

# NAG Library Routine Document

## S21CBF

**Note:** before using this routine, please read the Users' Note for your implementation to check the interpretation of *bold italicised* terms and other implementation-dependent details.

### 1 Purpose

S21CBF evaluates the Jacobian elliptic functions  $\operatorname{sn} z$ ,  $\operatorname{cn} z$  and  $\operatorname{dn} z$  for a complex argument  $z$ .

### 2 Specification

```
SUBROUTINE S21CBF (Z, AK2, SN, CN, DN, IFAIL)
```

```
INTEGER IFAIL
REAL (KIND=nag_wp) AK2
COMPLEX (KIND=nag_wp) Z, SN, CN, DN
```

### 3 Description

S21CBF 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  $\phi$  (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 S21CCF).

Another nine elliptic functions may be computed via the formulae

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

(see Abramowitz and Stegun (1972)).

The values of  $\operatorname{sn} z$ ,  $\operatorname{cn} z$  and  $\operatorname{dn} z$  are obtained by calls to S21CAF. Further details can be found in Section 8.

## 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 Parameters

1: Z – COMPLEX (KIND=nag\_wp) *Input*

*On entry:* the argument  $z$  of the functions.

*Constraints:*

$$\begin{aligned}
 \operatorname{abs}(Z.\mathbf{re}) &\leq \sqrt{\lambda}; \\
 \operatorname{abs}(Z.\mathbf{im}) &\leq \sqrt{\lambda}, \text{ where } \lambda = 1/X02AMF.
 \end{aligned}$$

2: AK2 – REAL (KIND=nag\_wp) *Input*

*On entry:* the value of  $k^2$ .

*Constraint:*  $0.0 \leq \text{AK2} \leq 1.0$ .

3: SN – COMPLEX (KIND=nag\_wp) *Output*

4: CN – COMPLEX (KIND=nag\_wp) *Output*

5: DN – COMPLEX (KIND=nag\_wp) *Output*

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

6: IFAIL – INTEGER *Input/Output*

*On entry:* IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this parameter, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

*On exit:* IFAIL = 0 unless the routine detects an error or a warning has been flagged (see Section 6).

## 6 Error Indicators and Warnings

If on entry `IFAIL = 0` or `-1`, explanatory error messages are output on the current error message unit (as defined by `X04AAF`).

Errors or warnings detected by the routine:

`IFAIL = 1`

On entry, `AK2 < 0.0`,  
 or `AK2 > 1.0`,  
 or `abs(Re(Z)) > sqrt(lambda)`,  
 or `abs(Im(Z)) > sqrt(lambda)`, where  $\lambda = 1/\text{X02AMF}$ .

## 7 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 standard elementary functions such as `SIN` and `COS`.

## 8 Further Comments

The values of `sn z`, `cn z` and `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*).

## 9 Example

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

### 9.1 Program Text

```

Program s21cbfe
!      S21CBF Example Program Text
!
!      Mark 24 Release. NAG Copyright 2012.
!
!      .. Use Statements ..
Use nag_library, Only: nag_wp, s21cbf
!      .. Implicit None Statement ..
Implicit None
!      .. Parameters ..
Integer, Parameter          :: nin = 5, nout = 6
!      .. Local Scalars ..
Complex (Kind=nag_wp)       :: cn, dn, sn, z
Real (Kind=nag_wp)         :: ak2
Integer                     :: ifail
!      .. Executable Statements ..
Write (nout,*) 'S21CBF Example Program Results'
```

```

!      Skip heading in data file
      Read (nin,*)

      Write (nout,*)

      Read (nin,*) z, ak2

      ifail = 0
      Call s21cbf(z,ak2,sn,cn,dn,ifail)

      Write (nout,*) '          Z          AK2'
      Write (nout,99999) z, ak2
      Write (nout,*)
      Write (nout,*) '          SN          CN', &
      '          DN'
      Write (nout,99998) sn, cn, dn

99999 Format (1X,'(,F8.4,',',F8.4,')',5X,F10.2)
99998 Format (3(' (,F8.4,',',F8.4,')',3X))
      End Program s21cbfe

```

## 9.2 Program Data

S21CBF Example Program Data  
 (-2.0, 3.0) 0.25 : Values of Z and AK2

## 9.3 Program Results

S21CBF Example Program Results

Z	AK2	SN	CN	DN
( -2.0000, 3.0000)	0.25	( -1.5865, 0.2456)	( 0.3125, 1.2468)	( -0.6395, -0.1523)

---