NAG C Library Function Document

nag_daxpby (f16ecc)

1 Purpose

nag_daxpby (f16ecc) performs the operation

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

2 Specification

```
void nag_daxpby (Integer n, double alpha, const double x[], Integer incx,
     double beta, double y[], Integer incy, NagError *fail)
```

3 Description

nag_daxpby (f16ecc) 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 one, this function returns immediately.

4 References

The BLAS Technical Forum Standard (2001) *www.netlib.org/blas/blast-forum*

5 Parameters

1: **n** – Integer *Input*

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

Constraint: **n** ≥ 0.

2: **alpha** – double *Input*

On entry: the scalar α.

3: **x**[dim] – const double *Input*

Note: the dimension, dim, of the array **x** must be at least $1 + (\mathbf{n} - 1)|\mathbf{incx}|$.

On entry: the vector x.

4: **incx** – Integer *Input*

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

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

5: **beta** – double *Input*

On entry: the scalar β.

6: **y**[dim] – double *Input/Output*

Note: the dimension, dim, of the array **y** must be at least $1 + (\mathbf{n} - 1)|\mathbf{incy}|$.

On entry: the vector y.

On exit: the updated vector y.

7: **incy** – Integer *Input*

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

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

8: **fail** – NagError * *Input/Output*

The NAG error parameter (see the Essential Introduction).

6 Error Indicators and Warnings

NE_INT

On entry, **n** = <*value*>.
Constraint: **n** ≥ 0.

On entry, **incx** = <*value*>.
Constraint: **incx** $\neq 0$.

On entry, **incy** = <*value*>.
Constraint: **incy** $\neq 0$.

NE_BAD_PARAM

On entry, parameter <*value*> had an illegal value.

7 Accuracy

The BLAS standard requires accurate implementations which avoid unnecessary over/underflow (see section 2.7 of The BLAS Technical Forum Standard (2001)).

8 Further Comments

None.

9 Example

None.