NAG Toolbox for MATLAB

m01ca

1 Purpose

m01ca rearranges a vector of double numbers into ascending or descending order.

2 Syntax

```
[rv, ifail] = m01ca(rv, m1, order, 'm2', m2)
```

3 Description

m01ca is based on Singleton's implementation of the 'median-of-three' Quicksort algorithm (see Singleton 1969), but with two additional modifications. First, small subfiles are sorted by an insertion sort on a separate final pass (see Sedgewick 1978). Second, if a subfile is partitioned into two very unbalanced subfiles, the larger of them is flagged for special treatment: before it is partitioned, its end points are swapped with two random points within it; this makes the worst case behaviour extremely unlikely.

4 References

Sedgewick R 1978 Implementing Quicksort programs *Comm. ACM* **21** 847–857

Singleton R C 1969 An efficient algorithm for sorting with minimal storage: Algorithm 347 *Comm. ACM* **12** 185–187

5 Parameters

5.1 Compulsory Input Parameters

1: **rv**(**m2**) – **double array**

Elements **m1** to **m2** of **rv** must contain double values to be sorted.

2: **m1 – int32 scalar**

The index of the first element of **rv** to be sorted.

Constraint: **m1** > 0.

3: **order – string**

If **order** = 'A', the values will be sorted into ascending (i.e., nondecreasing) order.

If **order** = 'D', into descending order.

Constraint: **order** = 'A' or 'D'.

5.2 Optional Input Parameters

1: **m2 – int32 scalar**

Default: The dimension of the array **rv**.

the index of the last element of **rv** to be sorted.

Constraint: **m2** $\geq$ **m1**.

5.3 Input Parameters Omitted from the MATLAB Interface

None.

5.4 Output Parameters

1: **rv(m2) – double array**

These values are rearranged into sorted order.

2: **ifail – int32 scalar**

0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

On entry, **m2** < 1,
or **m1** < 1,
or **m1** $>$ **m2**.

ifail $= 2$

On entry, **order** is not 'A' or 'D'.

7 Accuracy

Not applicable.

8 Further Comments

The average time taken by m01ca is approximately proportional to $n \times \log n$, where $n = \mathbf{m2} - \mathbf{m1} + 1$. The worst case time is proportional to n^2 but this is extremely unlikely to occur.

9 Example

```
rv = [1.3;
      5.9;
      4.1;
      2.3;
      0.5;
      5.8;
      1.3;
      6.5;
      2.3;
      0.5;
      6.5;
      9.9;
      2.1;
      1.1;
      1.2;
      8.6];
m1 = int32(1);
order = 'Ascending';
[rvOut, ifail] = m01ca(rv, m1, order)
```

```
rvOut =
    0.5000
    0.5000
    1.1000
    1.2000
    1.3000
    1.3000
```

```
    2.1000
    2.3000
    2.3000
    4.1000
    5.8000
    5.9000
    6.5000
    6.5000
    8.6000
    9.9000
ifail =
           0
```