

## NAG Toolbox

### nag\_blast\_zwaxpby (f16gh)

## 1 Purpose

nag\_blast\_zwaxpby (f16gh) computes the sum of two scaled vectors, preserving input, for complex scalars and vectors.

## 2 Syntax

```
[w] = nag_blast_zwaxpby(n, alpha, x, incx, beta, y, incy, incw)
[w] = f16gh(n, alpha, x, incx, beta, y, incy, incw)
```

## 3 Description

nag\_blast\_zwaxpby (f16gh) performs the operation

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

where  $x$  and  $y$  are  $n$ -element complex vectors, and  $\alpha$  and  $\beta$  are complex 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.orgblas/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** – COMPLEX (KIND=nag\_wp)

The scalar  $\alpha$ .

3: **x(1 + (n - 1) × |incx|)** – COMPLEX (KIND=nag\_wp) array

The  $n$ -element vector  $x$ .

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

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

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

4: **incx** – INTEGER

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

*Constraint:* **incx** ≠ 0.

5: **beta** – COMPLEX (KIND=nag\_wp)

The scalar  $\beta$ .

6:  $\mathbf{y}(1 + (\mathbf{n} - 1) \times |\mathbf{incy}|)$  – COMPLEX (KIND=nag\_wp) array

The  $n$ -element vector  $y$ .

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

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

Intermediate elements of  $\mathbf{y}$  are not referenced.

7:  $\mathbf{incy}$  – INTEGER

The increment in the subscripts of  $\mathbf{y}$  between successive elements of  $y$ .

*Constraint:*  $\mathbf{incy} \neq 0$ .

8:  $\mathbf{inew}$  – INTEGER

The increment in the subscripts of  $\mathbf{w}$  between successive elements of  $w$ .

*Constraint:*  $\mathbf{inew} \neq 0$ .

## 5.2 Optional Input Parameters

None.

## 5.3 Output Parameters

1:  $\mathbf{w}(1 + (\mathbf{n} - 1) \times |\mathbf{inew}|)$  – COMPLEX (KIND=nag\_wp) array

The  $n$ -element vector  $w$ .

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

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

Intermediate elements of  $\mathbf{w}$  are not referenced.

## 6 Error Indicators and Warnings

If  $\mathbf{inex} = 0$  or  $\mathbf{incy} = 0$  or  $\mathbf{inew} = 0$ , 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

$$\begin{aligned}\alpha &= 3 + 2i, & x &= (-6 + 1.2i, 3.7 + 4.5i, -4 + 2.1i)^T, \\ \beta &= -i, & y &= (-5.1, 6.4 - 5i, -3 - 2.4i)^T.\end{aligned}$$

$x$  and  $y$ , and also the sum vector  $w$ , are stored in reverse order.

## 9.1 Program Text

```

function f16gh_example

fprintf('f16gh example results\n\n');

n = nag_int(3);
x = [ -4 + 2.1i    3.7 + 4.5i    -6    + 1.2i];
y = [ -3 - 2.4i    6.4 - 5.0i    -5.1 + 0.0i];

% z = alpha*x +beta*y;
alpha = 3 + 2i;
beta = 0 - 1i;

fprintf('w = alpha*x + beta*y\n');
fprintf('alpha = %5.1f%+5.1fi      beta = %5.1f%+5.1fi\n', ...
real(alpha), imag(alpha), real(beta), imag(beta));

incx = nag_int(1);
incy = nag_int(1);
incw = nag_int(1);
[w] = f16gh( ...
n, alpha, x, incx, beta, y, incy, incw);

disp('          x                  y                  w');
disp([x; y; w']);

```

## 9.2 Program Results

```

f16gh example results

w = alpha*x + beta*y
alpha = 3.0 +2.0i      beta = 0.0 -1.0i
          x                  y                  w
-4.0000 - 2.1000i  -3.0000 + 2.4000i  -18.6000 + 1.3000i
 3.7000 - 4.5000i   6.4000 + 5.0000i  -2.9000 +14.5000i
-6.0000 - 1.2000i  -5.1000 + 0.0000i  -20.4000 - 3.3000i

```

---