NAG Toolbox for MATLAB

d03ma

1 Purpose

d03ma places a triangular mesh over a given two-dimensional region. The region may have any shape, including one with holes.

2 Syntax

```
[npts, places, indx, ifail] = d03ma(h, m, n, nb, sdindx, isin)
```

3 Description

d03ma begins with a uniform triangular grid as shown in Figure 1 and assumes that the region to be triangulated lies within the rectangle given by the inequalities

$$0 < x < \sqrt{3}(m-1)h, \qquad 0 < y < (n-1)h.$$

This rectangle is drawn in bold in Figure 1. The region is specified by the user-supplied integer function **isin** which must determine whether any given point (x,y) lies in the region. The uniform grid is processed column-wise, with (x_1,y_1) preceding (x_2,y_2) if $x_1 < x_2$ or $x_1 = x_2$, $y_1 < y_2$. Points near the boundary are moved onto it and points well outside the boundary are omitted. The direction of movement is chosen to avoid pathologically thin triangles. The points accepted are numbered in exactly the same order as the corresponding points of the uniform grid were scanned. The output consists of the x,y co-ordinates of all grid points and integers indicating whether they are internal and to which other points they are joined by triangle sides.

The mesh size h must be chosen small enough for the essential features of the region to be apparent from testing all points of the original uniform grid for being inside the region. For instance if any hole is within $2h$ of another hole or the outer boundary then a triangle may be found with all vertices within $\frac{1}{2}h$ of a boundary. Such a triangle is taken to be external to the region so the effect will be to join the hole to another hole or to the external region.

Further details of the algorithm are given in the references.

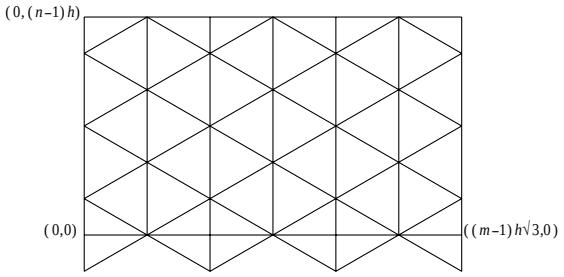

Figure 1

4 References

Reid J K 1970 Fortran subroutines for the solutions of Laplace's equation over a general routine in two dimensions *Harwell Report TP422*

Reid J K 1972 On the construction and convergence of a finite-element solution of Laplace's equation *J. Instr. Math. Appl.* **9** 1–13

5 Parameters

5.1 Compulsory Input Parameters

1: **h – double scalar**

h, the required length for the sides of the triangles of the uniform mesh.

2: **m – int32 scalar**

3: **n – int32 scalar**

Values m and n such that all points (x,y) inside the region satisfy the inequalities

$$0 \leq x \leq \sqrt{3}(m-1)h,$$
$$0 \leq y \leq (n-1)h.$$

Constraint: **m**, **n** > 2.

4: **nb – int32 scalar**

The number of times a triangle side is bisected to find a point on the boundary. A value of 10 is adequate for most purposes (see Section 7).

Constraint: **nb** ≥ 1.

5: **sdindx – int32 scalar**

6: **isin – string containing name of m-file**

isin must return the value 1 if the given point (**x**,**y**) lies inside the region, and 0 if it lies outside.

Its specification is:

```
[result] = isin(x, y)
```

Input Parameters

1: **x – double scalar**

2: **y – double scalar**

The co-ordinates of the given point.

Output Parameters

1: **result – int32 scalar**

The result of the function.

5.2 Optional Input Parameters

None.

5.3 Input Parameters Omitted from the MATLAB Interface

dist, sddist

5.4 Output Parameters

1: **npts – int32 scalar**

The number of points in the triangulation.

2: **places(2,sdindx) – double array**

The x and y co-ordinates respectively of the ith point of the triangulation.

3: **indx(4,sdindx) – int32 array**

indx$(1,i)$ contains i if point i is inside the region and $-i$ if it is on the boundary. For each triangle side between points i and j with $j > i$, **indx**(k,i), $k > 1$, contains j or $-j$ according to whether point j is internal or on the boundary. There can never be more than three such points. If there are less, then some values **indx**(k,i), $k > 1$, are zero.

4: **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 $= 1$

sdindx is too small.

ifail $= 2$

A point inside the region violates one of the constraints (see parameters **m** and **n**).

ifail $= 3$

sddist is too small.

ifail $= 4$

m ≤ 2.

ifail $= 5$

n ≤ 2.

ifail $= 6$

nb ≤ 0.

7 Accuracy

Points are moved onto the boundary by bisecting a triangle side **nb** times. The accuracy is therefore $h \times 2^{-\mathbf{nb}}$.

8 Further Comments

The time taken is approximately proportional to $m \times n$.

9 Example

d03ma_isin.m

```
function result = in(x,y)
      if ((x-7)^2+(y-7)^2 <= 36)
         result = int32(1);
      else
         result = int32(0);
```

```
      end
```

```
h = 4;
m = int32(3);
n = int32(5);
nb = int32(10);
sdindx = int32(100);
[npts, places, index, ifail] = d03ma(h, m, n, nb, sdindx, 'd03ma_isin')
```

```
npts =
          14
places =
     array elided
index =
     array elided
ifail =
           0
```