Final Project: 2D Integral

NUMA01: Computational Programming with Python

Malin Christersson, Robert Klöfkorn

This assignment has 8 tasks.

The goal of the final project is to experience programming in a group. Proceed in the following way

- discuss the mathematical part of the project in the group first until you fully understand the problem
- divide the problem in subproblems
- discuss how the different parts should be tested
- prepare a max 20 min presentation which should present the mathematical background, the organization of your work, your solutions and maybe alternative attempts, which you decided to reject.

Bring your Laptop or an USB stick for the presentation.

Background

This project describes a very common subtask, which occurs when partial differential equations have to be solved numerically with the finite element method.
We want to compute $I_\Omega = \int_\Omega f(x,y)\,\mathrm{d}x\,\mathrm{d}y$ where $\Omega \subset \mathbb{R}^2$. For this purpose the region is devided into small triangles T_i, called *elements*. You should compute first $I_i = \int_{T_i} f(x,y)\,\mathrm{d}x\,\mathrm{d}y$ and then compute $I_\Omega = \sum_i I_i$.

The integral over a unit triangle T_{unit} with the corner points $(0,0),(1,0),(0,1)$ (see Figure 2) can be approximated by the following formula

$$\int_{T_{\text{unit}}} f(x,y)\,\mathrm{d}x\,\mathrm{d}y \approx \frac{1}{2}\left(\frac{1}{3}f(0,0)+\frac{1}{3}f(0,1)+\frac{1}{3}f(1,0)\right).$$

An arbitrary triangle has to be transformed first to the unit triangle before this integration formula can be applied. How this is done is described in the appendix.
The region Ω is given to you already triangularized as on Figure 1.

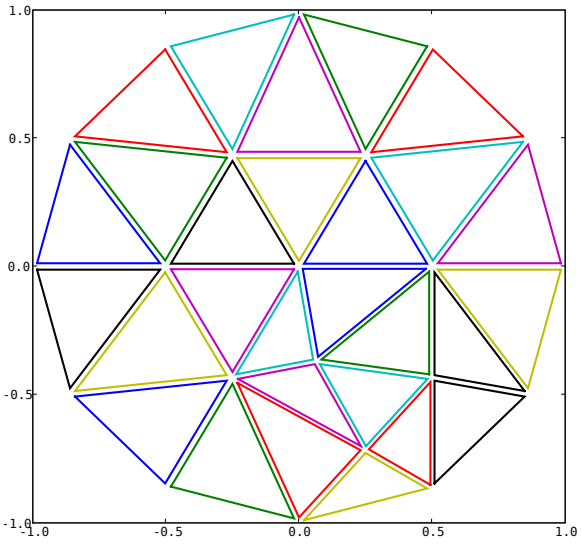

Figure 1: Mesh Example

Task 1

Define a class `Mesh` that is initialised with a *mesh*. A mesh is a tuple of two matrices, one which represents the *coordinates* of the nodes, one which represents the *elements* (the triangles) using the node numbers (numbering the first node with *one*, not zero!).

An example of a mesh is given in the two downloadable files `coord1.txt` (node coordinates) and `elementnode1.txt` (the elements description). Another mesh is described in the files `coord2.txt` and `elementnode2.txt`. These files are also contained in the `meshes.zip` on the project canvas page.

Task 2

Write a method that computes the Jacobian described in the appendix for one given element.

Task 3

Write a method that, given an element, computes the minimum angle between its edges.

Task 4

Write a method that computes a list of all the determinants for all the elements. This method should *raise an exception* if one of the triangles is too thin (that is, if one of the angles is too small).

Make sure that this method is called during the initialisation of the object.

Task 5

Write a method which computes I_Ω for any given function f, using the list computed in the last task. Test the program with different functions f of your choice.

Task 6

Write a method that computes the total area (the sum of all the triangle area) and compare it to the integral of the constant function one.

Task 7

Write a method that draws the mesh as in Figure 1. Each triangle should be visible separately so that one can visually check that no triangle overlaps.

Task 8

Apply your code also to some of the other meshes included in the file meshes.zip on the project canvas page. In particular, calculate the area of the dolphin.[1]

Appendix A: Integrals with coordinate transformation and transforming triangles

Here you get just a brief sketch by some formulas:
Transformation:

$$x = x_1 + (x_2 - x_1)\xi + (x_3 - x_1)\eta$$
$$y = y_1 + (y_2 - y_1)\xi + (y_3 - y_1)\eta$$

[1] Dolphin meshes courtesy of Anders Logg, FEniCS Project.

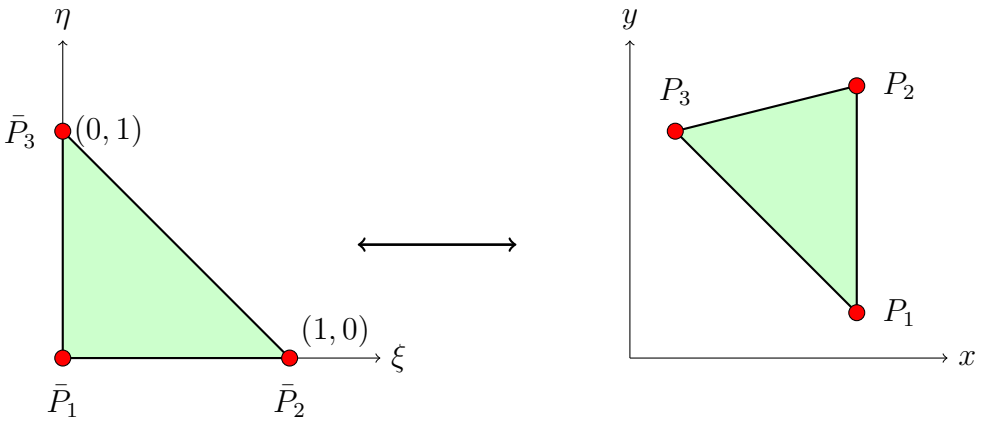

Figure 2: Coordinate Transformation

Note,

$$\int_T f(x,y)\,\mathrm{d}x\,\mathrm{d}y = \int_{T_{\text{unit}}} \bar{f}(\xi,\eta)|J|\,\mathrm{d}\xi\,\mathrm{d}\eta$$

with $\bar{f}(\xi,\eta) = f\Big(x(\xi,\eta), y(\xi,\eta)\Big)$ and J being the determinant of the Jacobian of the transformation, i.e.

$$J := \det \begin{bmatrix} (x_2 - x_1) & (x_3 - x_1) \\ (y_2 - y_1) & (y_3 - y_1) \end{bmatrix}$$