

```

1 #include "GenericDomain.hpp"
2 #include "GenericMesh.hpp"
3 #include "MeshImport_Triangle.hpp"
4 #include <iostream>
5 #include "TriangleRefiner.hpp"
6
7 using namespace GeDiM;
8 using namespace Eigen;
9
10 int main(int argc, char *argv[])
11 {
12     /// PARAMETRI
13     double cellsize;
14     int percentuale;
15     if (argc != 3)
16     {
17         cerr << "Uso corretto: progetto.bin <cellsize> <percentuale>\n";
18         exit(EXIT_FAILURE);
19     }
20     else
21     {
22         cellsize = stod(argv[1]);
23         percentuale = stoi(argv[2]);
24     }
25
26     /// CREATE DOMAIN
27     const unsigned int numDomainVertices = 4;
28     GenericDomain2D domain(0, numDomainVertices);
29     vector<Vector3d> vertexCoords(numDomainVertices);
30     vertexCoords[0] << 0.0, 0.0, 0.0;
31     vertexCoords[1] << 1.0, 0.0, 0.0;
32     vertexCoords[2] << 1.0, 1.0, 0.0;
33     vertexCoords[3] << 0.0, 1.0, 0.0;
34     for (unsigned int i = 0; i < numDomainVertices; i++)
35     {
36         domain.AddVertex(vertexCoords[i]);
37         domain.AddEdge(i, (i + 1) % numDomainVertices);
38     }
39     domain.Initialize();
40
41     /// MESH DOMAIN
42     MeshImport_Triangle meshCreator;
43     meshCreator.SetMaximumCellSize(cellsize);
44     meshCreator.CreateTriangleInput(domain);
45     meshCreator.CreateTriangleOutput(domain);
46     GenericMesh mesh;
47     meshCreator.CreateMesh(domain, mesh);
48     Output::PrintGenericMessage("Triangle ha prodotto una mesh contenente %d
triangoli, %d nodi e %d lati", true, mesh.NumberOfCells(), mesh.NumberOfPoints(),
mesh.NumberOfEdges());
49
50     /// INPUT MESH TO MATLAB SCRIPT FOR VISUALIZATION

```

```

51     ofstream file("plotTriangleMesh.m", ofstream::out);
52     file << "nodesBefore = [";
53     for (unsigned int i = 0; i < mesh.NumberOfPoints(); i++)
54         file << mesh.Point(i)->Coordinates()(0) << "," <<
mesh.Point(i)->Coordinates()(1) << ";" << endl;
55     file << "];" << endl;
56
57     file << "trianglesBefore = [";
58     for (unsigned int i = 0; i < mesh.NumberOfCells(); i++)
59     {
60         file << mesh.Cell(i)->Point(0)->Id() + 1 << "," <<
mesh.Cell(i)->Point(1)->Id() + 1 << "," << mesh.Cell(i)->Point(2)->Id() + 1 << ";"
<< endl;
61     }
62     file << "];" << endl;
63     file << "figure;trimesh(trianglesBefore, nodesBefore(:,1),
nodesBefore(:,2));" << endl;
64
65     /// REFINE MESH
66     TriangleRefiner refiner(mesh);
67     if (percentuale == 100)
68     {
69         refiner.TaglioInQuattro();
70     }
71     else
72     {
73         srand(1);
74         for (unsigned i = 0; i < mesh.NumberOfCells(); i++)
75             if (rand() % 100 < percentuale)
76                 refiner.PrepareTriangle(i);
77         refiner.RefineMesh();
78     }
79     refiner.AggiornaInformazioniPunti();
80
81     /// OUTPUT MESH TO MATLAB SCRIPT FOR VISUALIZATION
82     mesh.CleanInactiveTreeNode();
83     file << "nodesAfter = [";
84     for (unsigned int i = 0; i < mesh.NumberOfPoints(); i++)
85         file << mesh.Point(i)->Coordinates()(0) << "," <<
mesh.Point(i)->Coordinates()(1) << ";" << endl;
86     file << "];" << endl;
87
88     file << "trianglesAfter = [";
89     for (unsigned int i = 0; i < mesh.NumberOfCells(); i++)
90     {
91         file << mesh.Cell(i)->Point(0)->Id() + 1 << "," <<
mesh.Cell(i)->Point(1)->Id() + 1 << "," << mesh.Cell(i)->Point(2)->Id() + 1 << ";"
<< endl;
92     }
93     file << "];" << endl;
94     file << "figure;trimesh(trianglesAfter, nodesAfter(:,1), nodesAfter(:,2));"
<< endl;
95     file << "figure;"

```

```
96         << "hold on;"
97         << "trimesh(trianglesAfter, nodesAfter(:,1),
nodesAfter(:,2), 'color', 'r');"
98         << "trimesh(trianglesBefore, nodesBefore(:,1),
nodesBefore(:,2), 'color', 'b');"
99         << "hold off;" << endl;
100     file.close();
101 }
102
```