NAG Library Routine Document

C02AJF

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

C02AJF determines the roots of a quadratic equation with real coefficients.

2 Specification

```
SUBROUTINE C02AJF (A, B, C, ZSM, ZLG, IFAIL)
INTEGER            IFAIL
REAL (KIND=nag_wp) A, B, C, ZSM(2), ZLG(2)
```

3 Description

C02AJF attempts to find the roots of the quadratic equation $az^2 + bz + c = 0$ (where a, b and c are real coefficients), by carefully evaluating the 'standard' closed formula

$$z = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

It is based on the routine QDRTC from Smith (1967).

Note: it is not necessary to scale the coefficients prior to calling the routine.

4 References

Smith B T (1967) ZERPOL: a zero finding algorithm for polynomials using Laguerre's method *Technical Report* Department of Computer Science, University of Toronto, Canada

5 Arguments

1: A – REAL (KIND=nag_wp) *Input*

On entry: must contain a, the coefficient of z^2.

2: B – REAL (KIND=nag_wp) *Input*

On entry: must contain b, the coefficient of z.

3: C – REAL (KIND=nag_wp) *Input*

On entry: must contain c, the constant coefficient.

4: ZSM(2) – REAL (KIND=nag_wp) array *Output*

On exit: the real and imaginary parts of the smallest root in magnitude are stored in ZSM(1) and ZSM(2) respectively.

5: ZLG(2) – REAL (KIND=nag_wp) array *Output*

On exit: the real and imaginary parts of the largest root in magnitude are stored in ZLG(1) and ZLG(2) respectively.

6: 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, A $= 0.0$. In this case, ZSM(1) contains the root $-c/b$ and ZSM(2) contains zero.

IFAIL $= 2$

On entry, A $= 0.0$ and B $= 0.0$. In this case, ZSM(1) contains the largest machine representable number (see X02ALF) and ZSM(2) contains zero.

IFAIL $= 3$

On entry, A $= 0.0$ and the root $-c/b$ overflows. In this case, ZSM(1) contains the largest machine representable number (see X02ALF) and ZSM(2) contains zero.

IFAIL $= 4$

On entry, C $= 0.0$ and the root $-b/a$ overflows. In this case, both ZSM(1) and ZSM(2) contain zero.

IFAIL $= 5$

On entry, b is so large that b^2 is indistinguishable from $b^2 - 4ac$ and the root $-b/a$ overflows. In this case, ZSM(1) contains the root $-c/b$ and ZSM(2) contains zero.

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.

If IFAIL > 0 on exit, then ZLG(1) contains the largest machine representable number (see X02ALF) and ZLG(2) contains zero.

7 Accuracy

If IFAIL $= 0$ on exit, then the computed roots should be accurate to within a small multiple of the ***machine precision*** except when underflow (or overflow) occurs, in which case the true roots are within a small multiple of the underflow (or overflow) threshold of the machine.

8 Parallelism and Performance

C02AJF is not threaded in any implementation.

9 Further Comments

None.

10 Example

This example finds the roots of the quadratic equation $z^2 + 3z - 10 = 0$.

10.1 Program Text

```
    Program c02ajfe

!     C02AJF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: c02ajf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: a, b, c
      Integer                          :: ifail
!     .. Local Arrays ..
      Real (Kind=nag_wp)               :: zlg(2), zsm(2)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: abs
!     .. Executable Statements ..
      Write (nout,*) 'C02AJF Example Program Results'

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

      Read (nin,*) a, b, c

      ifail = -1
      Call c02ajf(a,b,c,zsm,zlg,ifail)

      If (ifail==0) Then
        Write (nout,*)
        Write (nout,*) 'Roots of quadratic equation'
        Write (nout,*)

        If (zsm(2)==0.0E0_nag_wp) Then

!         2 real roots.

          Write (nout,99999) 'z = ', zsm(1)
          Write (nout,99999) 'z = ', zlg(1)
        Else

!         2 complex roots.

          Write (nout,99998) 'z = ', zsm(1), ' +/- ', abs(zsm(2)), '*i'
```

```
        End If

      End If

99999 Format (1X,A,1P,E12.4)
99998 Format (1X,A,1P,E12.4,A,E12.4,A)
    End Program c02ajfe
```

10.2 Program Data

```
C02AJF Example Program Data
 1.0    3.0   -10.0            :A  B  C
```

10.3 Program Results

```
 C02AJF Example Program Results

 Roots of quadratic equation

 z =   2.0000E+00
 z =  -5.0000E+00
```