NAG Library Function Document

nag_dsum (f16elc)

1 Purpose

nag_dsum (f16elc) sums the elements of a real vector.

2 Specification

```
#include <nag.h>
#include <nagf16.h>
double nag_dsum (Integer n, const double x[], Integer incx, NagError *fail)
```

3 Description

nag_dsum (f16elc) returns the sum

$$x_1 + x_2 + \cdots + x_n$$

of the elements of an n-element real vector x.

If $\mathbf{n} = 0$ on entry, nag_dsum (f16elc) returns the value 0.

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.

Constraint: $\mathbf{n} \geq 0$.

2: **x**[*dim*] – const double *Input*

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

On entry: the vector x. Element x_i is stored in $\mathbf{x}[(i - 1) \times |\mathbf{incx}|]$, 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: $\mathbf{incx} \neq 0$.

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

The NAG error argument (see Section 3.6 in the Essential Introduction).

6 Error Indicators and Warnings

NE_BAD_PARAM

On entry, argument $\langle value \rangle$ had an illegal value.

NE_INT

On entry, **incx** $= \langle value \rangle$.
Constraint: **incx** $\neq 0$.

On entry, **n** $= \langle value \rangle$.
Constraint: **n** ≥ 0.

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 sum of the elements of

$$x = (1.1, 10.2, 11.5, -2.7, 9.2)^{\mathrm{T}}.$$

10.1 Program Text

```
/* nag_dsum (f16elc) Example Program.
 *
 * Copyright 2005 Numerical Algorithms Group.
 *
 * Mark 9, 2009.
 */

#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagf16.h>

int main(void)
{
  /* Scalars */
  Integer  exit_status, i, incx, n, xlen;
  double   sumval;
  /* Arrays */
  double   *x = 0;
  /* Nag Types */
  NagError fail;

  exit_status = 0;
  INIT_FAIL(fail);

  printf("nag_dsum (f16elc) Example Program Results\n\n");

  /* Skip heading in data file */
  scanf("%*[^\n] ");
  /* Read the number of elements and the increment */
  scanf("%ld%ld%*[^\n] ", &n, &incx);

  xlen = MAX(1, 1 + (n - 1)*ABS(incx));

  if (n > 0)
    {
```

```
    /* Allocate memory */
    if (!(x = NAG_ALLOC(xlen, double)))
      {
        printf("Allocation failure\n");
        exit_status = -1;
        goto END;
      }
  }
else
  {
    printf("Invalid n\n");
    exit_status = 1;
    goto END;
  }
/* Input vector x */
for (i = 0; i < xlen; i = i + incx)
  scanf("%lf", &x[i]);
scanf("%*[^\n] ");

/* nag_dsum (f16elc).
 * Sum elements of a vector of doubles */
sumval = nag_dsum(n, x, incx, &fail);

if (fail.code != NE_NOERROR)
  {
    printf("Error from nag_dsum (f16elc).\n%s\n", fail.message);
    exit_status = 1;
    goto END;
  }

/* Print the result. */
printf("Sum of elements of x is %9.5f\n", sumval);

END:
 NAG_FREE(x);

 return exit_status;
}
```

10.2 Program Data

```
nag_dsum (f16elc) Example Program Data
  5   1                                                 : n and incx
  1.1   10.2   11.5   -2.7   9.2                        : Array x
```

10.3 Program Results

```
nag_dsum (f16elc) Example Program Results

Sum of elements of x is  29.30000
```