

NAG Toolbox for MATLAB

d03ry

1 Purpose

d03ry is designed to be used in conjunction with d03rb. It can be called from the user-supplied (sub)program **inidom** to check the user-specified initial grid data and to obtain a simple graphical representation of the initial grid.

2 Syntax

```
[pgrid, ifail] = d03ry(nx, ny, lrow, irow, icol, llbnd, ilbnd, lbnd,
'npts', npts, 'nrows', nrows, 'nbnds', nbnds, 'nbpts', nbpts)
```

3 Description

d03ry outputs a character array which can be printed to provide a simple graphical representation of the virtual and base grids supplied to d03rb. It must be called only from within the user-supplied (sub)program **inidom** after all output parameters of **inidom** (other than **ierr**) have been set. d03ry also checks the validity of the grid data specified in **inidom**.

You are strongly advised to call d03ry during the initial call of d03rb (at least) and to print the resulting character array in order to check that the base grid is exactly as required.

d03ry writes a representation of each point in the virtual and base grids to the character array **pgrid** as follows:

- internal base grid points are written as two dots (..);
- boundary base grid points are written as the **ilbnd** value (i.e., the type) of the boundary;
- points external to the base grid are written as XX.

As an example, consider a rectangular domain with a rectangular hole in which the virtual domain extends by one base grid point beyond the actual domain in all directions. The output when each row of **pgrid** is printed consecutively is as follows:

```
XX XX XX XX XX XX XX XX XX XX XX XX XX XX
XX 23  3  3  3  3  3  3  3  3  3  3  3 34 XX
XX  2  .. .. .. .. .. .. .. .. .. ..  4 XX
XX  2  .. .. .. .. .. .. .. .. .. ..  4 XX
XX  2  .. .. 14  1  1  1  1 21 .. ..  4 XX
XX  2  .. ..  4 XX XX XX XX  2  .. ..  4 XX
XX  2  .. ..  4 XX XX XX XX  2  .. ..  4 XX
XX  2  .. ..  4 XX XX XX XX  2  .. ..  4 XX
XX  2  .. ..  4 XX XX XX XX  2  .. ..  4 XX
XX  2  .. .. 43  3  3  3  3 32 .. ..  4 XX
XX  2  .. .. .. .. .. .. .. .. .. ..  4 XX
XX  2  .. .. .. .. .. .. .. .. .. ..  4 XX
XX 12  1  1  1  1  1  1  1  1  1  1  1 41 XX
XX XX XX XX XX XX XX XX XX XX XX XX XX XX
```

4 References

None.

5 Parameters

5.1 Compulsory Input Parameters

1: **nx** – **int32 scalar**

2: **ny** – **int32 scalar**

the number of virtual grid points in the x - and y -direction respectively (including the boundary points).

Constraint: **nx** and **ny** ≥ 4 .

3: **lrow(nrows)** – **int32 array**

lrow(i), for $i = 1, 2, \dots, \mathbf{nrows}$, contains the base grid index of the first grid point in base grid row i .

Constraints:

$$\begin{aligned} 1 &\leq \mathbf{lrow}(i) \leq \mathbf{npts}, \text{ for } i = 1, 2, \dots, \mathbf{nrows}; \\ \mathbf{lrow}(i-1) &< \mathbf{lrow}(i), \text{ for } i = 2, 3, \dots, \mathbf{nrows}. \end{aligned}$$

4: **irow(nrows)** – **int32 array**

irow(i), for $i = 1, 2, \dots, \mathbf{nrows}$, contains the virtual grid row number that corresponds to base grid row i .

Constraints:

$$\begin{aligned} 0 &\leq \mathbf{irow}(i) \leq \mathbf{ny}, \text{ for } i = 1, 2, \dots, \mathbf{nrows}; \\ \mathbf{irow}(i-1) &< \mathbf{irow}(i), \text{ for } i = 2, 3, \dots, \mathbf{nrows}. \end{aligned}$$

5: **icol(npts)** – **int32 array**

icol(i), for $i = 1, 2, \dots, \mathbf{npts}$, contains the virtual grid column number that contains base grid point i .

Constraint: $0 \leq \mathbf{icol}(i) \leq \mathbf{nx}$, for $i = 1, 2, \dots, \mathbf{npts}$.

6: **llbnd(nbnds)** – **int32 array**

llbnd(i), for $i = 1, 2, \dots, \mathbf{nbnds}$, contains the element of **lbnd** corresponding to the start of the i th boundary (or corner).

Constraints:

$$\begin{aligned} 1 &\leq \mathbf{llbnd}(i) \leq \mathbf{nbpts}, \text{ for } i = 1, 2, \dots, \mathbf{nbnds}; \\ \mathbf{llbnd}(i-1) &< \mathbf{llbnd}(i), \text{ for } i = 2, 3, \dots, \mathbf{nbnds}. \end{aligned}$$

7: **ilbnd(nbnds)** – **int32 array**

ilbnd(i), for $i = 1, 2, \dots, \mathbf{nbnds}$, contains the type of the i th boundary (or corner), as defined in d03rb.

Constraint: **ilbnd**(i) must be equal to one of the following: 1, 2, 3, 4, 12, 23, 34, 41, 21, 32, 43 or 14, for $i = 1, 2, \dots, \mathbf{nbnds}$.

8: **lbnd(nbpts)** – **int32 array**

lbnd(i), for $i = 1, 2, \dots, \mathbf{nbpts}$, contains the grid index of the i th boundary point.

Constraint: $1 \leq \mathbf{lbnd}(i) \leq \mathbf{npts}$, for $i = 1, 2, \dots, \mathbf{nbpts}$.

5.2 Optional Input Parameters

1: **npts** – int32 scalar

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

the total number of points in the base grid.

Constraint: $\mathbf{npts} \leq \mathbf{nx} \times \mathbf{ny}$.

2: **nrows** – int32 scalar

Default: The dimension of the arrays **lrow**, **irow**. (An error is raised if these dimensions are not equal.)

the total number of rows of the virtual grid that contain base grid points.

Constraint: $4 \leq \mathbf{nrows} \leq \mathbf{ny}$.

3: **nbnds** – int32 scalar

Default: The dimension of the arrays **llbnd**, **ilbnd**. (An error is raised if these dimensions are not equal.)

the total number of physical boundaries and corners in the base grid.

Constraint: $\mathbf{nbnds} \geq 8$.

4: **nbpts** – int32 scalar

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

the total number of boundary points in the base grid.

Constraint: $12 \leq \mathbf{nbpts} < \mathbf{npts}$.

5.3 Input Parameters Omitted from the MATLAB Interface

iwk, leniwk

5.4 Output Parameters

1: **pgrid**(ny) – string array

pgrid(*i*), for $i = 1, 2, \dots, \mathbf{ny}$, contains a graphical representation of row $\mathbf{ny} - i + 1$ of the virtual grid (see Section 3).

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, **nx** or **ny** < 4,
 or **npts** > $\mathbf{nx} \times \mathbf{ny}$,
 or **nrows** < 4,
 or **nrows** > **ny**,
 or **nbnds** < 8,
 or **nbpts** < 12,
 or **nbpts** ≥ **npts**,
 or **lrow**(*i*) < 1, for some $i = 1, 2, \dots, \mathbf{nrows}$,
 or **lrow**(*i*) > **npts**, for some $i = 1, 2, \dots, \mathbf{nrows}$,

or $\mathbf{lrow}(i) \leq \mathbf{lrow}(i-1)$, for some $i = 2, 3, \dots, \mathbf{nrows}$,
 or $\mathbf{irow}(i) < 0$, for some $i = 1, 2, \dots, \mathbf{nrows}$,
 or $\mathbf{irow}(i) > \mathbf{ny}$, for some $i = 1, 2, \dots, \mathbf{nrows}$,
 or $\mathbf{irow}(i) \leq \mathbf{irow}(i-1)$, for some $i = 2, 3, \dots, \mathbf{nrows}$,
 or $\mathbf{icol}(i) < 0$, for some $i = 1, 2, \dots, \mathbf{npts}$,
 or $\mathbf{icol}(i) > \mathbf{nx}$, for some $i = 1, 2, \dots, \mathbf{npts}$,
 or $\mathbf{llbnd}(i) < 1$, for some $i = 1, 2, \dots, \mathbf{nbnds}$,
 or $\mathbf{llbnd}(i) > \mathbf{nbpts}$, for some $i = 1, 2, \dots, \mathbf{nbnds}$,
 or $\mathbf{llbnd}(i) \leq \mathbf{llbnd}(i-1)$, for some $i = 2, 3, \dots, \mathbf{nbpts}$,
 or $\mathbf{ilbnd}(i) \neq 1, 2, 3, 4, 12, 23, 34, 41, 21, 32, 43$ or 14 , for some $i = 1, 2, \dots, \mathbf{nbnds}$,
 or $\mathbf{lbnd}(i) < 1$, for some $i = 1, 2, \dots, \mathbf{nbpts}$,
 or $\mathbf{lbnd}(i) > \mathbf{npts}$, for some $i = 1, 2, \dots, \mathbf{nbpts}$,
 or $\mathbf{leniwk} < \mathbf{nx} \times \mathbf{ny} + 1$,
 or $\mathbf{LEN}(\mathbf{pgrid}(1)) < 3 \times \mathbf{nx}$.

7 Accuracy

Not applicable.

8 Further Comments

None.

9 Example

```

nx = int32(11);
ny = int32(11);
lrow = [int32(1);
        int32(4);
        int32(15);
        int32(26);
        int32(37);
        int32(46);
        int32(57);
        int32(68);
        int32(79);
        int32(88);
        int32(97)];
irow = [int32(0);
        int32(1);
        int32(2);
        int32(3);
        int32(4);
        int32(5);
        int32(6);
        int32(7);
        int32(8);
        int32(9);
        int32(10)];
icol = [int32(0);
        int32(1);
        int32(2);
        int32(0);
        int32(1);
        int32(2);
        int32(3);
        int32(4);
        int32(5);
        int32(6);
        int32(7);
        int32(8);
        int32(9)];

```

```
int32(10);
int32(0);
int32(1);
int32(2);
int32(3);
int32(4);
int32(5);
int32(6);
int32(7);
int32(8);
int32(9);
int32(10);
int32(0);
int32(1);
int32(2);
int32(3);
int32(4);
int32(5);
int32(6);
int32(7);
int32(8);
int32(9);
int32(10);
int32(0);
int32(1);
int32(2);
int32(3);
int32(4);
int32(5);
int32(8);
int32(9);
int32(10);
int32(0);
int32(1);
int32(2);
int32(3);
int32(4);
int32(5);
int32(6);
int32(7);
int32(8);
int32(9);
int32(10);
int32(0);
int32(1);
int32(2);
int32(3);
int32(4);
int32(5);
int32(6);
int32(7);
int32(8);
int32(9);
int32(10);
int32(0);
int32(1);
int32(2);
int32(3);
int32(4);
int32(5);
int32(6);
int32(7);
int32(8);
int32(9);
int32(10);
int32(0);
int32(1);
int32(2);
int32(3);
int32(4);
```

```

        int32(5);
        int32(6);
        int32(7);
        int32(8);
        int32(0);
        int32(1);
        int32(2);
        int32(3);
        int32(4);
        int32(5);
        int32(6);
        int32(7);
        int32(8);
        int32(0);
        int32(1);
        int32(2);
        int32(3);
        int32(4);
        int32(5);
        int32(6);
        int32(7);
        int32(8)];
11bnd = [int32(1);
        int32(2);
        int32(11);
        int32(18);
        int32(19);
        int32(24);
        int32(31);
        int32(37);
        int32(42);
        int32(48);
        int32(53);
        int32(55);
        int32(56);
        int32(58);
        int32(59);
        int32(60);
        int32(61);
        int32(62);
        int32(63);
        int32(64);
        int32(65);
        int32(66);
        int32(67);
        int32(68);
        int32(69);
        int32(70);
        int32(71);
        int32(72)];
ilbnd = [int32(1);
        int32(2);
        int32(3);
        int32(4);
        int32(1);
        int32(4);
        int32(1);
        int32(2);
        int32(3);
        int32(4);
        int32(3);
        int32(4);
        int32(1);
        int32(2);
        int32(12);
        int32(23);
        int32(34);
        int32(41);
        int32(14);
        int32(41);

```

```

        int32(12);
        int32(23);
        int32(34);
        int32(41);
        int32(43);
        int32(14);
        int32(21);
        int32(32)];
    lbnd = [int32(2);
        int32(4);
        int32(15);
        int32(26);
        int32(37);
        int32(46);
        int32(57);
        int32(68);
        int32(79);
        int32(88);
        int32(98);
        int32(99);
        int32(100);
        int32(101);
        int32(102);
        int32(103);
        int32(104);
        int32(96);
        int32(86);
        int32(85);
        int32(84);
        int32(83);
        int32(82);
        int32(70);
        int32(59);
        int32(48);
        int32(39);
        int32(28);
        int32(17);
        int32(6);
        int32(8);
        int32(9);
        int32(10);
        int32(11);
        int32(12);
        int32(13);
        int32(18);
        int32(29);
        int32(40);
        int32(49);
        int32(60);
        int32(72);
        int32(73);
        int32(74);
        int32(75);
        int32(76);
        int32(77);
        int32(67);
        int32(56);
        int32(45);
        int32(36);
        int32(25);
        int32(33);
        int32(32);
        int32(42);
        int32(52);
        int32(53);
        int32(43);
        int32(1);
        int32(97);
        int32(105);
        int32(87);

```

```

    int32(81);
    int32(3);
    int32(7);
    int32(71);
    int32(78);
    int32(14);
    int32(31);
    int32(51);
    int32(54);
    int32(34)];
    [pgrid, ifail] = d03ry(nx, ny, lrow, irow, icol, llbnd, ilbnd, lbnd)

```

```

pgrid =
    ' 23  3  3  3  3  3  3  3 34 XX XX'
    '  2 .. .. .. .. .. .. 4 XX XX'
    '  2 .. 14  1  1  1  1  1 41 XX XX'
    '  2 ..  4 23  3  3  3  3  3 34'
    '  2 ..  4  2 .. .. .. .. .. 4'
    '  2 ..  4  2 .. 14  1  1 21 .. 4'
    '  2 ..  4  2 ..  4 XX XX  2 .. 4'
    '  2 ..  4  2 .. 43  3  3 32 .. 4'
    '  2 ..  4  2 .. .. .. .. .. 4'
    '  2 ..  4 12  1  1  1  1  1  1 41'
    ' 12  1 41 XX XX XX XX XX XX XX XX'
ifail =
    0

```