Surface coloring on barycentric subdivisions

František Nesveda and Marion Tommasi

May 13, 2016

Abstract

This work is the report of the project 1 of the Computational Topology course. It is about vertex and face coloring of triangulations and barycentric subdivisions.

1 Introduction

Given a simplicial complex, the barycentric subdivision is obtained by adding a vertex in the barycenter of each simplex. In this report, we'll show how to perform a barycentric subdivision and color its surfaces and vertices with respectively 2 and 3 colors.

2 Algorithms

2.1 Performing barycentric subdivision

When performing the barycentric subdivision of a simplificial complex, we have to subdivide each simplex of the complex separately. We have to subdivide each simplex recursively, first we subdivide its boundary, then we add a new vertex in the barycenter and then connect it with the new simplices obtained by subdividing the boundary. We have to make sure that when subdividing a simplex, if it was already subdivided during previous steps of the algorithm, we put the same vertex in its barycentre when subdividing again.

We implemented this algorithm in the accompanying file `subdivision.py`, which contains a function BARYCENTRICSUBDIVISION, which takes a simplificial complex in the form of a list of simplices of the highest dimension (simplices are represented as a tuple of vertex indices) and returns a subdivided complex in the same form.

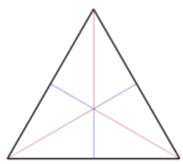

Figure 1: Barycentric subdivision of a triangle.

2.2 3-coloring vertices of a barycentric subdivision of an oriented surface

After performing the subdivision of a surface, there are three conditions that hold:

1. no two original vertices from the original complex are connected, as the edges connecting them were subdivided
2. no two vertices from those added to the barycenter of edges are connected, from the definition of the subdivision
3. no two vertices from those added to the barycenter of triangles are connected, from the definition of the subdivision

We can then color the original vertices with color 0, vertices added to barycenters of edges with color 1, and vertices in barycenters of triangles with color 2, and no two vertices connected with an edge will have the same color.

We implemented this algorithm in the accompanying file `3color.py`, which contains a function COLORSUBDIVISIONVERTICES, which takes a simplificial complex in the form of a list of simplices of the highest dimension (simplices are represented as a tuple of vertex indices) and returns a subdivided complex in the same form, and a dictionary which for each vertex stores the color of the vertex.

2.3 2-coloring of triangles of a barycentric subdivision of an oriented surface

If the surface is oriented, its triangulation is orientable, so all triangles can be oriented in the same way, for example clockwise. The same holds for triangles of the barycentric subdivision, because the subdivision still represents the same surface.

If we orient the triangles of the subdivision clockwise, we can take the triangles of the subdivision, and for each triangle, if the oriented edge from the vertex that was added as a barycenter of an original triangle leads to a vertex in the barycenter of an edge, we color the triangle with color 0, and if it leads to an original vertex, we color the triangle with color 1.

If two triangles share an edge, that edge has an opposite orientation in each of them, so it can never happen that two neighbouring triangles would have the same colour.

We implemented this algorithm in the accompanying file `2color.py`, which contains a function COLORSUBDIVISIONTRIANGLES, which takes a simplificial complex in the form of a list of simplices of the highest dimension (simplices are represented as a tuple of vertex indices) and returns a subdivided complex in the same form, and a dictionary which for each triangle stores the color of the triangle.

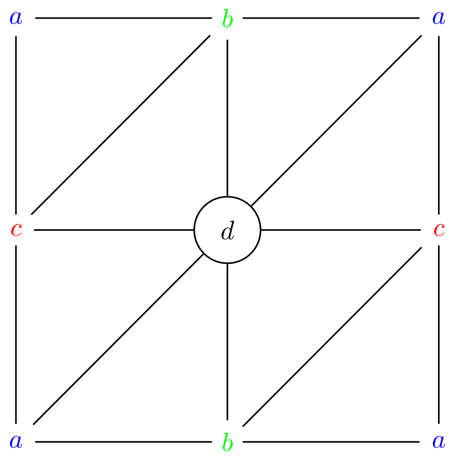

Figure 2: A triangulation not 3-colorable.

3 Counter-examples

3.1 Triangulation whose subdivision cannot be 2-colored

The triangles of a subdivision can be 2-colored only if the triangulation is orientable, so the subdivision of a non-orientable surface cannot have its triangles 2-colored, for example the subdivision of the Möbius strip or the Klein bottle.

The three coloring of vertices should work for a barycentric subdivision of any triangulation, the proof from the previous case should still hold.

3.2 Triangulations whose vertices cannot be 3-colored

Some triangulations of a Klein-Bottle or a Torus, the vertices cannot be 3-colored. In the figure 2, if we choose two colors for a and b, it forces c to a third color. Then, the point d needs a fourth color.

3.3 Triangulations whose faces cannot be 2-colored

V_5 in figure 3 is a triangulation whose faces cannot be 2-colored.

3.4 Triangulation whose vertices cannot be 3-colored and whose faces cannot be 2-colored

Some triangulations of the sphere can neither be 2-colored for their faces and 3-colored for their vertices, as shown in figure 4.

4 Division of work

We worked together on the main idea of the proofs and algorithms. We split the work for the programming. František Nesveda did the barycentric subdivision algorithm and Marion Tommasi the 2- and 3-coloring of triangles and vertices when no information about the barycentric subdivision is given.

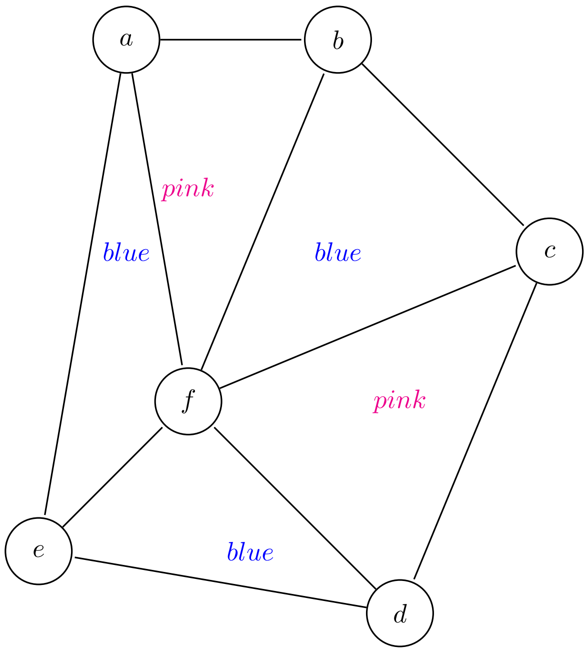

Figure 3: A triangulation whose faces are not 2-colorable. There shouldn't be two adjacent faces with the same color.

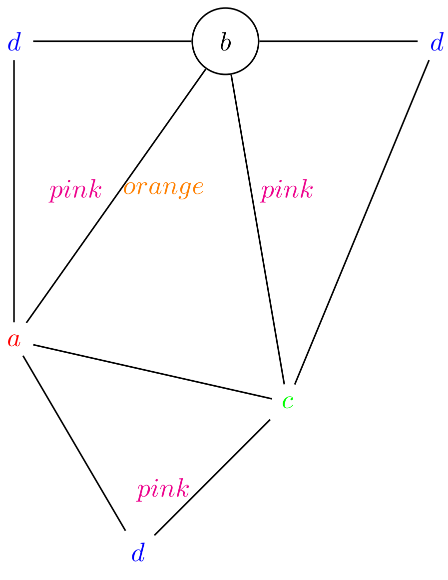

Figure 4: b cannot be colored, as its neighbors already have 3 different colors. The abd, acd and bcd triangles are adjacent, they shouldn't have the same color.

5 Bibliography

We used the book *Computational topology, an introduction* by Herbert Edelsbrunner and John Harer.