Module 3.7: nag_ell_fun
Elliptic Functions

`nag_ell_fun` contains a procedure for evaluating the Jacobian elliptic functions sn, cn and dn.

Contents

Introduction

This module contains a procedure for evaluating the Jacobian elliptic functions sn, cn and dn using the process of the arithmetic–geometric mean.

`nag_ell_jac` evaluates the Jacobian elliptic functions of argument u and parameter m,

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

where ϕ, called the *amplitude* of u, is defined by the integral

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

These elliptic functions are sometimes written simply as $\mathrm{sn}\,u$, $\mathrm{cn}\,u$ and $\mathrm{dn}\,u$, avoiding explicit reference to the parameter m.

Another nine elliptic functions may be computed via the formulae

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

For further details, see Abramowitz and Stegun [1], Chapter 16.

Procedure: nag_ell_jac

1 Description

nag_ell_jac evaluates an approximation to the Jacobian elliptic functions sn, cn and dn of argument u and parameter m. The procedure is based on a procedure given by Bulirsch [2] and uses the process of the arithmetic-geometric mean (Abramowitz and Stegun [1], Chapter 16).

2 Usage

```
USE nag_ell_fun
CALL nag_ell_jac(u, m, sn, cn, dn  [, optional arguments])
```

3 Arguments

3.1 Mandatory Arguments

u — real(kind=*wp*), intent(in)

m — real(kind=*wp*), intent(in)

Input: the argument u and the parameter m of the function, respectively.

Constraints: u and m must satisfy

ABS(u) $\leq 1/\sqrt{\lambda}$,

ABS(m) $\leq 1/\sqrt{\lambda}$ if ABS(u) $< \sqrt{\lambda}$,

where λ is the *safe range* parameter defined in Section 6.1.

sn — real(kind=*wp*), intent(out)

cn — real(kind=*wp*), intent(out)

dn — real(kind=*wp*), intent(out)

Output: the values of the functions sn u, cn u and dn u respectively.

3.2 Optional Argument

error — type(nag_error), intent(inout), optional

The NAG *fl*90 error-handling argument. See the Essential Introduction, or the module document nag_error_handling (1.2). You are recommended to omit this argument if you are unsure how to use it. If this argument is supplied, it *must* be initialized by a call to nag_set_error before this procedure is called.

4 Error Codes

Fatal errors (error%level = 3):

error%code	Description
301	An input argument has an invalid value.

5 Examples of Usage

A complete example of the use of this procedure appears in Example 1 of this module document.

6 Further Comments

6.1 Algorithmic Detail

The *safe range* parameter λ is implementation dependent; in most implementations it is the maximum of `TINY(1.0_`*wp*`)` and 1.0/`HUGE(1.0_`*wp*`)`, but may be larger in some implementations to avoid problems with the elementary functions.

6.2 Accuracy

In principle the procedure is capable of achieving full relative precision in the computed values. However, the accuracy obtainable in practice depends on the accuracy of the Fortran intrinsic functions for elementary functions such as `SIN` and `COS`.

Example 1: Evaluation of sn, cn and dn

The following program evaluates the functions sn, cn and dn, for given values of the argument u and parameter m and prints the results.

1 Program Text

Note. The listing of the example program presented below is double precision. Single precision users are referred to Section 5.2 of the Essential Introduction for further information.

```
PROGRAM nag_ell_fun_ex01

  ! Example Program Text for nag_ell_fun
  ! NAG fl90, Release 3. NAG Copyright 1997.

  ! .. Use Statements ..
  USE nag_examples_io, ONLY : nag_std_out
  USE nag_ell_fun, ONLY : nag_ell_jac
  ! .. Implicit None Statement ..
  IMPLICIT NONE
  ! .. Intrinsic Functions ..
  INTRINSIC KIND
  ! .. Parameters ..
  INTEGER, PARAMETER :: n = 4
  INTEGER, PARAMETER :: wp = KIND(1.0D0)
  ! .. Local Scalars ..
  INTEGER :: i
  REAL (wp) :: cn, dn, sn
  ! .. Local Arrays ..
  REAL (wp) :: m(n), u(n)
  ! .. Executable Statements ..

  WRITE (nag_std_out,*) 'Example Program Results for nag_ell_fun_ex01'

  WRITE (nag_std_out,*)
  WRITE (nag_std_out,*) &
   '         u            m            sn            cn            dn'

  u = (/ 0.2_wp, 5.0_wp, -0.5_wp, 10.0_wp/)
  m = (/ 0.3_wp, -1.0_wp, -0.1_wp, 11.0_wp/)

  DO i = 1, n

    CALL nag_ell_jac(u(i),m(i),sn,cn,dn)

    WRITE (nag_std_out,fmt='(3X,1P,5E13.4)') u(i), m(i), sn, cn, dn
  END DO

END PROGRAM nag_ell_fun_ex01
```

2 Program Data

None.

3 Program Results

```
Example Program Results for nag_ell_fun_ex01

        u            m            sn           cn           dn
    2.0000E-01   3.0000E-01   1.9828E-01   9.8015E-01   9.9409E-01
    5.0000E+00  -1.0000E+00  -2.4403E-01   9.6977E-01   1.0293E+00
   -5.0000E-01  -1.0000E-01  -4.8117E-01   8.7663E-01   1.0115E+00
    1.0000E+01   1.1000E+01   2.5125E-01   9.6792E-01   5.5283E-01
```

References

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

[2] Bulirsch R (1965) Numerical calculation of elliptic integrals and elliptic functions *Numer. Math.* **7** 76–90