NAG Library Function Document

nag_dmax_val (f16jnc)

1 Purpose

nag_dmax_val (f16jnc) computes the largest component of a real vector, along with the index of that component.

2 Specification

```
#include <nag.h>
#include <nagf16.h>
```

```
void nag_dmax_val (Integer n, const double x[], Integer incx, Integer *k,
     double *r, NagError *fail)
```

3 Description

nag_dmax_val (f16jnc) computes the largest component, r, of an n-element real vector x, and determines the smallest index, k, such that

$$r = x_k = \max_j x_j.$$

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 n-element vector x.

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

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

Intermediate elements of **x** are not referenced. If $\mathbf{n} = 0$, **x** is not referenced and may be **NULL**.

3: **incx** – Integer *Input*

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

Constraint: $\mathbf{incx} \neq 0$.

4: **k** – Integer * *Output*

On exit: k, the index, from the set $\{0, 1, \ldots, \mathbf{n} - 1\}$, of the largest component of x. If $\mathbf{n} = 0$ on input then **k** is returned as -1.

5: **r** – double * *Output*

On exit: r, the largest component of x. If $\mathbf{n} = 0$ on input then **r** is returned as 0.0.

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

The NAG error argument (see Section 2.7 in How to Use the NAG Library and its Documentation).

6 Error Indicators and Warnings

NE_ALLOC_FAIL

Dynamic memory allocation failed.
See Section 3.2.1.2 in How to Use the NAG Library and its Documentation for further information.

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.

NE_INTERNAL_ERROR

An unexpected error has been triggered by this function. Please contact NAG.
See Section 3.6.6 in How to Use the NAG Library and its Documentation for further information.

NE_NO_LICENCE

Your licence key may have expired or may not have been installed correctly.
See Section 3.6.5 in How to Use the NAG Library and its Documentation for further information.

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

nag_dmax_val (f16jnc) is not threaded in any implementation.

9 Further Comments

None.

10 Example

This example computes the largest component and index of that component for the vector

$$x = (1, 10, 11, -2, 9)^{\mathrm{T}}.$$

10.1 Program Text

```
/* nag_dmax_val (f16jnc) Example Program.
 *
 * NAGPRODCODE Version.
 *
 * Copyright 2016 Numerical Algorithms Group.
 *
 * Mark 26, 2016.
 */

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

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

  exit_status = 0;
  INIT_FAIL(fail);

  printf("nag_dmax_val (f16jnc) Example Program Results\n\n");

  /* Skip heading in data file */
#ifdef _WIN32
  scanf_s("%*[^\n] ");
#else
  scanf("%*[^\n] ");
#endif

  /* Read the number of elements and the increment */
#ifdef _WIN32
  scanf_s("%" NAG_IFMT "%" NAG_IFMT "%*[^\n] ", &n, &incx);
#else
  scanf("%" NAG_IFMT "%" NAG_IFMT "%*[^\n] ", &n, &incx);
#endif

  if (n > 0) {
    /* Allocate memory */
    if (!(x = NAG_ALLOC(MAX(1, 1 + (n - 1) * ABS(incx)), double)))
    {
      printf("Allocation failure\n");
      exit_status = -1;
      goto END;
    }
  }
  else {
    printf("Invalid n\n");
    exit_status = 1;
    goto END;
  }

  /* Read the vector x and store forwards or backwards
   * as determined by incx. */
  for (i = 0, ix = (incx > 0 ? 0 : (1-n)*incx); i < n; i++, ix += incx)
#ifdef _WIN32
    scanf_s("%lf", &x[ix]);
#else
    scanf("%lf", &x[ix]);
#endif
#ifdef _WIN32
  scanf_s("%*[^\n] ");
#else
```

```
  scanf("%*[^\n] ");
#endif

  /* nag_dmax_val (f16jnc).
   * Get maximum value (r) and location of that value (k)
   * of double array */
  nag_dmax_val(n, x, incx, &k, &r, &fail);

  if (fail.code != NE_NOERROR) {
    printf("Error from nag_dmax_val (f16jnc).\n%s\n", fail.message);
    exit_status = 1;
    goto END;
  }

  /* Print the maximum value */
  printf("Maximum element of x is %12.5f\n", r);
  /* Print its location */
  printf("Index of maximum element of x is %3" NAG_IFMT "\n", k);

END:
  NAG_FREE(x);

  return exit_status;
}
```

10.2 Program Data

```
nag_dmax_val (f16jnc) Example Program Data
  5   1                                                 : n and incx
  1.0   10.0   11.0   -2.0   9.0                        : Vector x
```

10.3 Program Results

```
nag_dmax_val (f16jnc) Example Program Results

Maximum element of x is     11.00000
Index of maximum element of x is   2
```