NAG Toolbox

nag_blast_dwaxpby (f16eh)

1 Purpose

nag_blast_dwaxpby (f16eh) computes the sum of two scaled vectors, preserving input, for real scalars and vectors.

2 Syntax

```
[w] = nag_blast_dwaxpby(n, alpha, x, incx, beta, y, incy, incw)
[w] = f16eh(n, alpha, x, incx, beta, y, incy, incw)
```

3 Description

nag_blast_dwaxpby (f16eh) performs the operation

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

where x and y are n-element real vectors, and α and β are real scalars.

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

5.1 Compulsory Input Parameters

1: **n** – INTEGER

n, the number of elements in x, y and w.

2: **alpha** – REAL (KIND=nag_wp)

The scalar α.

3: **x**$(\mathbf{1} + (\mathbf{n} - \mathbf{1}) \times |\mathbf{incx}|)$ – REAL (KIND=nag_wp) array

The n-element vector x.

If **incx** > 0, x_i must be stored in **x**$((i-1) \times |\mathbf{incx}| + 1)$, for $i = 1, 2, \ldots,$ **n**.

If **incx** < 0, x_i must be stored in **x**$((\mathbf{n} - i) \times |\mathbf{incx}| - 1)$, for $i = 1, 2, \ldots,$ **n**.

Intermediate elements of **x** are not referenced.

4: **incx** – INTEGER

The increment in the subscripts of **x** between successive elements of x.

Constraint: **incx** $\neq 0$.

5: **beta** – REAL (KIND=nag_wp)

The scalar β.

6: **y**$(\mathbf{1}+(\mathbf{n}-\mathbf{1})\times|\mathbf{incy}|)$ – REAL (KIND=nag_wp) array

The n-element vector y.

If **incy** > 0, y_i must be stored in **y**$(1+(i-1)\times\mathbf{incy})$, for $i=1,2,\ldots,\mathbf{n}$.

If **incy** < 0, y_i must be stored in **y**$(1-(\mathbf{n}-i)\times\mathbf{incy})$, for $i=1,2,\ldots,\mathbf{n}$.

Intermediate elements of **y** are not referenced.

7: **incy** – INTEGER

The increment in the subscripts of **y** between successive elements of y.

Constraint: **incy** $\neq 0$.

8: **incw** – INTEGER

The increment in the subscripts of **w** between successive elements of w.

Constraint: **incw** $\neq 0$.

5.2 Optional Input Parameters

None.

5.3 Output Parameters

1: **w**$(\mathbf{1}+(\mathbf{n}-\mathbf{1})\times|\mathbf{incw}|)$ – REAL (KIND=nag_wp) array

The n-element vector w.

If **incw** > 0, w_i is in **w**$(1+(i-1)\times\mathbf{incw})$, for $i=1,2,\ldots,\mathbf{n}$.

If **incw** < 0, w_i is in **w**$(1+(\mathbf{n}-i)\times\mathbf{incw})$, for $i=1,2,\ldots,\mathbf{n}$.

Intermediate elements of **w** are not referenced.

6 Error Indicators and Warnings

If **incx** $= 0$ or **incy** $= 0$ or **incw** $= 0$, 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 Further Comments

None.

9 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, and also the sum vector w, are stored in reverse order.

9.1 Program Text

```
     function f16eh_example

fprintf('f16eh example results\n\n');

% real vectors x and y;
n = nag_int(5);
x = [-4    2.1    3.7    4.5   -6.0];
y = [-3   -2.4    6.4   -5.0   -5.1];

% w = 3x - y;
alpha = 3;
beta = -1;

incx = nag_int(1);
incy = incx;
incw = incx;

[w] = f16eh(n, alpha, x, incx, beta, y, incy, incw);

fprintf('x = ');
fprintf('%5.1f',x);
fprintf('\ny = ');
fprintf('%5.1f',y);
fprintf('\n%4.1f x %+4.1f y = ',alpha,beta);
fprintf('%7.1f',w);
fprintf('\n');
```

9.2 Program Results

```
     f16eh example results

x =  -4.0  2.1  3.7  4.5 -6.0
y =  -3.0 -2.4  6.4 -5.0 -5.1
 3.0 x -1.0 y =    -9.0    8.7    4.7   18.5  -12.9
```