The ADCIRC file format

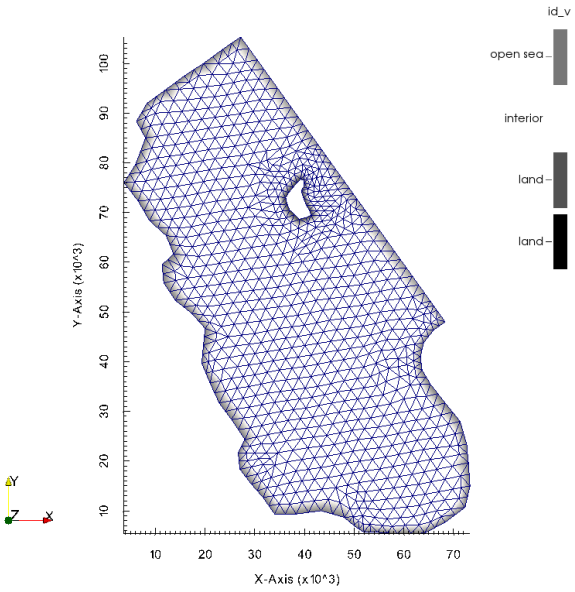

Figure 1: A plot of the given mesh.

ADCIRC (`http://adcirc.org`) is an ocean circulation model with its own file format to describe input meshes. Although there are many other file formats from other softwares to describe meshes, they often take a similar form and we chose the ADCIRC format for being relatively simple. The purpose of this programming task is for you to get a feeling for the handling of meshes in real applications.

An ADCIRC mesh file is an ASCII text-file that provides geometry (node coordinates), connectivity (nodes per element), and boundary information. These parts are given in a specific order, which allows to read the file line-by-line without having to seek for information in it. It supports annotations in form of end-of line-comments, which are put after the 'useful' information on a line and begin with '`!`'.

The general file structure and an example file for a simple mesh is given below. At the end of this document are some information on how to store the values from the file in the data structures of the template.

The general file structure

ADCIRC is an ocean simulator and each point's coordinates have a third component relating to the depth. You can ignore this value. Furthermore, you can assume that no line has more than 120 characters. Make sure you include proper checks for errors (invalid/incomplete files, failed reads, wrong ordering of nodes, elements, etc.). You may assume that the nodes within an element are always in counter-clockwise order. All comments must be ignored. In `C`, strings are terminated by '`\0`' and it is sufficient to replace a found '`!`' by '`\0`' to ignore anything after it.

Be aware that in the ADCIRC file format all indices start counting from 1. However, in `C` it is more convenient to start counting from 0 – this means, you have to decrease all indices in this file by 1 while reading it in.

```
mesh_name               ! Any sequence of characters
ne np                   ! Number of elements (ne) and points (ne)
1 x_1 y_1 depth_1       ! Vertex ID, X, Y and depth
2 x_2 y_2 depth_2
...
np x_np y_np depth_np
1 n i_11 i_12 ... i_1n ! Element ID, number of points and point IDs
2 n i_21 i_22 ... i_2n
...
ne n i_np1 i_np2 ... i_npn
N_NOPE                  ! Number of open sea boundary segments
n_NOPE                  ! Total number of points on open sea boundaries
n_NOPE_1                ! Number of points on open sea bnd segment 1
i_1                     ! vertex IDs
i_2
...
i_n_NOPE_1
n_NOPE_2                ! Number of points on open sea bnd segment 2
i_1
...
i_n_NOPE_2
...                     ! Further open sea boundary segments
N_NBOU                  ! Number of land boundary segments
n_NBOU                  ! Total number of points on land boundaries
n_NBOU_1 t_NBOU_1       ! Number points on land bnd 1 and type
i_1                     ! vertex IDs
i_2
...
i_n_NBOU_1
...                     ! Further land boundary segments
```

Boundaries can be split into multiple segments (e. g., the given mesh in figure 1 has two *land* boundaries: one on the outside and another at the island

in the interior). Each segment has its own block, beginning with the number of vertices in this block (e. g., `n_NOPE_1`) and followed by the IDs of all vertices, each in a separate line.

Example file for a simple mesh

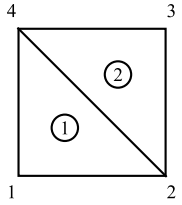

Figure 2: Illustration of an example mesh with vertex- and element-IDs.

For the simple mesh given in figure 2 with vertices $(0, 0)$, $(1, 0)$, $(1, 1)$, $(0, 1)$, and a single *open sea* boundary, an ADCIRC file takes the following form:

```
Testmesh
2 4                 ! name
1 0.00 0.00 0.00    ! vertex coordinates
2 1.00 0.00 0.00
3 1.00 1.00 0.00
4 0.00 1.00 0.00
1 3 1 2 4           ! triangle-vertex connectivity
2 3 2 3 4
1                   ! N_NOPE
4                   ! n_NOPE
4                   ! n_NOPE_1
1
2
3
4
0                   ! N_NBOU
0                   ! n_NBOU
```

Where to put things

In the first line of an ADCIRC file is the name of the mesh. This is stored in a new field `name` in the `mesh` data structure.

The number of elements and nodes allow to allocate `coords`, `t2v`, and `id_v`. Coordinates and triangle-vertex connectivity can then be stored in `coords` and `t2v`.

The rest of the file holds boundary information and allows to fill `id_v`.

All *open sea* boundaries should be considered as Neumann boundaries. Each *open sea* boundary segment should get a separate increasing ID in `id_v`, starting by 1.

All *land* boundary segments have a second parameter `t_NBOU` which gives the type of the boundary. Set their ID to `100+t_NBOU` and treat them as Dirichlet boundary nodes.