17.3 Double Precision Complex Computation

A. Purpose

These subprograms do the following computations for double precision complex data: sum, difference, product, quotient, square root, and absolute value. The Fortran 77 standard does not support a double precision complex data type. Using the convention of storing the real and imaginary parts of a complex number as an adjacent pair of double precision numbers, this set of subprograms provides the above operations while staying within the standard.

B. Usage

B.1 Program Prototype, Double Precision

DOUBLE PRECISION A(2)**, B**(2)**, RESULT**(2)**, ABSVAL, DZABS**

Assign values to A(), or to A() and B(), and access the appropriate subprogram. The results will be stored in RESULT() or ABSVAL.

CALL ZSUM(A, B, RESULT)	Result $= a + b$
CALL ZDIF(A, B, RESULT)	Result $= a - b$
CALL ZPRO(A, B, RESULT)	Result $= a \times b$
CALL ZQUO(A, B, RESULT)	Result $= a \ / \ b$
CALL ZSQRTX(A, RESULT)	Result $= \sqrt{a}$
ABSVAL = DZABS(A)	Absval $= \lvert a \rvert$

B.2 Argument Definitions

A() [in] Contains the double precision complex value, a, with the real part in A(1) and the imaginary part in A(2).

B() [in] Contains the double precision complex value, b, with the real part in B(1) and the imaginary part in B(2).

RESULT() [out] On return contains the double precision complex result with the real part in RESULT(1) and the imaginary part in RESULT(2). It is allowable for the array RESULT() to occupy the same storage locations as the arrays A() and/or B(). For example the statement, CALL ZPRO(Z, Z, Z), is permissible.

In the case of the complex square root, which is double valued, ZSQRTX returns the root with nonnegative real part. If the real part of the root is zero it returns the root whose imaginary part is nonnegative. The other root is always the negative of the returned root.

DZABS [out] Returns the double precision absolute value of the complex number represented by the pair, (A(1),A(2)).

C. Examples and Remarks

The program, DRZCOMP, with its output, ODZCOMP, illustrates the use of these subprograms. The example begins with three complex constants, a, b, and u. It computes $v = u + a$, $w = v \times b$, and $z = \sqrt{w}$. It then inverts this sequence of computations by computing $w_2 = z \times z$, $v_2 = w_2/b$, and $u_2 = v_2 - a$. Mathematically this should result in $w_2 = w$, $v_2 = v$, and $u_2 = u$. We test the last of these relations by computing TEST $= u_2 - u$. Additionally TEST2 is a measure of the error in an application of DZABS. Both TEST and TEST2 are seen to be acceptably small given that the computation was done with double precision IEEE arithmetic.

D. Functional Description

D.1 Method

To compute (u, v) as the square root of (x, y) the basic algorithm is

$r = \sqrt{x^2 + y^2}$

$u = \sqrt{(r + |x|)/2}$

$v = |y|/(2u)$

if $x < 0$ swap u and v

if $y < 0$ set $v = -v$

Subroutine ZSQRTX contains special treatment for the cases of $x = 0$ or $y = 0$, and uses scaling to avoid unnecessary overflow that could result from computing x^2 or y^2.

Subprograms ZQUO and DZABS use scaling to avoid unnecessary overflow. In the five subroutines the implementations permit the output array to occupy the same storage locations as the input arrays.

D.2 Accuracy tests

These subprograms have been tested using arguments on the four principal axes and interior to the eight octants. Results were consistent with the machine precision.

E. Error Procedures and Restrictions

If B(1) = 0 and B(2) = 0 in ZQUO, the subroutine will execute a division by zero. We assume the host system will produce a runtime error diagnostic in this case.

If the function DZABS is used its name must be typed as double precision in the referencing program.

F. Supporting Information

The source language is ANSI Fortran 77.

Designed by C. L. Lawson, JPL, May 1986.

Programmed by C. L. Lawson and S. Y. Chiu, JPL, May 1986, Feb. 1987, Nov. 1987.

Each of the subprograms, DZABS, ZDIF, ZPRO, ZQUO, ZSQRTX, and ZSUM is contained in a program file of the same name.

DRZCOMP

```
c      program DRZCOMP
c>> 2001-01-24 DRZCOMP  ZSQRT -> ZSQRTX to fix C lib. problems.
c>> 1998-01-22 DRZCOMP  Krogh Added ZDIF,..., ZSUM to exernal statement.
c>> 1996-05-28 DRZCOMP  Krogh Added external statement.
c>> 1993-02-04 CLL Added call to DZABS.
c>> 1987-12-09 DRZCOMP  Lawson  Initial Code.
c      Demo driver for DZABS, ZSUM, ZDIF, ZPRO, ZQUO, and ZSQRTX.
c      C. L. Lawson, JPL, 1987 Feb 17.
c     ------------------------------------------------------------------
      external DZABS, ZDIF, ZPRO, ZQUO, ZSQRTX, ZSUM
      double precision DZABS
      double precision A(2), B(2), C(2), DMAG, U(2), U2(2), V(2), V2(2)
      double precision W(2), W2(2), TEST(2), Z(2)
c     ------------------------------------------------------------------
      A(1) = 6.0D0/7.0D0
      A(2) = -14.0D0/15.0D0
      B(1) = -29.0D0/31.0D0
      B(2) = 47.0D0/43.0D0
      U(1) = 51.0D0/53.0D0
      U(2) = 73.0D0/71.0D0
      call ZSUM(U, A, V)
      call ZPRO(V, B, W)
      call ZSQRTX(W,    Z)
      call ZPRO(Z, Z, W2)
      call ZQUO(W2, B, V2)
      call ZDIF(V2, A, U2)
      call ZDIF(U2, U, TEST)
      C(1) =  3.0d0/7.0d0
      C(2) = -4.0d0/7.0d0
      DMAG = DZABS(C)
      print'(a/)','                                        ODZCOMP'
      print '(1x,a,f19.15,a,f19.15,a)',
     *    'A         = (',A(1),',',A(2),  ')',
     *    'B         = (',B(1),',',B(2),  ')',
     *    'U         = (',U(1),',',U(2),  ')',
     *    'V = U+A   = (',V(1),',',V(2),  ')',
     *    'W = V*B   = (',W(1),',',W(2),  ')',
     *    'Z=sqrt(W)= (',Z(1),',',Z(2),  ')',
     *    'W2 = Z*Z  = (',W2(1),',',W2(2),  ')',
     *    'V2= W2/B  = (',V2(1),',',V2(2),  ')',
     *    'U2= V2-A  = (',U2(1),',',U2(2),  ')'
      print '(1x,a,g19.3,a,g19.3,a)',
     *    'TEST=U2-U= (',TEST(1),',',TEST(2),  ')'
      C(1) =  3.0d0/7.0d0
      C(2) = -4.0d0/7.0d0
      DMAG = DZABS(C)
      print '(/1x,a,f19.15,a,f19.15,a)',
     *    'C         = (',C(1),',',C(2),  ')'
      print '(1x,a,g11.3)',
```

```
*     'TEST2 = DZABS(C)-(5/7) = ',DMAG-(5.0d0/7.0d0)
      stop

      end
```

ODZCOMP

ODZCOMP

```
A           = (   0.857142857142857, -0.933333333333333)
B           = (  -0.935483870967742,  1.093023255813953)
U           = (   0.962264150943396,  1.028169014084507)
V = U+A     = (   1.819407008086253,  0.094835680751174)
W = V*B     = (  -1.805683515332347,  1.899936921894192)
Z=sqrt(W)   = (   0.638526930480371,  1.487750031517738)
W2 = Z*Z    = (  -1.805683515332346,  1.899936921894192)
V2= W2/B    = (   1.819407008086253,  0.094835680751173)
U2= V2-A    = (   0.962264150943396,  1.028169014084507)
TEST=U2-U=  = (             -0.111E-15,           -0.222E-15)

C           = (   0.428571428571429, -0.571428571428571)
TEST2 = DZABS(C)-(5/7) =   -0.111E-15
```