OpenMathMap 2 – The Shape of Mathematics

Pavel Kostarev[1], Michael Kohlhase[1], and Dennis Müller[1]

Computer Science, FAU Erlangen Nürnberg, Germany

Abstract. Maps provide an easy-to-understand way to communicate proximity and connectedness knowledge at scale. This was used by the OpenMathMap project (`https://map.mathweb.org`) to give a global overview of the field of mathematics based on co-classification data of publication in Zentralblatt Math. OpenMathMap uses OpenStreetMap to render and interact with a map of a generated land-mass with surrounding islands. It shows the first two levels of the MSC classification as countries and provinces and adds capitals and cities in strategic locations.

Keywords: Dimensionality Reduction · OpenStreetMap (OSM) · Map-like visualization · Voronoi tessellations

1 Introduction

The enormous growth of information in science continues to pose new challenges, especially when large amounts of data need to be visualized [FG19]. In this paper, an example is provided to illustrate this tendency by visualising fields of mathematics that originate from scientific papers published in zbMATH, an open peer-review service for pure and applied mathematics [zbM24]. The initial prototype (OMM1) of OpenMathMap (OMM) was born about ten years ago, trying to solve the problem of visualizing different classes of mathematics in the form of a geographical map [Ace14; Doe13; DK13; DKL13]. The aforementioned work was inspired by the work of Dave Rusin and his "Math Atlas", adopting the concept of Mathematics Subject Classification (MSC) [Rus; Rus24] as a source of geometric relations.

The underlying zbMATH dataset (articles in pure and applied mathematics) for the OMM comes in an XML format and provides a MSC classification code and metadata for every article. The MSC classification uses a hierarchical label set where the first two characters, e.g. 68 (class of Computer Science), represent the first-level classes [n = 63]. The first three characters, e.g., 68T (field of Artificial Intelligence), stand for the second-level classes [n = 528], and finally, five characters represent the third-level classes, i.e. 68T30 (class of Knowledge representation) [n = 5607].

The map generation methodology of OMM1 was influenced by the concept of thematic software visualisation as pioneered by Kuhn and Niestrasz [Kuh+10] and could be described by the following steps: *i*) Computation of a similarity matrix for geometrical representation of classes to represent proximity between

MSC classes; *ii*) Projection of a similarity matrix onto a two-dimensional space (via multidimensional scale); *iii*) Assigning each MSC class an extra dimension for emphasizing the size of each class; *iv*) Encoding the coordinates and borders into the OSM framework [Ace14; Doe13; DK13; DKL13]. OpenMathMap interprets the rate of co-classification as a proximity measure between MSC classes to be presented in the generated map. This yielded a sophisticated and effective big data visualization, which facilitated the analysis of interdependencies among fields of mathematics.

But OMM1 had some weaknesses, (some of) which we overcome with OpenMathMap 2, the system we present in this paper. The OMM1 map was based on a single OSM file rendered by a desktop map visualization software without exploiting the possibilities of a database, which limited the scalability of the first version of the map. The software, used for rendering of the map, called "Maperitive" was able to return PNG 256x256 image tiles for different zoom levels of the map from the XML data [DKL13; Map13]. Moreover, the large similarity matrices posed a huge computational problem and forced the authors to perform local computations for every top-level class independently [DKL13], therefore the multidimensional scaling for projection of the entire matrix for second and third level classes onto the map was not possible to calculate within a reasonable computational time. In addition, the first version of the map provided limited interactive services, such as zooming and panning of the map. From the user's perspective, search queries were possible only by a class name, which could possibly be extended to other features of every class, such as author or keyword. The interactivity of the map was confined to the highlighting of class names by clicking on the map (by adding markers to the map) [DKL13]. The authors did not exploit the other interactive features available in the library, such as map overlays and polylines to represent connections between classes when clicking on a class. Finally, the OpenStreetMap (OSM) framework possibilities have not been used fully, but only limited to a few, such as borders, areas, and cities, for example, no rivers or mountains.

In the next section we will give an overview of the OpenMathMap 2 system and in section 3 presents the user view of OMM2. Section 4 concludes the paper.

2 Map Generation in OpenMathMap 2

The map generation process consists of four steps: 1) Preprocessing of abstract data; 2) Spatialization; 3) Imitation; 4) Map-like visualization [HHS20].

1) Preprocessing of abstract data: As a first step, we represent the zbMath XML dataset (4988284 documents and metadata associated with every document [zbM24]) as subject-predicate-object triplets, creating an RDF model [Wor24]. For this, we extracted four relevant features from the metadata of each document: i) MSC classification code; ii) authors; iii) keywords; iv) publication year. For instance, a publication with the identifier 123456, published in 2004, can be expressed as: *123456, publication_year, 2004*. This format facilitates the storage of the dataset as a graph database.

2) Spatialization: To project the classes as cities with x and y coordinates onto the two-dimensional map plane, where the distances between the points would have a meaning, i.e., the closer the points, the more similar they are, while the further away the points are, the less similar they are, we calculated the distances between all the classes. To achieve this, we computed the similarity matrices for every level of the dataset using a Jaccard similarity index metric [OE23]. Because these matrices contain all pairs of classes in the dataset, their dimensions are large and to present them on a 2D scale, we need to employ the methods of Multidimensional Scaling as non-metric Multidimensional Scaling (nMDS) [sci24] and UMAP [**umap2025**], which can preserve the distances between points and neighborhoods of points while returning an embedding with two dimensional coordinates for every class.

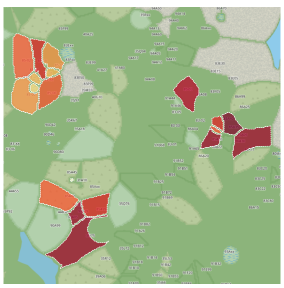
3) Imitation: Once we generated the points on the map, we assign a region around every point: For every class, we apply radial basis function (RBF) kernel function around every point to assign an area proportional to the number of class articles around every point. As some of the areas intersect, we combine the overlapping areas and combine them into islands. Consequently, on every island, we perform a weighted Voronoi tessellation to subdivide them into regions again and create natural-like borders, which would imitate the real geographical maps.

4) Map-like visualization: Finally, we extract the points and borders generated in the previous step and encode them into OSM framework and render them inside a web application. For more details, please refer [Kos24].

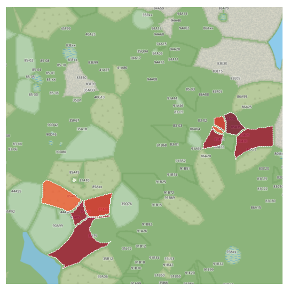
Fig. 1: Final view of the web application with an interactive color-coded layout, where lighter shades of brown indicate a smaller number of publications, while darker shades of brown represent classes with a large number of publications

3 OpenMathMap 2 – Map/Frontend

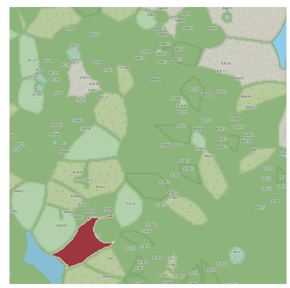
The final map is available online as a web application built on React.js for the front end and Express.js for the back end at `https://mathmap.kwarc.info/`. The integration of the OSM map-tiling server with the extended interactive features has been instrumental in facilitating an interactive experience for the user, enabling highlighting, clicking, and performing search queries by name or class classification code, author name, or even both at the same time for even better comparison of different areas of mathematics. Furthermore, the web service provides timeline statistics for every class and custom API services for retrieving latitude and longitude values of a city given a class name.



(a) Class filtering for query '85' (only first-level classes starting with 85 are highlighted)



(b) Class filtering for query '85A' (only second-level classes starting with 85A are highlighted)

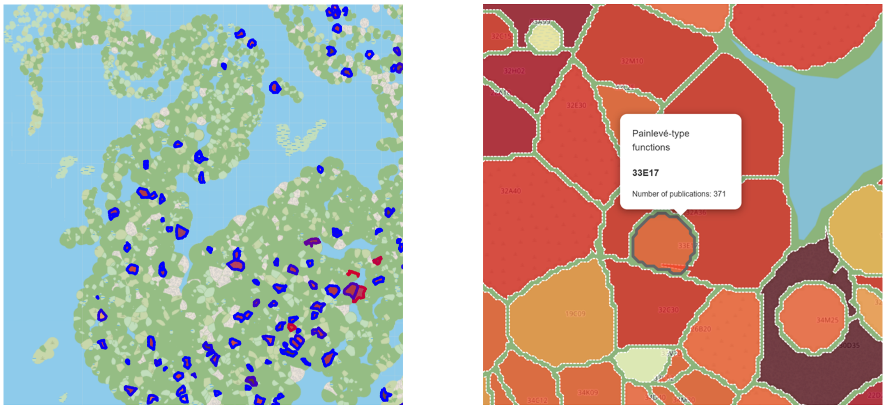


(c) Class filtering for query '85A25' (only third-level class 85A25 is highlighted)

Fig. 2: Interactive class filtering by search query

4 Conclusion

OpenMathMap2 achieved all intended objectives and improved the previous version: Firstly, with the new version we can now can display 6760 publication classes coming from all three hierarchical levels, surpassing the previous generation, where only 591 classes were visualized. This delivers an analysis that is deeper and meaningful between fields of mathematics for researchers. Secondly, the interactivity of the map is enhanced with detailed search and other mouse-overlaying options. Thirdly, the scalability of the map has been achieved by employing a database, instead of storing the data in a single file. For this, a special map data rendering library has been used (see Mapnik library [Map25]). Finally, the incorporation of geographical features (e.g., rivers, mountains, and land use types) into the map, in conjunction with the web application's advanced functionality, facilitates the consumption of information through a map-like visualisation, thereby enhancing the user experience.



(a) A search by author shows the boundaries of each publication class linearly interpolated between blue and red. The least publications are shown in blue, and the opposite is shown in red.

(b) A banner displaying the field name, classification code and number of publications is visible in an interactive overlay when the publication class is hovered over.

Fig. 3: Filtering classes by author and hovering over a class with the mouse.

In subsequent studies, the development of the map may be advanced through the execution of trials with end users, including mathematicians and researchers. The implementation of these tests would facilitate the identification of novel map features and functionalities, thereby enhancing the map's alignment with the user's needs.

References

[Ace14] C. Acevedo. *OpenMathMap: Interaction*. Bachelor's thesis, Computer Science, Jacobs University Bremen. Accessed: 2025-04-14. 2014. URL: `http://map.mathweb.org/media/openmathmap-interaction.pdf`.

[DK13] J. W. Dörrie and M. Kohlhase. "OpenMathMap: Accessing Math via Interactive Maps". In: *CICM 2013 Work in Progress Proceedings*. Workshop paper, CEUR-WS Vol. 1010. 2013. URL: `http://ceur-ws.org/Vol-1010/paper-12.pdf`.

[DKL13] J. W. Dörrie, M. Kohlhase, and L. Linsen. "OpenMathMap: Accessing Math via Interactive Maps". In: *Contemporary Issues in Mathematical Publishing, JMM San Diego Special Session*. Ed. by K. Kaiser, S. Krantz, and B. Wegner. EMIS, 2013, pp. 81–98. URL: `https://kwarc.info/kohlhase/papers/cimp13.pdf`.

[Doe13] J. W. Doerrie. *OpenMathMap: Accessing Math via Interactive Maps*. Bachelor's thesis, Jacobs University Bremen. Accessed: 2025-04-14.

2013. URL: https://kwarc.info/people/archive/pubs/bsc-2013/jdoerrie.pdf.

[FG19] M. Fire and C. Guestrin. "Over-optimization of academic publishing metrics: Observing Goodhart's Law in action". In: *GigaScience* 8 (June 2019). DOI: 10.1093/gigascience/giz053.

[HHS20] M. Hogräfer, M. Heitzler, and H.-J. Schulz. "The State of the Art in Map-Like Visualization". In: *Computer Graphics Forum* 39 (July 2020), pp. 647–674. DOI: 10.1111/cgf.14031.

[Kos24] P. Kostarev. "The Next Generation of OpenMathMap". Master's Thesis. Erlangen, Germany: Friedrich-Alexander-Universität Erlangen-Nürnberg, 2024.

[Kuh+10] A. Kuhn, D. Erni, P. Loretan, and O. Nierstrasz. "Software Cartography: Thematic software visualization with consistent layout". In: *Journal of Software Maintenance* 22 (Apr. 2010), pp. 191–210. DOI: 10.1002/smr.414.

[Map13] Maperitive. *Maperitive*. 2013. URL: http://maperitive.net/ (visited on 05/05/2013).

[Map25] Mapnik Developers. *Mapnik: Toolkit for Developing Mapping Applications*. https://mapnik.org/. Accessed: 2025-05-10. 2025.

[OE23] D. Ogwok and E. M. Ehlers. "Jaccard Index in Ensemble Image Segmentation: An Approach". In: *Proceedings of the 2022 5th International Conference on Computational Intelligence and Intelligent Systems*. CIIS '22. Quzhou, China: Association for Computing Machinery, 2023, 9–14. DOI: 10.1145/3581792.3581794. URL: https://doi.org/10.1145/3581792.3581794.

[Rus] D. Rusin. *Mathematical Atlas*. Dave Rusin. URL: https://books.google.de/books?id=zZG-DAEACAAJ.

[Rus24] D. Rusin. *Math Atlas. Mathematics Books Archive*. Accessed: 2024-10-17. 2024. URL: https://ns1.omnitech.net/data/Books/Math/.

[sci24] scikit-learn developers. *Multidimensional Scaling (MDS)*. Accessed: 2024-10-17. 2024. URL: https://scikit-learn.org/1.5/modules/manifold.html.

[Wor24] World Wide Web Consortium. *RDF - Resource Description Framework*. [Online; accessed 3-November-2024]. 2024. URL: https://www.w3.org/RDF/.

[zbM24] zbMATH Open. *The World's Most Comprehensive Database of Mathematical Literature*. Accessed: 17 October 2024. 2024. URL: https://zbmath.org/.