precision in the computed values. However, the accuracy obtainable in practice depends on the accuracy of the C standard library elementary functions such as $\sin$ and $\cos$.

6.2 References

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

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

7 See Also

`nag_jacobian_theta` (s21ccc)

8 Example

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

8.1 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;
    double ak2;
    Integer exit_status=0;
    NagError fail;
```

```

INIT_FAIL(fail);
Vprintf("s21cbc Example Program Results\n\n");

/* Skip heading in data file */
Vscanf("%*[^\\n] ");
while (scanf(" (%lf,%lf) %lf%*[^\\n] ", &z.re, &z.im, &ak2) != EOF)
{
    Vprintf("\n");
    s21cbc (z, ak2, &sn, &cn, &dn, &fail);
    Vprintf("          z          ak2\n");
    Vprintf(" (%8.4f,%8.4f)      %10.2f\n\n", z.re, z.im, ak2);
    if (fail.code == NE_NOERROR)
    {
        Vprintf("          sn          cn          dn\n");
        Vprintf(" (%8.4f,%8.4f)      ", sn.re, sn.im);
        Vprintf(" (%8.4f,%8.4f)      ", cn.re, cn.im);
        Vprintf(" (%8.4f,%8.4f)", dn.re, dn.im);
        Vprintf("\n");
    }
    else
    {
        Vprintf("Error from s21cbc.\n%s\n", fail.message);
        exit_status = 1;
        goto END;
    }
}
END:
return exit_status;
}

```

8.2 Program Data

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

8.3 Program Results

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)
