Proceedings of the 5th Croatian Combinatorial Days
September 19–20, 2024

ISBN: 978-953-8168-77-2
DOI: 10.5592/CO/CCD.2024.04

Plotting planar and toroidal maps

Mathieu Dutour Sikirić

Abstract

Plotting planar and toroidal maps is a common problem in graph visualization. We present here an algorithmic implementation that has been used in [3]. The implementation is publicly available.

1 Introduction

The graphs considered in this study can have multiple edges, and it is also possible that both endpoints of a face are the same. A graph is said to be of genus g if it can be embedded on a surface of genus g. If $g = 0$, the graph is called *planar* and if $g = 1$, it is called *toroidal*. A *face* in such an embedding is bounded by a sequence of edges and is homeomorphic to a disk. This embedding gives positions of vertices, edges, and faces.

The combinatorial structure of the vertices, edges, and faces can be studied without having a specific embedding, as the individual positions of the vertices do not matter very much. However, if one has built a graph with edges, vertices, and faces, we may want to find an embedding, as this can be helpful for scientific purposes and also for visualization.

The problem of graph embedding has been considered from various viewpoints. In [9] a method to use the eigenvectors of the largest eigenvalues of the adjacency matrix was used. The method works well for plane graphs but suffers from one key problem: the inner faces tend to be very small and not visible. Also, the method works only for planar graphs.

The program CaGe (see [1, 2]) uses an iterative process in order to get an embedding. A priori, it works only for planar graphs. Another class of methods is to minimize the functional F

$$F = \sum_{e=(i,j)\in E(G)} f(\|x_i - x_j\|)$$

(Mathieu Dutour Sikirić) MSM programming, Karlovačka Cesta 28B, 10450 Jastrebarsko, Klinča Sela, Croatia, mathieu.dutour@gmail.com

A more powerful and conceptual technique is to use primal-dual circle packing for obtaining a drawing (see [6]). The technique works for any genus and graphs that are 3-connected.

2 Combinatorial representation of maps

A general 2-dimensional map $\mathcal{M}$ can be represented by vertices, edges, and faces. The incidence relation between those objects is best encoded by the notion of flags. A flag is a triple (v, e, F) with v a vertex, e an edge, F a face and $v \in e \subset F$. Denote by $Flag(\mathcal{M})$ the set of all flags of $\mathcal{M}$. From the flags, we can define the flag operators σ_0, σ_1, and σ_2. For a flag $f = (v, e, F)$, the image $\sigma_0(f)$ is defined as the flag that is identical to f except at the vertex, $\sigma_1(f)$ the flag that is identical to f except at the edge and $\sigma_2(f)$ the flag that is identical to f except at the face. The flag operators are uniquely determined by those constraints. They satisfy the relation $\sigma_i^2 = Id$ and $\sigma_0\sigma_2 = \sigma_2\sigma_0$.

The flag operators are permutations on the set of flags, and they can be used for representing the maps efficiently. For example, the vertices correspond to the orbits of the group generated by $\{\sigma_1, \sigma_2\}$ on $Flag(\mathcal{M})$. The formalism of flags works well even for the non-orientable maps such as the projective plane or the Klein bottle.

However, for the purpose of this work, the maps that we consider are oriented. So instead of something so general, we use the directed edge formalism. At every vertex and every edge, we associate a direction. We define two operators:

1. The next operator n that sends a directed edge to the next one in the positive (counterclockwise) orientation.

2. The inverse operator i that reverses the direction of the directed edge.

See Figure 1 for an example of such an action on a directed edge. Since a planar graph has a finite number of directed edges, those operators can be represented as two permutations. The vertices, edges, and faces correspond to the orbits of the group generated by $\{n\}$, $\{i\}$, and $\{n \circ i\}$.

3 The primal-dual circle method

It is a remarkable fact that in dimension 2, we have a meta-theorem that combinatorics=geometry. That is, combinatorics alone suffices to encode an object. One example of such a theorem is Steinitz theorem ([8] and [10, Chapter 4]) that states that a graph is the skeleton of a 3-dimensional

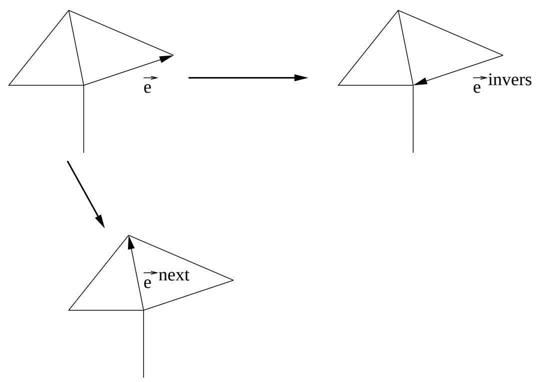


Figure 1: The action of the next and inverse operator on a directed edge

polytope if and only if it is planar and 3-connected. The primal-dual circle packings are an effective technique that allows to represent graphs given by their combinatorial information (see Chapter 5 of [5]).

The idea is to put circles $C_{vert}(v)$ in the vertex center and face center $C_{face}(F)$ so that:

- The interiors of the circles $C_{vert}(v)$ are all distinct. Two circles $C_{vert}(v)$ and $C_{vert}(v')$ intersect in a point if and only if the vertices v and v' are adjacent. Denote $P_{vert}(v, v')$ the intersection point.

- The interiors of the circles $C_{face}(F)$ are all distinct. Two circles $C_{face}(F)$ and $C_{face}(F')$ intersect in a point if and only if the faces F and F' share an edge. Denote $P_{face}(F, F')$ the intersection point.

- If v and v' are adjacent vertices with an edge e shared between faces F and F' then we have $P_{vert}(v, v') = F_{face}(F, F')$

Those properties are represented in Figure 2.

According to [6], if a graph G of genus g is 3-connected and simply connected then there exists an embedding of the universal cover of G into the sphere (if $g = 0$), the plane (if $g = 1$) or the hyperbolic plane (if $g > 1$). See Figure 3 for one such example.

Due to the invariance by duality of the primal-dual circle packing, it makes sense to consider the medial graph $Med(G)$ for a graph of genus g. It is the graph whose vertices are the vertices and faces of G, and two vertices of $Med(G)$ are adjacent if one is a vertex u of G, and another is a face F of G and $u \in F$. The graph $Med(G)$ is also of genus g and its faces are of size 4. This is the method used in this work.

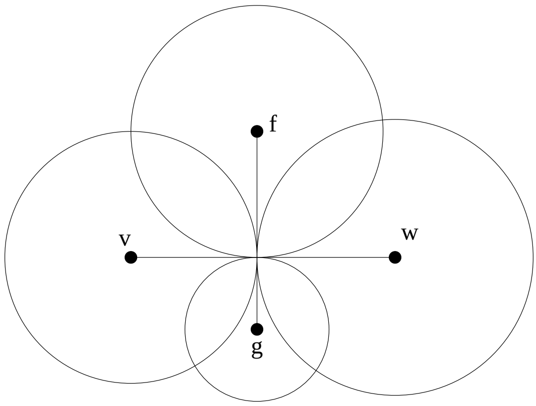


Figure 2: The local picture of a primal-dual circle representation

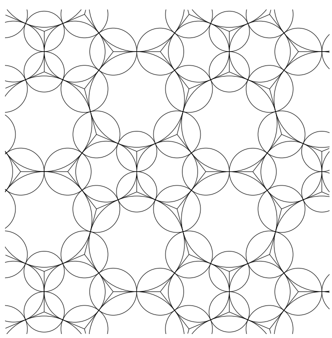


Figure 3: The edges, circle, and face circles of a primal-dual representation

4 Numerical techniques for the planar case

The equation system that needs to be resolved for plotting the graph represents the fact that the sum of the angles is 2π at each vertex. The angles are computed from the radii of the circles $C_{vert}(u)$ and $C_{face}(F)$

In the case of genus $g = 1$, the angle equations are simpler to write down since we do not have to deal with spherical or hyperbolic trigonometry. We define for each vertex v of $Med(G)$ the angle sum:

$$\phi_v = \sum_{uv \in E(Med(G))} \arctan\left(\frac{r_u}{r_v}\right).$$

The equations to resolve are thus

$$\phi_v = \pi \text{ for } v \in V(G).$$

This is a system of non-linear equations, and we have the equation

$$\sum_{v \in V(Med(G))} \phi_v - \pi = 0. \tag{4.1}$$

This conservation equation renders the system underdetermined. Thus, if the collection (r_u) is a solution, then (λr_u) is also a solution.

[6] has given an algorithm for computing the primal dual circle packings radii. It consists of computing the defect at every node and increasing/decreasing the radius value according to $\phi_v > \pi$ or $\phi_v < \pi$. It is a geometric method that is quite efficient, but in some cases it is very slow.

Newton's method cannot be applied right away as the problem is underdetermined. We address this by restricting ourselves to the vector space of directions that do not change the sum of the radius of all the circles. Another more essential problem is that when we apply the Newton method, we can obtain a negative value for some radii. The technique that we use is instead to rescale the increment by a factor c in the following way:

$$x^{(n+1)} = x^{(n)} - c\frac{f(x^{(n)})}{f'(x^{(n)})} \text{ with } 0 < c \leq 1$$

In the terminology of [7] it is Newton method with a line search. We start with a factor $c = 1$ and then decrease it by a factor 1.2 until:

1. All the radii are positive.
2. The sum of the absolute error $\sum_v (\phi_v - \pi)^2$ has decreased.

This turns out to work very well.

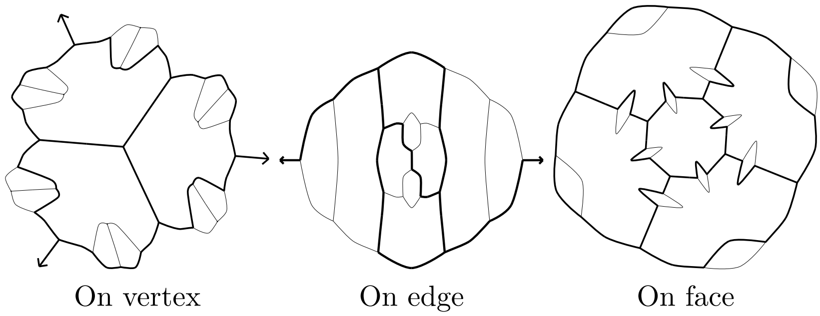


Figure 4: The k-fold axis position can be on a vertex, an edge, or a face

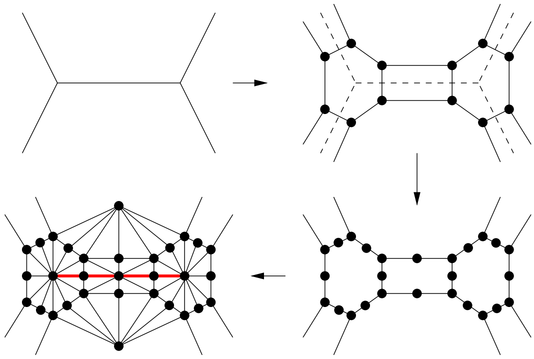


Figure 5: The scheme used for transform a graph into another that is easier to draw.

5 Graphical niceties

Obtaining a graphical representation of a map can be helpful. However, we sometimes want something nicer to look at. For example, if a graph has a 3-fold axis of symmetry, then we may want to have this axis directly visible on the map. If the axis is passing through a face, then we just need to select the appropriate face. If, however, the axis passes through two vertices, then we have a problem. We need to represent it, but we must choose the correct axis. It is also possible for a 2-fold axis to pass through two edges. Some examples are shown in Figure 4.

The primal-dual technique requires 3-connectivity and will not work with 2-gons and 1-gons. The technique is to refine it. First, for a map M we replace it with the order complex map $Ord(M) = Trunc(Med(M))$. Then we insert a vertex on each edge. Finally, we put a vertex on each face and connect it to all incident vertices. The resulting triangulation is 3-connected.

6 CaGe process and the integrated algorithm

The algorithm of [1] is used for the drawing of planar graphs. It works in the following way:

- Select an external face F and puts the vertices in a circle.
- Put a point x_F in the center of each face that is not external. We have a collection of points $(x_v) \cup (x_F)$.
- For each point x that is not on the outer circle and that is contained in the triangles (x, x^i, x^{i+1}), we have

$$x = \frac{1}{\sum_{i=1}^{m} A_T^2(x, x^i, x^{i+1})} \sum_{i=1}^{m} A_T^2(x, x^i, x^{i+1}) \frac{x + x^i + x^{i+1}}{3}$$

  with $A_T(x, y, z)$ the area of the triangle of vertices x, y and z.
- The equation can be solved by fixed-point iterations.

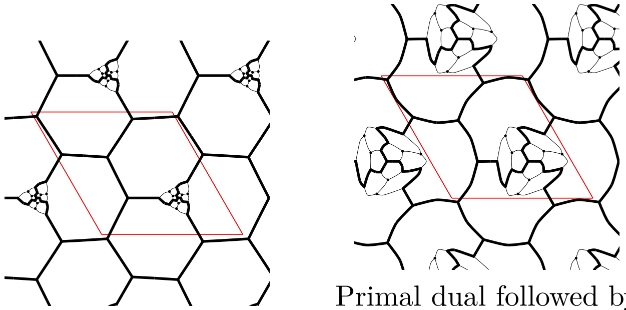
For a planar graph, we apply the following construction:

- Apply the truncation scheme of Section 5.
- Depending on the choice of the external face (vertex, edge, or face), select the corresponding external face.
- Apply the CaGe process to find an embedding.
- From the positions of the various points, build the path that corresponds to an edge of the graph.

The CaGe process is an averaging process where we average the position of the neighbor in order to update the position of a point. This same process can be applied in the toroidal case. This can yield a clear improvement to the obtained graph as shown in Figure 6.

7 The effective design of the software and how to use it

It is important to have Open Source code so that other users can benefit from the created software. But, that is not quite sufficient if the Open source software is very difficult to use. Thus, we have tried to make a Python-based solution which should be convenient as Python has emerged



Primal dual

Primal dual followed by CaGe process

Figure 6: The graph of a toroidal map obtained by the primal-dual processed followed by the one with the CaGe process applied afterward

as the equivalent of English in Computer Science: not a perfect language, but everybody is using it.

The computation of the coordinates is done with a code written in **C++**. The matrix algebra is done by using the Eigen library ([4]). The input file of the program follows the Fortran Namelist format, which is a simple yet reasonable to use format.

The data are then exported to a `SVG` file. The `SVG` file format is adequate for this purpose since it is a symbolic text file format that can be used in Web browsers. Also, `SVG` files can be edited by **inkscape**. This allows users to easily edit according to their wishes.

The Python code is accessed via

```
pip3 install https://github.com/MathieuDutSik/PyPlot_orientedmap
```

Alas, not everything is so simple with the program usage. The difficulty is in creating the input. As convenient as the format with directed edges is, it has a distinct computational aspect to it.

References

[1] G. Brinkmann, O. Delgado Friedrichs, A. Dress, and T. Harmuth. Cage-a virtual environment for studying some special classes of large molecules. *MATCH Communications in Mathematical and in Computer Chemistry*, (36):233–237, 1997.

[2] G. Brinkmann, O. D. Friedrichs, S. Lisken, A. Peeters, and N. Van Cleemput. Cage—a virtual environment for studying some special classes of plane graphs—an update. *MATCH Commun. Math. Comput. Chem*, 63(3):533–552, 2010.

[3] M.-M. Deza and M. D. Sikirić. Lego-like spheres and tori. *Journal of Mathematical Chemistry*, 55(3):752–798, 2017.

[4] G. Guennebaud, B. Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.

[5] L. Lovász. *Graphs and geometry*, volume 65. American Mathematical Soc., 2019.

[6] B. Mohar. Circle packings of maps in polynomial time. *European Journal of Combinatorics*, 18(7):785–805, 1997.

[7] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, New York, 2nd edition, 2006.

[8] E. Steinitz. Polyeder und raumeinteilungen. In *Enzyklopädie der mathematischen Wissenschaften mit Einschluss ihrer Anwendungen*, volume 3, chapter 12, pages 1–139. Teubner, Leipzig, 1922.

[9] W. T. Tutte. How to draw a graph. *Proceedings of the London Mathematical Society*, 3(1):743–767, 1963.

[10] G. M. Ziegler. *Lectures on Polytopes*, volume 152 of *Graduate Texts in Mathematics*. Springer, New York, 1995.