Université Gustave Eiffel, France
loic.dubois@univ-eiffel.fr
loic-dubois.github.io
https://orcid.org/0000-0002-0366-4959
https://scholar.google.com/citations?user=7aW3tOAAAAAJ&hl=fr

Loïc Dubois

Research statement and curriculum vitae

My thesis will end around June 2025, I am looking for a post-doctoral position starting in September 2025. My three selected papers are [A1](SODA'24), [A7](SODA'25), [A3](preprint). They are presented below along with my other works.

I design and analyze efficient algorithms for geometric (and topological) problems. This line of research, computational geometry [B7, B12, B14] [1], emerged in the 70s in response to the needs of computer graphics, computer-aided design, robotics, and geographic information systems. Populated by traditional geometric notions (line arrangements, manifolds, Voronoi diagrams), the field has equipped itself with unique algorithms and paradigms (convex hull algorithms, line sweeping, epsilon nets), and has developped interactions with other areas of computer science. My focus is on the design of polynomial time algorithms for representations of surfaces, and for graphs drawn on surfaces. That involves discretizing geometric notions and providing the appropriate data structures.

Previous work

Untangling Graphs on Surfaces. A drawing of a graph G on a (topological) surface S maps every vertex of G to a point of S, and every edge of G to a path in S between the images of its end-vertices. In a series of papers [A7, A1, A4] with Éric Colin de Verdière and Vincent Despré, we provide algorithms to remove crossings and overlaps from such a drawing by sliding it along S, or to correctly assert that this is not possible, with refinements such as allowing flexibility in the output drawing, or minimizing the number of crossings when the drawing cannot be untangled and G is a collection of cycles. At the root of our results lies a powerful tool that we introduced, *reducing triangulations*, a discrete analog of hyperbolic surfaces.

Tessellations. A surface can be obtained without reference to $\mathbb{R}^3$ from a *tessellation*, a collection of *plane* polygons with matched edges. Not all tessellations are suitable for computation. Prominently, shortest path algorithms are affected by the maximum number of times a shortest path visits a face, which is unbounded (in stark constrast with polyhedral meshes). In [A3] I provide an efficient algorithm to compute the Delaunay tessellation from any other tessellation of the surface. This implies algorithms to pre-process a tessellation before computing shortest paths on its surface, and to determine if two tessellations represent the same surface.

Delaunay flip algorithm. A Delaunay flip algorithm greedily flips the edges of an input triangulation T until it reaches a Delaunay triangulation [B7, Chapter 9]. In the Euclidean plane $O(n^2)$ flips occur, where n is the number of vertices of T; The bound is tight. The algorithm extends to triangulations of piecewise-flat and hyperbolic surfaces [B8, B9], where the number of flips is finite but not bounded by any function of n, and the exact asymptotics are unknown. On flat tori I prove in [A6] that $O(n^2L)$ flips occur, and that this is tight, where L is the (normalized) maximum edge length of T. On closed hyperbolic surfaces, I implemented the algorithm with Vincent Despré and Monique Teillaud, for integration as a package in the next release of the Computational Geometry Algorithms Library (CGAL) [A2].

Research directions

Two common themes in my work are computations operating on graphs topologically embedded on surfaces, and computations operating on metric surfaces.

[1] References in which I am author or co-author are labeled “A”, the others are labeled “B”.

Algorithms for graphs topologically embedded on surfaces. I plan to develop efficient algorithms for graphs topologically embedded on surfaces. This is a very active trend in computer science, with recent developments such as algorithms computing the eccentricities of planar graphs in sub-quadratic time (which is impossible on general graphs, assuming the strong exponential time hypothesis) [B10, B2], or approximating planar metrics by simpler structures with low distortion [B3, B4]. I would start with the following typical distance realization problem. Consider a topological surface S with boundary (possibly of high genus), a finite set X of points on the boundary of S, and a matrix $(d_{x,y})_{x,y\in X}$ of numbers. Does there exist a graph G embedded on S, whose vertex set contains X, such that for every $x, y \in X$ the shortest path distance between x and y in G is equal to $d_{x,y}$? If so, can you construct G, possibly of "small size"? When the surface S is the disk D, the matrices that can be realized this way are characterized by a simple Monge condition. Moreover, for every n there exists a graph G embedded on D with n vertices on ∂D such that every Monge matrix of size n can be realized by varying the weigths on the edges of G [B5]. Such results have been very recently extended to *directed* graphs in a SODA 2025 paper [B6], again still when S is the disk. The cases where S has higher genus, or several boundary components, seem open, even in the undirected case.

Algorithms for metric surfaces. To represent a topological surface S (of arbitrarily high genus, possibly with boundary...) on a computer, one cuts S into "manageable" pieces along a graph, then stores the pieces independently, and remembers how their boundary edges are matched in S. Any data attached to S is then encoded within the pieces. This simple and general model has a variety of specializations, depending on the topological type of the pieces (topological triangles, topological spheres with three holes, a single piece that is topological polygon...), and on the data attached to them. A typical example of data is a distance function, or *metric* (flat, piecewise-flat, hyperbolic...). I plan to study algorithms that operate on those representations of metric surfaces. In particular, very little is known on computing a representation of a metric surface from another representation of the same metric surface. The situation is most critical on hyperbolic surfaces where there is no algorithm to transform a hyperbolic triangulation into a polygonal schema, a pants decomposition, or a thin-thick decomposition [B1]. Within a given model, converting an instance to another is also challenging. In that regard, a natural class of algorithms perform *local* moves *greedily.* We already discussed the Delaunay flip algorithm. Similarly [B13] flip edges, but with a different heuristic aimed at finding geodesics. Those algorithms are simple to implement, and easy to transfer from a model of surface to another, but they lack complexity analysis: I plan to study them. In that matter, it is an interesting question of whether technics similar to that of [A3, A6] could apply. Also, considering inputs that are *happy* in the sense of [B11] is a strong but realistic assumption (verified by all polyhedral meshes), whose implications are vastly open.

While excited by the interface of algorithms and geometry, I am also interested in algorithm design in general, and I am open to new kinds of problems and collaborations.

Work where I am author or co-author

[A1] Éric Colin de Verdière, Vincent Despré, and Loïc Dubois. Untangling Graphs on Surfaces. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 4909–4941. SIAM, 2024. doi:10.1137/1.9781611977912.175.

[A2] Vincent Despré, Loïc Dubois, and Monique Teillaud. Hyperbolic Surface Triangulations. URL: https://cgal.github.io/8259/v0/Hyperbolic_surface_triangulation_2/index.html#Chapter_Hyperbolic_Surface_Triangulations.

[A3] Loïc Dubois. Computing Happy Triangulations of PL surfaces (preprint).

[A4] Loïc Dubois. Making Multicurves Cross Minimally on Surfaces. In *32nd Annual European Symposium on Algorithms (ESA 2024)*, volume 308, pages 50:1–50:15, 2024. doi:10.4230/LIPIcs.ESA.2024.50.

[A5] Loïc Dubois, Gwenaël Joret, Guillem Perarnau, Marcin Pilipczuk, and François Pitois. Two lower bounds for p-centered colorings. *Discrete Mathematics & Theoretical Computer Science*, 22(Graph Theory), 2020. doi:10.23638/DMTCS-22-4-9.

[A6] Loïc Dubois. A Bound for Delaunay Flip Algorithms on Flat Tori. *Computing in Geometry and Topology*, 2(2):6:1–6:13, 2023. doi:10.57717/cgt.v2i2.27.

[A7] Éric Colin de Verdière, Vincent Despré, and Loïc Dubois. A discrete analog of tutte's barycentric embeddings on surfaces. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 5114–5146. SIAM, 2025.

References

[B1] Peter Buser. *Geometry and spectra of compact Riemann surfaces.* Springer Science & Business Media, 2010.

[B2] Sergio Cabello. Subquadratic algorithms for the diameter and the sum of pairwise distances in planar graphs. *ACM Transactions on Algorithms (TALG)*, 15(2):1–38, 2018.

[B3] Hsien-Chih Chang, Jonathan Conroy, Hung Le, Lazar Milenkovic, Shay Solomon, and Cuong Than. Covering planar metrics (and beyond): $o(1)$ trees suffice. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 2231–2261. IEEE, 2023.

[B4] Hsien-Chih Chang, Robert Krauthgamer, and Zihan Tan. Near-linear ϵ-emulators for planar graphs. *arXiv preprint arXiv:2206.10681*, 2022.

[B5] Hsien-Chih Chang, Tim Ophelders, J Mark Keil, and Debajyoti Mondal. Planar emulators for monge matrices. In *CCCG*, pages 141–147, 2020.

[B6] Yu Chen and Zihan Tan. Path and intersections: Characterization of quasi-metrics in directed okamura-seymour instances. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2467–2490. SIAM, 2025.

[B7] Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf. *Computational Geometry*, pages 147–218. Springer Berlin Heidelberg, Berlin, Heidelberg, 1997.

[B8] Vincent Despré, Jean-Marc Schlenker, and Monique Teillaud. Flipping geometric triangulations on hyperbolic surfaces. In *SoCG 2020-36th International Symposium on Computational Geometry*, 2020.

[B9] M. Fisher, B. Springborn, P. Schröder, and A. I. Bobenko. An algorithm for the construction of intrinsic delaunay triangulations with applications to digital geometry processing. *Computing*, 81(2):199–213, 2007.

[B10] Paweł Gawrychowski, Haim Kaplan, Shay Mozes, Micha Sharir, and Oren Weimann. Voronoi diagrams on planar graphs, and computing the diameter in deterministic $\tilde{o}(n^{5/3})$ time. *SIAM Journal on Computing*, 50(2):509–554, 2021.

[B11] Maarten Löffler, Tim Ophelders, Rodrigo I. Silveira, and Frank Staals. Shortest Paths in Portalgons. In *39th International Symposium on Computational Geometry (SoCG 2023)*, volume 258, pages 48:1–48:16, 2023.

[B12] Franco P Preparata and Michael I Shamos. *Computational Geometry: An Introduction.* Springer Science & Business Media, 2012.

[B13] Nicholas Sharp and Keenan Crane. You can find geodesic paths in triangle meshes by just flipping edges. *ACM Trans. Graph.*, 39(6), November 2020.

[B14] Csaba D Toth, Joseph O'Rourke, and Jacob E Goodman. *Handbook of discrete and computational geometry.* CRC press, 2017.

Education

I was a paid civil servant at École Normale Supérieure de Lyon.

2022-now	**PhD candidate** at Université Gustave Eiffel, advised by Éric Colin de Verdière and Vincent Despré.
2021-2022	**Diploma** of École Normale Supérieure de Lyon. **Internship** at Université Gustave Eiffel, advised by Éric Colin de Verdière and Vincent Despré.

	Internship at Technische Universität Berlin, advised by Stefan Felsner.
2019-2021	**Master in Computer Science** of École Normale Supérieure de Lyon.
	Internship at Inria Nancy, advised by Vincent Despré and Monique Teillaud.
	Internship, remote due to covid, advised by Guillem Perarnau.
2018-2019	**Bachelor in Computer Science** of École Normale Supérieure de Lyon.
	Bachelor in Mathematics for Engineering of Université Claude Bernard.

Teaching

I have been in charge of 192 hours of exercise sessions over the past 3 years, for the following courses:

2022-2025	OpenGL (64h), Assembler nasm (36h)
2024-2025	SQL (24h), Mathematics for Computer Science (12h)
2022-2024	Algorithms and Programming in Python (56h)