

# NAG Library Routine Document

## S21BGF

**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

S21BGF returns a value of the classical (Legendre) form of the incomplete elliptic integral of the third kind, via the function name.

### 2 Specification

```
FUNCTION S21BGF (DN, PHI, DM, IFAIL)
REAL (KIND=nag_wp) S21BGF
INTEGER          IFAIL
REAL (KIND=nag_wp) DN, PHI, DM
```

### 3 Description

S21BGF calculates an approximation to the integral

$$\Pi(n; \phi | m) = \int_0^\phi (1 - n \sin^2 \theta)^{-1} (1 - m \sin^2 \theta)^{-\frac{1}{2}} d\theta,$$

where  $0 \leq \phi \leq \frac{\pi}{2}$ ,  $m \sin^2 \phi \leq 1$ ,  $m$  and  $\sin \phi$  may not both equal one, and  $n \sin^2 \phi \neq 1$ .

The integral is computed using the symmetrised elliptic integrals of Carlson (Carlson (1979) and Carlson (1988)). The relevant identity is

$$\Pi(n; \phi | m) = \sin \phi R_F(q, r, 1) + \frac{1}{3} n \sin^3 \phi R_J(q, r, 1, s),$$

where  $q = \cos^2 \phi$ ,  $r = 1 - m \sin^2 \phi$ ,  $s = 1 - n \sin^2 \phi$ ,  $R_F$  is the Carlson symmetrised incomplete elliptic integral of the first kind (see S21BBF) and  $R_J$  is the Carlson symmetrised incomplete elliptic integral of the third kind (see S21BDF).

### 4 References

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

Carlson B C (1979) Computing elliptic integrals by duplication *Numerische Mathematik* **33** 1–16

Carlson B C (1988) A table of elliptic integrals of the third kind *Math. Comput.* **51** 267–280

### 5 Arguments

1:	DN – REAL (KIND=nag_wp)	<i>Input</i>
2:	PHI – REAL (KIND=nag_wp)	<i>Input</i>
3:	DM – REAL (KIND=nag_wp)	<i>Input</i>

*On entry:* the arguments  $n$ ,  $\phi$  and  $m$  of the function.

*Constraints:*

$0.0 \leq \text{PHI} \leq \frac{\pi}{2}$ ;  
 $\text{DM} \times \sin^2(\text{PHI}) \leq 1.0$ ;  
 Only one of  $\sin(\text{PHI})$  and  $\text{DM}$  may be 1.0;

$$DN \times \sin^2(\text{PHI}) \neq 1.0.$$

Note that  $DM \times \sin^2(\text{PHI}) = 1.0$  is allowable, as long as  $DM \neq 1.0$ .

#### 4: IFAIL – INTEGER

*Input/Output*

*On entry:* IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation 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 argument, 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, PHI =  $\langle value \rangle$ .  
Constraint:  $0 \leq \text{PHI} \leq (\pi/2)$ .

IFAIL = 2

On entry, PHI =  $\langle value \rangle$  and DM =  $\langle value \rangle$ ; the integral is undefined.  
Constraint:  $DM \times \sin^2(\text{PHI}) \leq 1.0$ .

IFAIL = 3

On entry,  $\sin(\text{PHI}) = 1$  and DM = 1.0; the integral is infinite.

IFAIL = 4

On entry, PHI =  $\langle value \rangle$  and DN =  $\langle value \rangle$ ; the integral is infinite.  
Constraint:  $DN \times \sin^2(\text{PHI}) \neq 1.0$ .

IFAIL = -99

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL = -399

Your licence key may have expired or may not have been installed correctly.

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL = -999

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

## 7 Accuracy

In principle S21BGF is capable of producing full *machine precision*. However round-off errors in internal arithmetic will result in slight loss of accuracy. This loss should never be excessive as the algorithm does not involve any significant amplification of round-off error. It is reasonable to assume that the result is accurate to within a small multiple of the *machine precision*.

## 8 Parallelism and Performance

S21BGF is not threaded in any implementation.

## 9 Further Comments

You should consult the S Chapter Introduction, which shows the relationship between this routine and the Carlson definitions of the elliptic integrals. In particular, the relationship between the argument-constraints for both forms becomes clear.

For more information on the algorithms used to compute  $R_F$  and  $R_J$ , see the routine documents for S21BBF and S21BDF, respectively.

If you wish to input a value of PHI outside the range allowed by this routine you should refer to Section 17.4 of Abramowitz and Stegun (1972) for useful identities.

## 10 Example

This example simply generates a small set of nonextreme arguments that are used with the routine to produce the table of results.

### 10.1 Program Text

```

Program s21bgfe

!      S21BGF Example Program Text

!      Mark 26 Release. NAG Copyright 2016.

!      .. Use Statements ..
Use nag_library, Only: nag_wp, s21bgf, x01aaf
!      .. Implicit None Statement ..
Implicit None
!      .. Parameters ..
Integer, Parameter          :: nout = 6
!      .. Local Scalars ..
Real (Kind=nag_wp)         :: dm, dn, f, phi, pi
Integer                     :: ifail, ix
!      .. Intrinsic Procedures ..
Intrinsic                   :: real
!      .. Executable Statements ..
Write (nout,*) 'S21BGF Example Program Results'

Write (nout,*)
Write (nout,*) '      DN      PHI      DM      S21BGF'
Write (nout,*)

pi = x01aaf(pi)

data: Do ix = 1, 3
      phi = real(ix,kind=nag_wp)*pi/6.0E0_nag_wp
      dm = real(ix,kind=nag_wp)*0.25E0_nag_wp
      dn = ((-1.0E0_nag_wp)**(ix+1))*real(ix,kind=nag_wp)*0.1E0_nag_wp

      ifail = -1
      f = s21bgf(dn,phi,dm,ifail)

      If (ifail<0) Then

```

```
      Exit data
    End If

    Write (nout,99999) dn, phi, dm, f
  End Do data

99999 Format (1X,3F7.2,F12.4)
End Program s21bgfe
```

## 10.2 Program Data

None.

## 10.3 Program Results

S21BGF Example Program Results

DN	PHI	DM	S21BGF
0.10	0.52	0.25	0.5341
-0.20	1.05	0.50	1.0778
0.30	1.57	0.75	2.6568

---