1 SUMMARY

To find **all the roots of a polynomial** with **complex coefficients,** that is calculate the roots of

$$a_0 + a_1 x + a_2 x^2 + ... + a_n x^n = 0.$$

The user can supply error bounds on the coefficients of the polynomial and the routine returns bounds on the moduli of the errors in the roots.

The roots are found by the method of Madsen (BIT **13**, 71-75, 1973) and error bounds by the application of Rouché's theorem as recommended by Wilkinson (J. Inst. Maths. Applics. **8**, 16-35, 1971), see also K.Madsen and J.K.Reid, Harwell report R.7986 (1975).

ATTRIBUTES — Version: 1.1.0. (22 February 2005) **Types:** Real (single, double). **Calls:** `FD15`. **Original date:** September 1990. **Origin:** K.Madsen, Copenhagen, and J.K.Reid, Rutherford Appleton Laboratory.

2 HOW TO USE THE PACKAGE

Although there are both single and double precision versions of the routine available, the user is **strongly advised** to use the double precision version unless single precision on his or her machine actually means 8-byte arithmetic.

2.1 Argument list

The single precision version

```
CALL PA16A(A,N,ROOT,E,W,F,IG)
```

The double precision version

```
CALL PA16AD(A,N,ROOT,E,W,F,IG)
```

`A` is a `COMPLEX` (`COMPLEX*16` in the D version) array of dimension `(0:N)` that must be set by the user to the coefficients of the polynomial, that is, set `A(j)` = a_j, j = `0,1,..,N`. The array is altered by the routine, but is reset to its original value before return.

`N` is an `INTEGER` variable which must be set by the user to n, the degree of the polynomial. It is not altered. **Restriction:** `N` ≥ `1`.

`ROOT` is a `COMPLEX` (`COMPLEX*16` in the D version) array of size `N` that need not be set by the user and is used to return the roots. For each infinite root corresponding to a zero leading coefficient, the dummy value `HUGE` is returned, where `HUGE` = `FD15AD('H')` is a number near the overflow limit.

`E` is a `REAL` (`DOUBLE PRECISION` in the D version) array of dimension `(0:N)` that the user must set to error bounds for the coefficients, or to zero if these are accurate to machine precision. On return, `E(i)` will have been set to an approximate bound on the modulus of the error in `ROOT(i),i=1,2,..,N`.

`W` is a `COMPLEX` (`COMPLEX*16` in the D version) work array of dimension `(0:N)`.

`F` is a `REAL` (`DOUBLE PRECISION` in the D version) work array of dimension `(-N:N)`.

`IG` is an `INTEGER` work array of length `N`.

2.2 Alternative entry

The error analysis part of a call to `PA16A/AD` takes typically about 20% of its time. If speed is important and error bounds are not wanted then a call of the form

The single precision version

```
CALL PA16B(A,N,ROOT,W)
```

The double precision version

```
CALL PA16BD(A,N,ROOT,W)
```

should be made. The arguments are the same as those for the main entry described in §2.1 except that the arguments `E`, `F`, and `IG` are omitted.

3 GENERAL INFORMATION

Workspace: Provided by the user in arguments `W`, `F`, and `IG`.

Use of common: None.

Other routines called directly: `FD15A/AD`, `PA16C/CD`, and `PA16D/DD`.

Input/output: None.

Restrictions: `N` ≥ 1.

4 METHOD

The roots are found by the algorithm of Madsen, BIT (1973) **13**, 71-75, the principal features of which are Newton iteration followed by deflation. The error bounds are found by the application of Rouché's theorem as recommended by Wilkinson, J. Inst. Maths. Applics. (1971) **8**, 16-35, except that discs are always taken with centres on the approximate roots and errors in multiplying out the polynomial $\prod_{i=1}^{n}(z-r_i)$ are ignored. The disc for each root is such that it contains exactly the same number of approximate roots r_i as exact roots of the true polynomial. Note that in the case of true multiple roots the corresponding approximate roots may be quite well separated, but each will lie in the disc of all the others and their mean will be a good estimate of the true multiple root. Further documentation can be found in K.Madsen and J.K.Reid, Harwell report R.7986 (1975).

This is a revised version `PA06`. The revisions consist of the use of `COMPLEX*16`, simplification of the way workspace is organized, use of library calls to set machine parameters, some renaming of variables for greater clarity, and additional comments. The algorithm is unchanged.

5 EXAMPLE OF USE

The following program reads a polynomial of degree up to 10 and finds its roots:

```
      COMPLEX*16 A(0:10),ROOT(10),W(0:10)
      DOUBLE PRECISION E(0:10),F(-10:10)
      INTEGER IG(10),N,I
C
      EXTERNAL PA16AD
C
C Read polynomial and set array E.
      READ(*,*)N,(A(I),I=N,0,-1)
      WRITE(*,10)(A(I),I=N,0,-1)
10    FORMAT(/' Polynomial:'/2(:'  (', E12.5, ',', E12.5, ')' ) )
      DO 20 I=0,N
         E(I)=0
20    CONTINUE
C
C Call PA16 and write calculated roots and error bounds
      CALL PA16AD(A,N,ROOT,E,W,F,IG)
      WRITE(*,30)(ROOT(I),E(I),I=1,N)
30    FORMAT(/' Roots and error bounds:'/
     *        1P,('  (', E12.5, ',', E12.5, ')', E12.5) )
      END
```

The following data is suitable for the polynomial $x^5 - 1$:

```
5
(1.0,0.0) 4*(0.0,0.0) (-1.0,0.0)
```

and produces the following output:

```
 Polynomial:
  ( 0.10000E+01, 0.00000E+00)  ( 0.00000E+00, 0.00000E+00)
  ( 0.00000E+00, 0.00000E+00)  ( 0.00000E+00, 0.00000E+00)
  ( 0.00000E+00, 0.00000E+00)  (-0.10000E+01, 0.00000E+00)

 Roots and error bounds:
  ( 3.09017E-01, 9.51057E-01) 1.73857E-16
  ( 3.09017E-01,-9.51057E-01) 1.73857E-16
  (-8.09017E-01, 5.87785E-01) 1.73857E-16
  (-8.09017E-01,-5.87785E-01) 1.73857E-16
  ( 1.00000E+00, 0.00000E+00) 1.73857E-16
```