# An *EasyMesh* Example

Andreas Stahel

24. November 1999

## 1 Introduction

The program *EasyMesh*[1] generates meshes for domain in $\mathbb{R}^2$.

As a simple example we consider a rectangle $0 < x < 5$ and $0 < y < 4$. We aim for a mesh with very few nodes. We consider a problem with Dirichlet condition at the lower and upper boundary (type 1) and Neumann conditions at the left and right boundary (type 2). The input file `test1.d` does contain all necessary information (see the documentation of *EasyMesh*) and the command `EasyMesh test1` will generate three output files:

| | |
|---|---|
| `test1.d` | input file with all information on the domain |
| `test1.n` | output file with data of the nodes |
| `test1.e` | output file with data of the elements |
| `test1.s` | output file with data of the line segments |

On X Window System the mesh can be visualized by the command `ViewMesh test1`. A window should pop up, showing the generated mesh and information about the mesh. Both programs are installed on the Solaris system at the HTA Biel.

In this not we consider this very simple example, illustrated by the figures below. The original documentation and source for EasyMesh is also available through a WWW page at the HTA Biel[2].

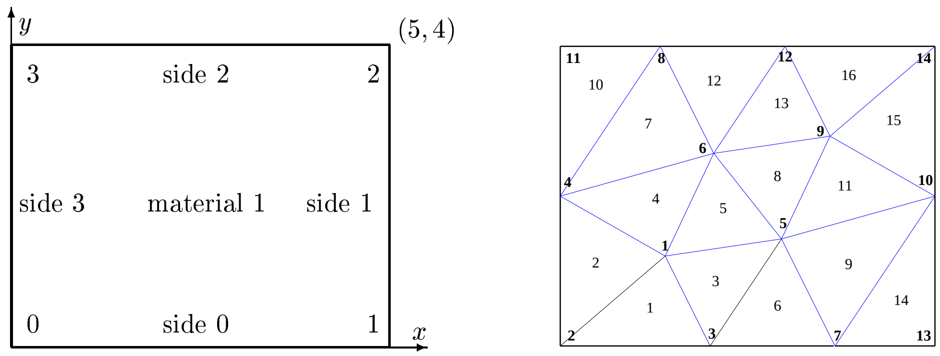


Figure 1: Input information for *EasyMesh* and the resulting mesh

[1] `http://www-dinma.univ.trieste.it/~nirftc/research/easymesh/`

[2] `http://www.hta-bi.bfh.ch/~sha` through the node *classroom notes*

## 2 Format of the input file

The input file contains all information about the mesh that has to be generated. The extension of the filename has to be `.d`, e.g. `test1.d` . Comments begin and end with `#` and should be used to document your mesh. Numbering of nodes and segments has to start at 0 and not at 1, as one might expect.

- The total number of nodes, whose data is given on subsequent lines
- For each node give the following information on one line
  1. The number of the node, starting at 0, followed by a column "':"'
  2. The $x$ and $y$ coordinates of the node, given as rational or integer numbers
  3. The typical length of a edge of a triangle to be generated close to this point. This allow to specify the size of the mesh, the smaller this number, the more triangles will be generated. The typical length has to be strictly larger than 0, otherwise to program *EasyMesh* might fail.
  4. A boundary marker, which we do not use in the current setup.
- The total number of boundary segments
- For each boundary segment give the following information on one line
  1. The number of the segment, starting at 0, followed by a column "':"'
  2. The number of the first node, then the number of the second node
  3. A boundary marker, to indicate the type of boundary condition to be used on this segment. In the current code a Dirichlet condition is generated by a 1 as marker and a 2 leads to a Neumann condition.
- Node which define the boundary of a domain have to be given in anti-clockwise direction. A clockwise direction represents a hole in the domain.

| | | | | |
|---|---|---|---|---|
| Number of nodes | | | | |
| for each node entry | | | | |
| Number of node: | $x$ value | $y$ value | mesh length | boundary marker |
| Number of segments | | | | |
| for each segment | | | | |
| Number of segment: | start point | end point | boundary marker | |

```
# this is the file test1.d #
#=========
| POINTS |
=========#
5 # number of points #

# Nodes which define the boundary #
0:   0   0   2   1
1:   5   0   2   1
2:   5   4   2   1
3:   0   4   2   1

# material marker #
4: 2   2    0   1 # material 1 #

#===========
| SEGMENTS |
===========#
4 # Number of segments #

# Boundary segments #
0:   0   1   1
1:   1   2   2
2:   2   3   1
3:   3   0   2
```

# 3 Format of the output files

Calling *EasyMesh* with an input file leads to three output files for nodes, elements and segments.

## 3.1 Format of the node file

This file has the extension `.n`, e.g. `test1.n` and gives the total number of nodes in the mesh and there coordinates. The example below illustrates the format and needs no further explanation. Observe that the numbering of the nodes does **not** coincide with the numbering in the input file.

```
14
  0:  1.400000000000000e+00 1.199965932698794e+00  0
  1:  0.000000000000000e+00 0.000000000000000e+00  1
  2:  2.000000000000000e+00 0.000000000000000e+00  1
  3:  0.000000000000000e+00 2.000000000000000e+00  2
  4:  2.952380950412123e+00 1.428531732976555e+00  0
  5:  2.047619049587878e+00 2.571388878335854e+00  0
  6:  3.666666666666667e+00 0.000000000000000e+00  1
  7:  1.333333333333333e+00 4.000000000000000e+00  1
  8:  3.600000000000000e+00 2.799965932698794e+00  0
  9:  5.000000000000000e+00 2.000000000000000e+00  2
 10:  0.000000000000000e+00 4.000000000000000e+00  1
 11:  3.000000000000000e+00 4.000000000000000e+00  1
 12:  5.000000000000000e+00 0.000000000000000e+00  1
 13:  5.000000000000000e+00 4.000000000000000e+00  1
-----------------------------------------------------------
  n:  x                     y                       mark
```

## 3.2 Format of the element file

This file has the extension `.e`, e.g. `test1.e` and lists the information about all triangles in the generated mesh. The first line gives the total number of elements (triangles). Each line below contains information about one element.

1. The number of the element, followed by a column "‘:"’
2. The numbers of the three nodes
3. The numbers of the three elements that share an edge with this element and then the numbers of the three element sides.
4. The $x$ and $y$ coordinates of the center of the circle through the three nodes. A material marker, this information is only useful if more than one material is to be used.

The only information used in the current code are the number of the elements and the numbers of the nodes.

```
16
  0:     0    1    2    -1    2    1     2    1    0  1.0000000000e+00 2.4997302977e-01      1
  1:     0    3    1    -1    0    3     4    0    3  3.5713799080e-01 1.0000000000e+00      1
  2:     0    2    4     5    4    0     7    5    1  2.2414913239e+00 8.7073631484e-01      1
  3:     0    5    3     6    1    4     9    3    6  1.0461342199e+00 2.2056920576e+00      1
  4:     0    4    5     7    3    2    10    6    5  2.1195725236e+00 1.6987885555e+00      1
  5:     4    2    6    -1    8    2     8   12    7  2.8333333333e+00 4.7616401211e-01      1
  6:     3    5    7    11    9    3    14   11    9  8.6055134800e-01 2.8707435457e+00      1
  7:     4    8    5    12    4   10    15   10   13  2.8804307151e+00 2.3011346198e+00      1
  8:     4    6    9    13   10    5    18   16   12  4.1394629095e+00 1.1292469491e+00      1
  9:     3    7   10    -1   -1    6    20   17   11  6.6666666666e-01 3.0000000000e+00      1
 10:     4    9    8    14    7    8    21   13   16  3.9538723143e+00 1.7942337211e+00      1
 11:     5   11    7    -1    6   12    23   14   19  2.1666666666e+00 3.5237830620e+00      1
 12:     5    8   11    15   11    7    24   19   15  2.7584981290e+00 3.1292397171e+00      1
 13:     9    6   12    -1   -1    8    22   25   18  4.3333333333e+00 9.9999999999e-01      1
 14:     8    9   13    -1   15   10    27   26   21  4.6428522756e+00 3.0000000000e+00      1
 15:     8   13   11    -1   12   14    28   24   26  4.0000000000e+00 3.7499730303e+00      1
-------------------------------------------------------------------------------------------------
  e:    i,   j,   k,   ei,  ej,  ek,   si,  sj,  sk  xV,              yV                  sign
```

## 3.3 Format of the segment file

This file has the extension `.s`, e.g. `test1.s` and lists the information about all segments in the mesh. We only need the boundary segments in our current code. The first line gives the total number of segments (sides). Each line below contains information about one segment.

1. The number of the segment, followed by a column "':"'
2. The number of the starting node and the number of the end node of the segment.
3. The number of the element on the left side and the number of the element on the right side. If one of those numbers equals $-1$ then there is no element on that side and the segment is thus a boundary segment of the domain.
4. The boundary marker, indicating the type of segment
   - 0 : interior segment
   - 1 : Dirichlet boundary
   - 2 : Neumann boundary

We only use the segments on the boundary.

```
29
   0:     1    0    1    0  0
   1:     2    0    0    2  0
   2:     1    2    0   -1  1
   3:     3    0    3    1  0
   4:     1    3   -1    1  2
   5:     4    0    2    4  0
   6:     5    0    4    3  0
   7:     2    4    2    5  0
   8:     2    6    5   -1  1
   9:     3    5    6    3  0
  10:     5    4    7    4  0
  11:     7    3    6    9  0
  12:     6    4    5    8  0
  13:     4    8    7   10  0
  14:     7    5   11    6  0
  15:     5    8   12    7  0
  16:     9    4    8   10  0
  17:    10    3    9   -1  2
  18:     6    9    8   13  0
  19:    11    5   12   11  0
  20:     7   10    9   -1  1
  21:     9    8   10   14  0
  22:     6   12   13   -1  1
  23:    11    7   11   -1  1
  24:    11    8   15   12  0
  25:    12    9   13   -1  2
  26:    13    8   14   15  0
  27:     9   13   14   -1  2
  28:    13   11   15   -1  1
--------------------------------
   s:    c    d   ea   eb   mark
```