NAG Library Routine Document

F16ECF (BLAS_DAXPBY)

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

F16ECF (BLAS_DAXPBY) computes the sum of two scaled vectors, for real vectors and scalars.

2 Specification

```
SUBROUTINE F16ECF (N, ALPHA, X, INCX, BETA, Y, INCY)
INTEGER            N, INCX, INCY
REAL (KIND=nag_wp) ALPHA, X(1+(N-1)*ABS(INCX)), BETA,                  &
                   Y(1+(N-1)*ABS(INCY))
```

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

3 Description

F16ECF (BLAS_DAXPBY) performs the operation

$$y \leftarrow \alpha x + \beta y$$

where x and y are n-element real vectors, and α and β real scalars. If n is equal to zero, or if α is equal to zero and β is equal to 1, this routine returns immediately.

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 and y.

2: ALPHA – REAL (KIND=nag_wp) *Input*

On entry: the scalar α.

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

On entry: the n-element vector x.

If INCX > 0, x_i must be stored in X$((i-1) \times \text{INCX} + 1)$, for $i = 1, 2, \ldots, \text{N}$.

If INCX < 0, x_i must be stored in X$((\text{N} - i) \times |\text{INCX}| + 1)$, for $i = 1, 2, \ldots, \text{N}$.

Intermediate elements of X are not referenced. If N $= 0$, X is not referenced.

4: INCX – INTEGER *Input*

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

Constraint: INCX $\neq 0$.

5: BETA – REAL (KIND=nag_wp) *Input*

On entry: the scalar β.

6: Y$(1 + (\mathrm{N} - 1) \times |\mathrm{INCY}|)$ – REAL (KIND=nag_wp) array *Input/Output*

On entry: the n-element vector y.

If INCY > 0, y_i must be stored in Y$((i-1) \times \mathrm{INCY} + 1)$, for $i = 1, 2, \ldots, \mathrm{N}$.

If INCY < 0, y_i must be stored in Y$((\mathrm{N} - i) \times |\mathrm{INCY}| + 1)$, for $i = 1, 2, \ldots, \mathrm{N}$.

Intermediate elements of Y are not referenced.

On exit: the updated vector y stored in the array elements used to supply the original vector y.

Intermediate elements of Y are unchanged.

7: INCY – INTEGER *Input*

On entry: the increment in the subscripts of Y between successive elements of y.

Constraint: INCY $\neq 0$.

6 Error Indicators and Warnings

If INCX $= 0$ or INCY $= 0$, 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

F16ECF (BLAS_DAXPBY) makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

None.

10 Example

This example computes the result of a scaled vector accumulation for

$$\begin{aligned} \alpha &= 3, & x &= (-6, 4.5, 3.7, 2.1, -4)^{\mathrm{T}}, \\ \beta &= -1, & y &= (-5.1, -5, 6.4, -2.4, -3)^{\mathrm{T}}. \end{aligned}$$

x and y are stored in reverse order.

10.1 Program Text

```
    Program f16ecfe

!     F16ECF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: blas_daxpby, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: alpha, beta
      Integer                          :: i, incx, incy, ix, iy, n
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: x(:), y(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: abs
!     .. Executable Statements ..
      Write (nout,*) 'F16ECF Example Program Results'

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

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

      Read (nin,*) alpha, beta

!     Read the vectors x and y and store forwards or backwards
!     as determined by incx (resp. incy).
      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

      If (incy>0) Then
        iy = 1
      Else
        iy = 1 - (n-1)*incy
      End If

      Do i = 1, n
        Read (nin,*) y(iy)
        iy = iy + incy
      End Do

!     Compute y = alpha*x + beta*y

      Call blas_daxpby(n,alpha,x,incx,beta,y,incy)

!     Display the vector y forwards or backwards
!     as determined by incy.
      Write (nout,*)
      Write (nout,99999)
      If (incy>0) Then
        Write (nout,99998) y(1:1+(n-1)*incy:incy)
      Else
        Write (nout,99998) y(1-(n-1)*incy:1:incy)
```

```
      End If

99999 Format (1X,'Result of scaled vector addition is')
99998 Format (1X,'y =',5F9.4)
    End Program f16ecfe
```

10.2 Program Data

```
F16ECF Example Program Data
   5                                  : n
  -1    -1                            : incx and incy
   3.0  -1.0                          : alpha and beta
  -6.0
   4.5
   3.7
   2.1
  -4.0                                : Vector x
  -5.1
  -5.0
   6.4
  -2.4
  -3.0                                : Vector y
```

10.3 Program Results

```
 F16ECF Example Program Results

 Result of scaled vector addition is
 y = -12.9000  18.5000   4.7000   8.7000  -9.0000
```