

NAG Library Routine Document

F16JTF (BLAS_ZAMIN_VAL)

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

F16JTF (BLAS_ZAMIN_VAL) computes, with respect to absolute value, the smallest component of a complex vector, along with the index of that component.

2 Specification

```
SUBROUTINE F16JTF (N, X, INCX, K, R)
  INTEGER          N, INCX, K
  REAL (KIND=nag_wp) R
  COMPLEX (KIND=nag_wp) X(1+(N-1)*ABS(INCX))
```

The routine may be called by its BLAST name ***blas_zamin_val***.

3 Description

F16JTF (BLAS_ZAMIN_VAL) computes, with respect to absolute value, the smallest component, r , of an n -element complex vector x , and determines the smallest index, k , such that

$$r = |\operatorname{Re} x_k| + |\operatorname{Im} x_k| = \min_j |\operatorname{Re} x_j| + |\operatorname{Im} x_j|.$$

4 References

Basic Linear Algebra Subprograms Technical (BLAST) Forum (2001) *Basic Linear Algebra Subprograms Technical (BLAST) Forum Standard* University of Tennessee, Knoxville, Tennessee <http://www.netlib.org/blas/blast-forum/blas-report.pdf>

5 Arguments

- 1: N – INTEGER *Input*
On entry: n , the number of elements in x .
- 2: X(1 + (N – 1) × |INCX|) – COMPLEX (KIND=nag_wp) array *Input*
On entry: the n -element vector x .
 If INCX > 0, x_i must be stored in X(($i - 1$) × INCX + 1), for $i = 1, 2, \dots, N$.
 If INCX < 0, x_i must be stored in X((N – i) × |INCX| + 1), for $i = 1, 2, \dots, N$.
 Intermediate elements of X are not referenced. If N = 0, X is not referenced.
- 3: INCX – INTEGER *Input*
On entry: the increment in the subscripts of X between successive elements of x .
Constraint: INCX ≠ 0.
- 4: K – INTEGER *Output*
On exit: k , the index, from the set {1, 2, ..., N}, of the smallest component of x with respect to absolute value. If N ≤ 0 on input then K is returned as 0.

5: R – REAL (KIND=nag_wp)

Output

On exit: r , the smallest component of x with respect to absolute value. If $N \leq 0$ on input then R is returned as 0.0.

6 Error Indicators and Warnings

If $INCX = 0$, an error message is printed and program execution is terminated.

7 Accuracy

The BLAS standard requires accurate implementations which avoid unnecessary over/underflow (see Section 2.7 of Basic Linear Algebra Subprograms Technical (BLAST) Forum (2001)).

8 Parallelism and Performance

F16JTF (BLAS_ZAMIN_VAL) is not threaded in any implementation.

9 Further Comments

None.

10 Example

This example computes the smallest component with respect to absolute value and index of that component for the vector

$$x = (-4 + 2.1i, 3.7 + 4.5i, -6 + 1.2i)^T.$$

10.1 Program Text

```

Program f16jtfe

!      F16JTF Example Program Text

!      Mark 26 Release. NAG Copyright 2016.

!      .. Use Statements ..
!      Use nag_library, Only: blas_zamin_val, nag_wp
!      .. Implicit None Statement ..
!      Implicit None
!      .. Parameters ..
!      Integer, Parameter          :: nin = 5, nout = 6
!      .. Local Scalars ..
!      Real (Kind=nag_wp)          :: r
!      Integer                     :: i, incx, ix, k, n
!      .. Local Arrays ..
!      Complex (Kind=nag_wp), Allocatable :: x(:)
!      .. Intrinsic Procedures ..
!      Intrinsic                   :: abs
!      .. Executable Statements ..
!      Write (nout,*) 'F16JTF Example Program Results'

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

!      Read (nin,*) n, incx
!      Allocate (x(1+(n-1)*abs(incx)))

!      Read the vector x and store forwards or backwards
!      as determined by incx.
!      If (incx>0) Then
!          ix = 1
!      Else

```

```

        ix = 1 - (n-1)*incx
    End If

    Do i = 1, n
        Read (nin,*) x(ix)
        ix = ix + incx
    End Do

!      Find k = argmin(abs(Re(x))+abs(Im(x))) and
!              r = min(abs(Re(x))+abs(Im(x))).

    Call blas_zamin_val(n,x,incx,k,r)

    Write (nout,*)
    Write (nout,99999) k
    Write (nout,99998) r

99999 Format (1X,'Index of absolutely smallest component of x is',I3)
99998 Format (1X,'Absolutely smallest component of x is',F12.5)
    End Program f16jtfe

```

10.2 Program Data

F16JTF Example Program Data

3 1	: n and incx
(-4., 2.1)	
(3.7, 4.5)	
(-6., 1.2)	: Vector x

10.3 Program Results

F16JTF Example Program Results

```

Index of absolutely smallest component of x is 1
Absolutely smallest component of x is      6.10000

```
