MTB Challenge – Summer Term 2016/2017

Čapek, M., Štrambach, M., Adler, V.

March 17, 2017

1 Introduction

Let us suppose a planar polygon, discretized with Delaunay triangulation [1] as $\Omega \to \Omega^N$ into N triangles with E inner edges, see Fig. 1. A graph can be constructed using this unstructured grid:

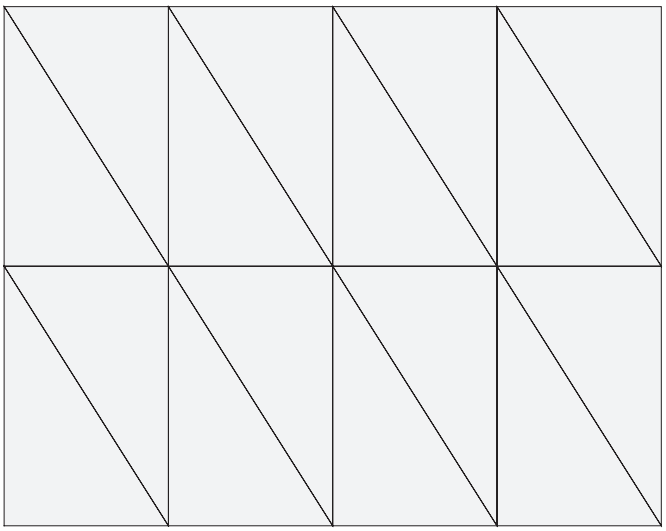


Figure 1: Polygon discretized using Delaunay triangulation.

- Undirected graph connecting all adjacent triangles. All corresponding center points of the neighbouring triangles are connected, see Fig. 2. This graph is called here T-graph.

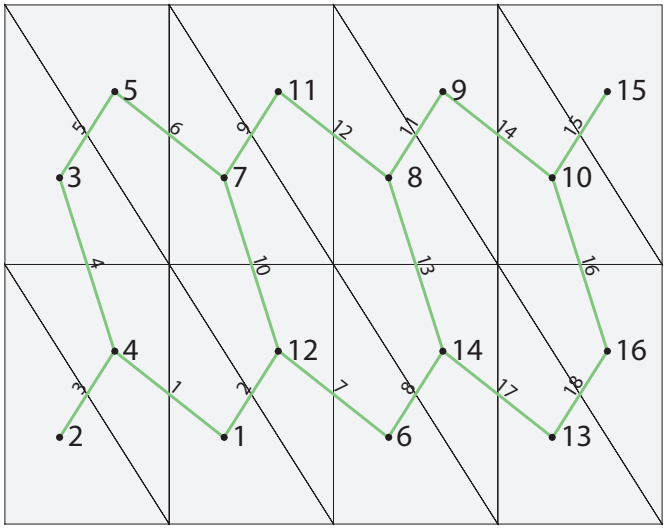
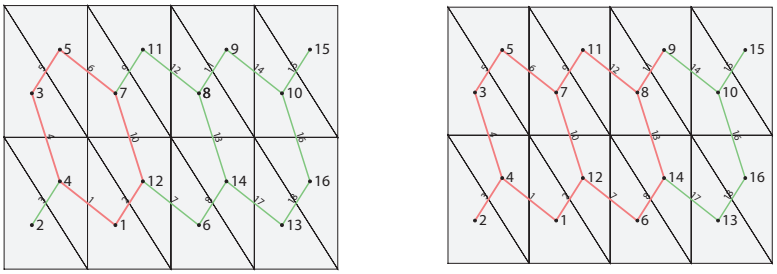


Figure 2: T-graph.

Considering T-graph, we can construct cycles like highlighted in Fig. 3a. We can also calculate the length of the cycle (minimum length is 3), maximum length is proportional to the number of edges forming the loop, *e.g.*, the cycle depicted in Fig. 3a has length $l_T = 6$. Notice that valid cycles are only those sharing all points only for two edges, therefore, cycles in Fig. 3b are invalid.



(a) Valid cycle (b) Invalid cycle

Figure 3: Example of subgraphs.

2 Competition Setting

The task is to find all cycles with their length $l_T \geq L_T - 5$, in which L_T is the longest loop existing in T-graph. Prepare function called `C = findCycles(nodes, connectivityList)`, in which `C` is a Matlab cell containing all cycles defined by their edges sorted with respect to their length. Illustrative example of all cycles with $l_T \geq L_T$ for a graph from Fig. 2 is attached in Section 4. Edges in a cycle are sorted in ascending order. Variable `nodes` is a Nx2 matrix with node coordinates and `connectivityList` is Nx3 represents triangle connectivity, see documentation to Delaunay triangulation in Matlab [2]. Nodes of T-graph are numbered based on a triangle position in the `connectivityList`. Every edge in T-graph has two end nodes. Edges are numbered in lexicographical order of their end nodes.
Another part of your task is to visualize results from cell `C`. You must be able to show T-graph with higlighted cycles, but method you choose is completely up to you. Be creative! *e.g.* you can create simple GUI with a running animation with cycles of a given length.

3 Criteria

- This project can be selected by unlimited number of students. However, no collaboration between students is expected.
- Project should be submitted including short documentation describing how the algorithm works.
- Like for regular projects, short presentation (couple of minutes) is expected.
- To be awarded with credits, it is enough to find all cycles according requirements in prevision Section 2.
- To participate in the competition (and have a chance to get some awards), both sets of cycles have to be found (test Matlab file calculating correct solution will be available). In that case, the computational time required to get the correct solution will be measured on reference PC (Win7 + up-to-date edition on Matlab). Fist three student with the fastest codes will be awarded. List of awards is attached at the end of the document.
- No toolboxes or external codes and libraries (dll, mex) are allowed.
- It is possible to always withdraw from the competition and select of of regular projects. This decision should be discussed with lecturers and their approval is required.

4 Testing Example

Example input

```
nodes =  [
         0          0          0;
         0     1.5000          0;
         0     3.0000          0;
    0.5000          0          0;
    0.5000     1.5000          0;
    0.5000     3.0000          0;
    1.0000          0          0;
    1.0000     1.5000          0;
    1.0000     3.0000          0;
    1.5000          0          0;
    1.5000     1.5000          0;
    1.5000     3.0000          0;
    2.0000          0          0;
    2.0000     1.5000          0;
    2.0000     3.0000          0;];

connectivityList = [
     4     7     5;
     1     4     2;
     5     3     2;
     4     5     2;
     5     6     3;
     8     7    10;
     5     8     6;
     8    11     9;
    11    12     9;
    11    14    12;
     6     8     9;
     5     7     8;
    11    10    13;
    11     8    10;
    12    14    15;
    11    13    14;];
```

Example output

```
C = {
        []
        []
        [1  2  4  5  6  10
         7  8  9  10  12  13
         11  13  14  16  17  18]
        [1  2  4  5  6  7  8  9  12  13
         7  8  9  10  11  12  14  16  17  18]
        [1  2  4  5  6  7  8  9  11  12  14  16  17  18]
    }
```

5 List of Awards

Following awards will be available:

- student licences of Matlab (3×)
- training course in advance techniques in Matlab (OOP, GUI,...), provided by Humusoft company
- pint glass with logo of Department of Electromagnetic Field
- and others...

References

[1] J. A. De Loera, J. Rambau, and F. Santos, *Triangulations – Structures for Algorithms and Applications*. Berlin, Germany: Springer, 2010.

[2] (2016) The Matlab. The MathWorks. [Online]. Available: www.mathworks.com