NAG Fortran Library Routine Document

E02RAF

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

E02RAF calculates the coefficients in a Padé approximant to a function from its user-supplied Maclaurin expansion.

2 Specification

```
SUBROUTINE E02RAF(IA, IB, C, IC, A, B, W, JW, IFAIL)
INTEGER          IA, IB, IC, JW, IFAIL
real             C(IC), A(IA), B(IB), W(JW)
```

3 Description

Given a power series

$$c_0 + c_1x + c_2x^2 + \cdots + c_{l+m}x^{l+m} + \cdots$$

this routine uses the coefficients c_i, for $i = 0, 1, \ldots, l + m$, to form the $[l/m]$ Padé approximant of the form

$$\frac{a_0 + a_1x + a_2x^2 + \cdots + a_lx^l}{b_0 + b_1x + b_2x^2 + \cdots + b_mx^m}$$

with b_0 defined to be unity. The two sets of coefficients a_j, for $j = 0, 1, \ldots, l$ and b_k, for $k = 0, 1, \ldots, m$ in the numerator and denominator are calculated by direct solution of the Padé equations (see Graves-Morris (1979)); these values are returned through the argument list unless the approximant is degenerate.

Padé approximation is a useful technique when values of a function are to be obtained from its Maclaurin expansion but convergence of the series is unacceptably slow or even non-existent. It is based on the hypothesis of the existence of a sequence of convergent rational approximations, as described in Baker and Graves-Morris (1981) and Graves-Morris (1979).

Unless there are reasons to the contrary (as discussed in Baker and Graves-Morris (1981) Chapter 4, Section 2, Chapters 5 and 6), one normally uses the diagonal sequence of Padé approximants, namely

$$\{[m/m], m = 0, 1, 2, \ldots\}.$$

Subsequent evaluation of the approximant at a given value of x may be carried out using E02RBF.

4 References

Baker G A Jr and Graves-Morris P R (1981) Padé approximants, Part 1: Basic theory *Encyclopaedia of Mathematics and its Applications* Addison-Wesley

Graves-Morris P R (1979) The numerical calculation of Padé approximants *Padé Approximation and its Applications. Lecture Notes in Mathematics* (ed L Wuytack) **765** 231–245 Addison-Wesley

5 Parameters

1: IA – INTEGER *Input*

2: IB – INTEGER *Input*

On entry: IA must specify $l + 1$ and IB must specify $m + 1$, where l and m are the degrees of the numerator and denominator of the approximant, respectively.

Constraint: IA and IB ≥ 1.

3: C(IC) – ***real*** array *Input*

On entry: $\mathrm{C}(i)$ must specify, for $i = 1, 2, \ldots, l + m + 1$, the coefficient of x^{i-1} in the given power series.

4: IC – INTEGER *Input*

On entry: the dimension of the array C as declared in the (sub)program from which E02RAF is called.

Constraint: $\mathrm{IC} \geq \mathrm{IA} + \mathrm{IB} - 1$.

5: A(IA) – ***real*** array *Output*

On exit: $\mathrm{A}(j + 1)$, for $j = 1, 2, \ldots, l + 1$, contains the coefficient a_j in the numerator of the approximant.

6: B(IB) – ***real*** array *Output*

On exit: $\mathrm{B}(k + 1)$, for $k = 1, 2, \ldots, m + 1$, contains the coefficient b_k in the denominator of the approximant.

7: W(JW) – ***real*** array *Workspace*

8: JW – INTEGER *Input*

On entry: the dimension of the array W as declared in the (sub)program from which E02RAF is called.

Constraint: $\mathrm{JW} \geq \mathrm{IB} \times (2 \times \mathrm{IB} + 3)$.

9: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.

On exit: $\mathrm{IFAIL} = 0$ unless the routine detects an error (see Section 6).

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, for users 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.**

6 Error Indicators and Warnings

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

Errors or warnings detected by the routine:

$\mathrm{IFAIL} = 1$

On entry, $\mathrm{JW} < \mathrm{IB} \times (2 \times \mathrm{IB} + 3)$,
or $\mathrm{IA} < 1$,
or $\mathrm{IB} < 1$,
or $\mathrm{IC} < \mathrm{IA} + \mathrm{IB} - 1$

(so there are insufficient coefficients in the given power series to calculate the desired approximant).

IFAIL = 2

The Padé approximant is degenerate.

7 Accuracy

The solution should be the best possible to the extent to which the solution is determined by the input coefficients. It is recommended that the user determines the locations of the zeros of the numerator and denominator polynomials, both to examine compatibility with the analytic structure of the given function and to detect defects. (Defects are nearby pole-zero pairs; defects close to $x = 0.0$ characterise ill-conditioning in the construction of the approximant.) Defects occur in regions where the approximation is necessarily inaccurate. The example program calls C02AGF to determine the above zeros.

It is easy to test the stability of the computed numerator and denominator coefficients by making small perturbations of the original Maclaurin series coefficients (e.g., c_l or c_{l+m}). These questions of intrinsic error of the approximants and computational error in their calculation are discussed in Chapter 2 of Baker and Graves-Morris (1981).

8 Further Comments

The time taken by the routine is approximately proportional to m^3.

9 Example

The example program calculates the [4/4] Padé approximant of e^x (whose power-series coefficients are first stored in the array CC). The poles and zeros are then calculated to check the character of the [4/4] Padé approximant.

9.1 Program Text

Note: the listing of the example program presented below uses ***bold italicised*** terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     E02RAF Example Program Text.
*     Mark 16 Revised. NAG Copyright 1993.
*     .. Parameters ..
      INTEGER          L, M, IA, IB, IC, IW
      PARAMETER        (L=4,M=4,IA=L+1,IB=M+1,IC=IA+IB-1,IW=IB*(2*IB+3))
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
      LOGICAL          SCALE
      PARAMETER        (SCALE=.TRUE.)
*     .. Local Scalars ..
      INTEGER          I, IFAIL
*     .. Local Arrays ..
      real             AA(IA), BB(IB), CC(IC), DD(IA+IB), W(IW),
     +                 WORK(2*(L+M+1)), Z(2,L+M)
*     .. External Subroutines ..
      EXTERNAL         C02AGF, E02RAF
*     .. Intrinsic Functions ..
      INTRINSIC        real
*     .. Executable Statements ..
      WRITE (NOUT,*) 'E02RAF Example Program Results'
*     Power series coefficients in CC
      CC(1) = 1.0e0
      DO 20 I = 1, IC - 1
         CC(I+1) = CC(I)/real(I)
   20 CONTINUE
      WRITE (NOUT,*)
      WRITE (NOUT,*) 'The given series coefficients are'
      WRITE (NOUT,99999) (CC(I),I=1,IC)
      IFAIL = 0
```

```
*
      CALL E02RAF(IA,IB,CC,IC,AA,BB,W,IW,IFAIL)
*
      WRITE (NOUT,*)
      WRITE (NOUT,*) 'Numerator coefficients'
      WRITE (NOUT,99999) (AA(I),I=1,IA)
      WRITE (NOUT,*)
      WRITE (NOUT,*) 'Denominator coefficients'
      WRITE (NOUT,99999) (BB(I),I=1,IB)
*     Calculate zeros of the approximant using C02AGF
*     First need to reverse order of coefficients
      DO 40 I = 1, IA
         DD(IA-I+1) = AA(I)
   40 CONTINUE
      IFAIL = 0
*
      CALL C02AGF(DD,L,SCALE,Z,WORK,IFAIL)
*
      WRITE (NOUT,*)
      WRITE (NOUT,*) 'Zeros of approximant are at'
      WRITE (NOUT,*)
      WRITE (NOUT,*) '    Real part    Imag part'
      WRITE (NOUT,99998) (Z(1,I),Z(2,I),I=1,L)
*     Calculate poles of the approximant using C02AGF
*     Reverse order of coefficients
      DO 60 I = 1, IB
         DD(IB-I+1) = BB(I)
   60 CONTINUE
      IFAIL = 0
*
      CALL C02AGF(DD,M,SCALE,Z,WORK,IFAIL)
*
      WRITE (NOUT,*)
      WRITE (NOUT,*) 'Poles of approximant are at'
      WRITE (NOUT,*)
      WRITE (NOUT,*) '    Real part    Imag part'
      WRITE (NOUT,99998) (Z(1,I),Z(2,I),I=1,M)
      STOP
*
99999 FORMAT (1X,5e13.4)
99998 FORMAT (1X,2e13.4)
      END
```

9.2 Program Data

None.

9.3 Program Results

```
 E02RAF Example Program Results

 The given series coefficients are
    0.1000E+01   0.1000E+01   0.5000E+00   0.1667E+00   0.4167E-01
    0.8333E-02   0.1389E-02   0.1984E-03   0.2480E-04

 Numerator coefficients
    0.1000E+01   0.5000E+00   0.1071E+00   0.1190E-01   0.5952E-03

 Denominator coefficients
    0.1000E+01  -0.5000E+00   0.1071E+00  -0.1190E-01   0.5952E-03

 Zeros of approximant are at

     Real part    Imag part
   -0.5792E+01   0.1734E+01
   -0.5792E+01  -0.1734E+01
   -0.4208E+01   0.5315E+01
   -0.4208E+01  -0.5315E+01

 Poles of approximant are at
```

```
 Real part    Imag part
0.5792E+01   0.1734E+01
0.5792E+01  -0.1734E+01
0.4208E+01   0.5315E+01
0.4208E+01  -0.5315E+01
```