# Folding paper from planar tilings

Huma Akkem David Goncharov Matt Grode Natalia Sliwinska
University of Illinois at Chicago

MSCS Undergraduate Research Laboratory

## Summary

Origami is an art of folding materials, usually paper or other decorative fabrics, to create three-dimensional shapes or tilings. Corrugations and tessellations are particular types of origami, usually being built from certain types of repeated and convex objects. The folding of origami requires a carefully calculated two-dimensional crease pattern. We built a computer program that automatically turns $k$-uniform tilings into a crease pattern that is ready-to-fold or input into an origami simulator.

## Previous work

This project was inspired by the work of origami artists Uyen Nguyen and Ben Fritzson. In Nguyen and Fritzson's research paper, *Origami Explorations of Convex Uniform Tilings Through the Lens of Ron Resch's Linear Flower*, the artists proposed a process for going from a $k$-uniform tiling to its crease pattern. A $k$-uniform tiling is one which consists edge-to-edge connected convex regular polygons with $k$ orbits of vertices. The process of creating a going from a tiling to its crease pattern is roughly as follows:

First, choose a polygon and make a reduced copy of that polygon. Second, make a midpoint polygon whose edges are connected to the reduced polygon at its midpoints. Third, make the shared legs at each corner of the polygon. Fourth, add side crease lines that connect the midpoint polygon to the legs.

In addition, Fritzson and Nguyen suggest that varying the distance between the starting and reduced polygons are allowable within certain bounds, which would change the overall height of the tiling. In order to assure foldability, one needs to compute a shifting quantity known as a $B$-shift, which consists of solving a system of equations numerically.

## Motivation

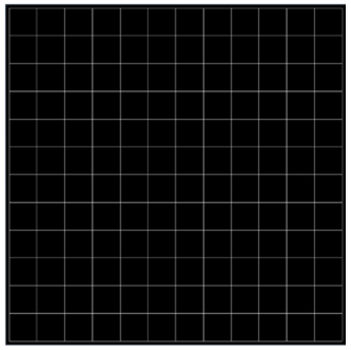
Fritzson and Nguyen's algorithm is time consuming to implement by hand and requires numerically solving systems of equations in addition to highly repetitive planar geometry calculations. This made it unreasonably difficult to construct to origami from more complicated patterns. By automating the algorithm, we can speed up the process and handle more complicated $k$-uniform tilings.

## $k$-uniform tilings

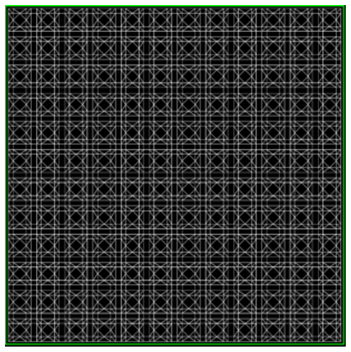
The tilings we consider tile the plane by normal convex polygons so that there no empty space between the shapes. Further, we restrict ourselves to tilings where each edge is the same length, and each edge touches exactly two polygons. Such a tiling is $k$-uniform if there are exactly $k$ types of vertices, meaning that we may partition the set of vertices into $k$ sets, where all vertices in a given set can be mapped to each other.

## The process for a square tiling

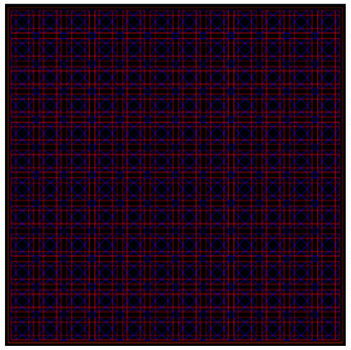
First we draw the tiling.



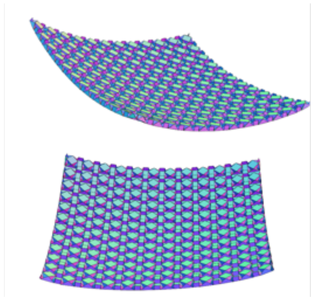
Then we fill in the crease pattern using Nguyen and Fritzson's algorithm



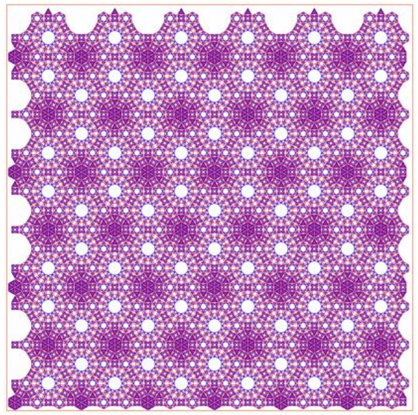
Then color mountains and valleys

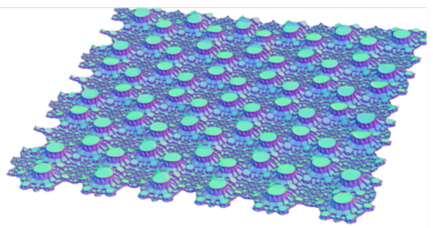


Then feed the crease pattern into an online origami simulator.



## A more complicated tiling





## List of $k$-uniform tilings

| $k$ | Number of $k$-uniform tilings |
|---|---|
| 1 | 11 |
| 2 | 20 |
| 3 | 61 |
| 4 | 151 |
| 5 | 332 |
| 6 | 673 |
| 7 | 1472 |
| 8 | 2850 |
| 9 | 5960 |
| 10 | 11866 |
| 11 | 24459 |
| 12 | 46794 |
| 13 | 103082 |

The numbers for $k = 8-13$ appear to only recently have been computed in 2021 by Marek Čtrnáct via a brute-force search. The values for $k \geq 14$ do not appear to have been computed.

## A description of the program

The program first draws the tiling by adding one vertex at a time according to a "gluing pattern"; then adds in the crease pattern according to the algorithm by Nguyen and Fritzson; and finally colors the edges with mountains and valley folds so that it can be fed into the origami simulator. The program takes in arguments for dimensions, edge size, a shift factor $h$, and starting polygon.

## Gluing patterns

Tilings are described by gluing patterns, which describe the angles where each edge appears as well as what edges of each vertex are glued to. For instance, the square tiling has the following gluing pattern and angle list:

$$[0, \pi/2, \pi, 3\pi/2]$$
$$\{(0,0):(0,2),(0,1):(0,3),(0,2):(0,0),(0,3):(0,1)\}$$

The program has built-in many possible gluing patterns ranging from simple 1-uniform tilings to a few 5-uniform tilings, as well as the option of adding in a new one.

## $B$-shift

To ensure foldability, for each type of polygon we have to solve the following system of equations, where $c_1, c_2, c_3$ are parameters depending on the shape and $x, y, z, w$ are variables we solve for:

$$x + y = c_1$$
$$z + w + c_2 = \pi$$
$$c_3 \sin(c_2) = x \sin(w)$$
$$y \sin(c_2) = x \sin(z)$$

This is done numerically in Python.

## Future directions

Among the many possible future directions are adding more flexibility to allow different types of tilings in addition to $k$-uniform tilings. Additionally, for larger patterns the program is currently fairly slow, taking roughly 30 minutes, and so speeding it up to run near-instantaneously would be useful.

## References

U. Nguyen, B. Fritzson. *Origami Explorations of Convex Uniform Tilings Through the Lens of Ron Resch's Linear Flower*. Proceedings of Bridges 2018: Mathematics, Art, Music, Architecture, Education, Culture. 2018.

*Online Encyclopedia of Integer Sequences, sequence A068599*. `https://oeis.org/A068599`.