NAG Library Routine Document

F16JRF (BLAS_DAMIN_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

F16JRF (BLAS_DAMIN_VAL) computes, with respect to absolute value, the smallest component of a real vector, along with the index of that component.

2 Specification

```
SUBROUTINE F16JRF (N, X, INCX, K, R)
INTEGER              N, INCX, K
REAL (KIND=nag_wp) X(1+(N-1)*ABS(INCX)), R
```

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

3 Description

F16JRF (BLAS_DAMIN_VAL) computes, with respect to absolute value, the smallest component, r, of an n-element real vector x, and determines the smallest index, k, such that

$$r = |x_k| = \min_j |x_j|.$$

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 Parameters

1: N – INTEGER *Input*

On entry: n, the number of elements in x.

2: X($1 + (\text{N} - 1) \times |\text{INCX}|$) – REAL (KIND=nag_wp) array *Input*

On entry: the vector x. Element x_i is stored in X($(i-1) \times |\text{INCX}| + 1$), for $i = 1, 2, \ldots, n$.

3: INCX – INTEGER *Input*

On entry: the increment in the subscripts of X between successive elements of x.

Constraint: INCX $\neq 0$.

4: K – INTEGER *Output*

On exit: k, the index, from the set $\{1, 1 + |\text{INCX}|, \ldots, 1 + (\text{N} - 1) \times |\text{INCX}|\}$, 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 ≤ 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

Not applicable.

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

$$x = (1, 10, 11, -2, 9)^{\mathrm{T}}.$$

10.1 Program Text

```
    Program f16jrfe

!     F16JRF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: blas_damin_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, k, n
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: x(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: abs
!     .. Executable Statements ..
      Write (nout,*) 'F16JRF Example Program Results'

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

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

      Read (nin,*)(x(i),i=1,1+(n-1)*abs(incx),incx)

!     Find K = ARGMIN(ABS(X)) and R = MIN(ABS(X)).

      Call blas_damin_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 value is',F12.5)
    End Program f16jrfe
```

10.2 Program Data

```
F16JRF Example Program Data
  5   1                                                  : N and INCX
  1.0   10.0   11.0   -2.0   9.0                         : Array X
```

10.3 Program Results

```
 F16JRF Example Program Results

 Index of absolutely smallest component of X is  1
 Absolutely smallest value is     1.00000
```