

# Surface Toppling on Cylindrical and Polyhedral Sandpiles

Benjamin R. Trube

Marion, Ohio, USA; bentrubewriter@gmail.com

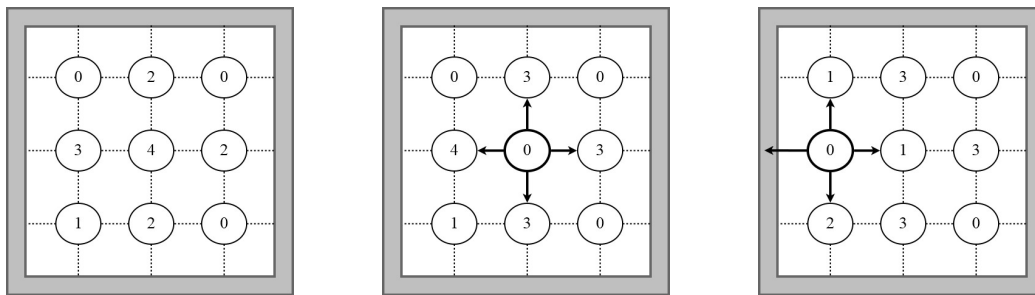
## Abstract

Abelian Sandpiles are commonly investigated on unbounded two-dimensional square lattices. Less explored is the behavior of sandpiles on three-dimensional surfaces. In this paper I investigate partially and fully closed sandpile graphs that can be wrapped around cylinders and cubes. Aside from attractive symmetries not possible on a uniform flat lattice, I am interested in the ways that bending the sandpile in on itself this way can produce novel interactions. I study how the quantity and placement of drop spots affected the final stabilized pattern aesthetically and the density of sand throughout the sandpile.

## Abelian Sandpiles

*Abelian Sandpiles* are a type of chip-firing model, first introduced by Bak, Tang, and Wiesenfeld [1]. Like other chip-firing models, the sandpile can be described as an undirected graph  $G$  containing vertices and edges. Edges connect neighboring vertices, and each vertex contains a whole-number quantity of grains of sand. Initially, grains of sand are distributed among the vertices of  $G$ , commonly a high quantity of sand in one vertex.

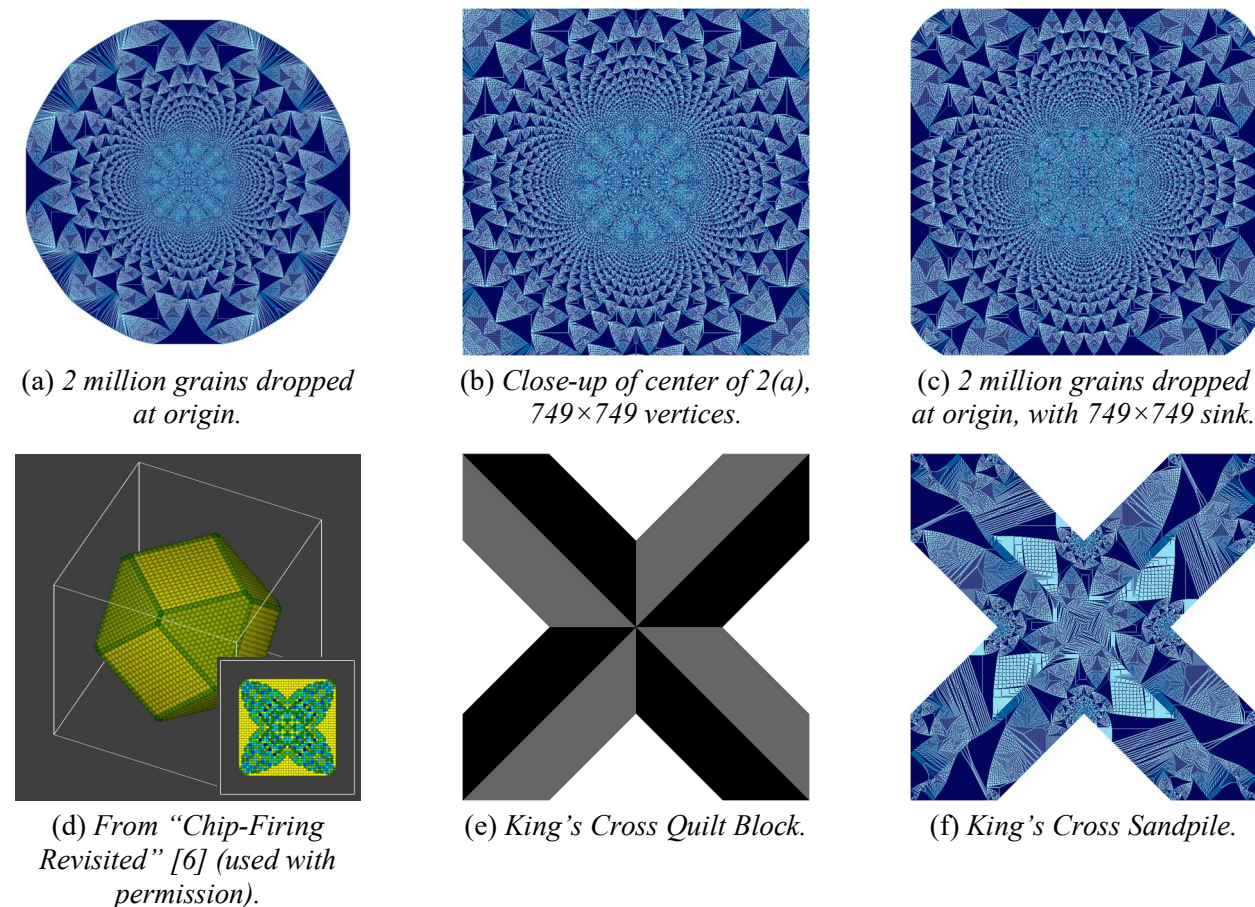
After the sand has been dropped, we check to see if any vertices contain a quantity of sand that is greater than or equal to the *toppling threshold*. Vertices in this condition are *toppled*, which causes sand from that vertex to be evenly distributed among its neighbors. This process is then continued until the model reaches a stable state where no vertices can be toppled. The threshold is equal to the degree of the vertex. For a square grid, as in Figure 1, each vertex has degree 4; it has 4 edges connecting to its 4 neighbors.



**Figure 1:** *Demonstration of toppling on a grid bounded by a sink, with numbers indicating quantities of sand grains. Left: Initial state. Middle: The vertex in the center has been toppled into its neighboring vertices. Right: The vertex on the left has been toppled into its neighboring vertices and the sink.*

Edges can also connect to an outer boundary called the *sink*. We can imagine sand being dropped on a table, with some sand falling over the side. The table's edge is the sink, and grains of sand that fall into the sink are no longer part of the sandpile. The right-hand configuration in Figure 1 shows a vertex that has toppled sand to vertices above, below, and to the right, and also a grain into the sink on the left.

Unbounded graphs, as well as bounded graphs with at least one vertex connected to a sink, are guaranteed to stabilize [4], assuming a finite quantity of sand. Conversely, a closed graph can fail to stabilize depending on the initial configuration and quantity of sand [2]. This will become relevant as we consider closed graphs like the cube surface sandpile described later in this paper. All sandpile graphs considered in this paper contain vertices of degree 4.



**Figure 2:** Sandpile visualizations. 2(a)–(c): Classical model. 2(d)–(f): Previous Bridges contributions.

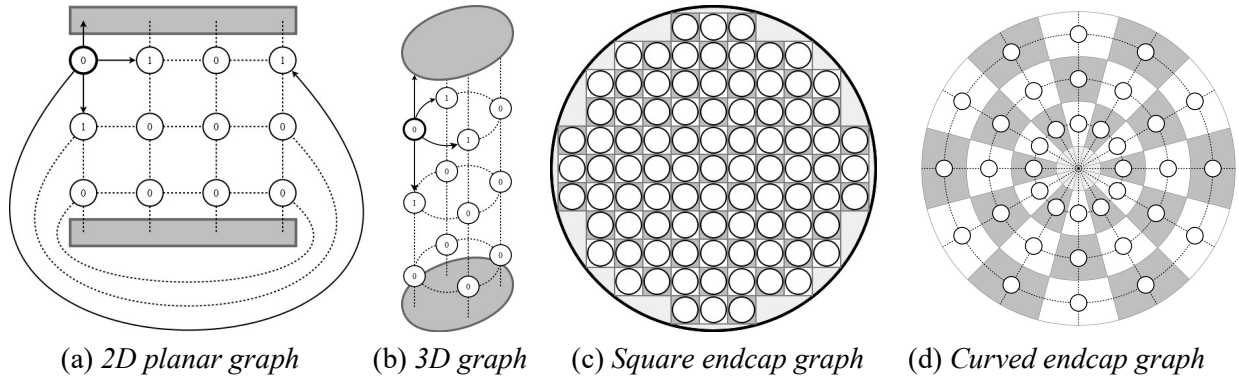
Once a sandpile graph has stabilized, it can be colored according to the quantity of grains of sand at each vertex: 0, 1, 2, or 3. The result is a fractal-like pattern (Figure 2(a)–(c)). Throughout this paper, vertices on grid lattices will be colored light blue if they contain 0 grains of sand (hex value: ■ #CAF0F8), and navy blue if they contain 3 grains of sand (hex value: ■ #03045E). Vertices containing 1 grain of sand are colored a slightly darker blue (hex value: ■ #00B4D8) than those containing 2 grains (hex value: ■ #90E0EF) for better contrast of internal features. The color choices were made to show a strong contrast between vertices containing the maximum number of sand grains (3) and all other vertices.

Bak, Tang, and Wiesenfeld considered toppling models in one, two, and three dimensions before settling on a two-dimensional cellular automaton [1]. The sandpile has been extensively studied in two dimensions, though nothing in the basic dynamics of the sandpile requires this limitation. Previous Bridges publications have explored the possibility of lifting the Abelian Sandpile into the third dimension. Martin Skrodzki and Ulrich Reitebuch considered various three-dimensional neighborhood configurations and initial uniform values for each vertex to produce solid sandpiles whose internal features could be explored by slicing [6]. The outer shells of these sandpiles resembled different polyhedrons (Figure 2(d)), and the internal slices produced unique patterns that could not be reproduced in two dimensions.

My own previous Bridges work used quilting blocks as a way of defining the boundary sink of a sandpile, with the internal features of the block used to provide initial values for the vertices inside the sandpile [7]. Figure 2(e) shows an initial quilt block, and Figure 2(f) shows the resulting “quilting sandpile” with a square spiral feature at the center. Both works explore ways to produce varying designs in the resulting stabilized sandpile by changing the boundaries or lattice structure of the sandpile graph, and this paper considers another such technique, something I’m calling *surface toppling*.

### Surface Toppling on a Cylinder

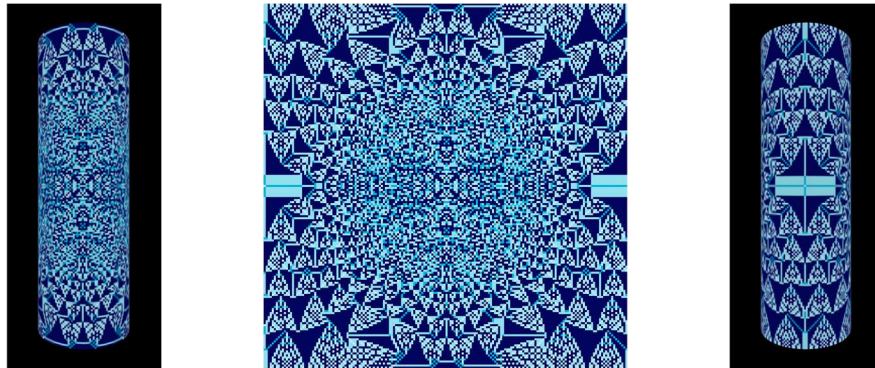
Consider again a graph on a square grid. Instead of connecting the vertices on the left and right sides to the sink, we create another edge that connects the left and right sides to each other. Toppling a vertex on the left sends grains of sand to the right and vice versa. The vertices on the top and bottom of the graph are still connected to the sink, meaning that the sandpile will reach a stable configuration. Figure 3(a) demonstrates this behavior on a 2D planar graph. The vertex in the top left has been toppled, sending one grain of sand to the right, one grain downward, one grain up into the sink, and one grain to the left, which ends up on the right side of the graph. Figure 3(b) shows the same vertex being toppled in three dimensions. Sand is toppled along the surface of a hollow cylinder but is not toppled through the inside or over the endcaps.



**Figure 3:** Cylindrical toppling behavior and unused endcap configurations.

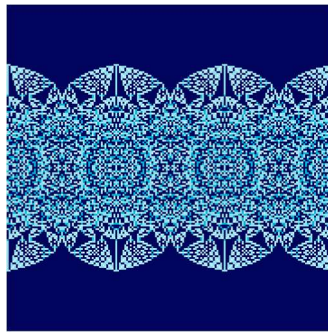
When the cylinder is digitally rendered, the  $m \times n$  planar graph is drawn using  $m \times n$  pixels, as shown in the middle image of Figure 4. This 2D image texture is then wrapped around a 3D cylinder frame (Figure 4 left and right images). Unlike the body of the cylinder, which can be represented in 2D with a square grid, the endcaps are circular. Figure 3(c) shows a possible endcap arrangement with square pixels, but there are gaps around the outside. Figure 3(d) is an alternative arrangement that keeps four neighbors for each vertex, but to render this you would need curved trapezoids instead of squares. As you can see from Figures 3(c) and 3(d), a circle cannot be cleanly mapped to a square grid or drawn using just square pixels. By leaving off the endcaps, I make it easier to render the cylinder sandpile, and the endcaps also function as a sink, guaranteeing stabilization.

For the cylinders in this paper, I used a  $180 \times 181$ -vertex lattice and dropped grains of sand at various points along the line around the middle of the cylinder. Figure 4 shows the resulting sandpile after 100,000 grains of sand are dropped at a single point and allowed to stabilize.

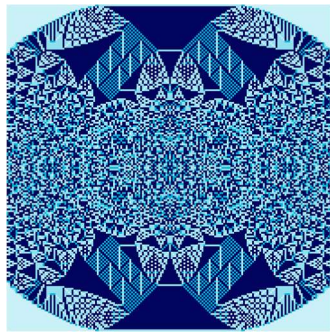


**Figure 4:** Cylindrical sandpiles. Left: Three-dimensional, viewed from the drop point. Middle: Two-dimensional rendering. Right: Three-dimensional, viewing the intersection of the left and right sides.

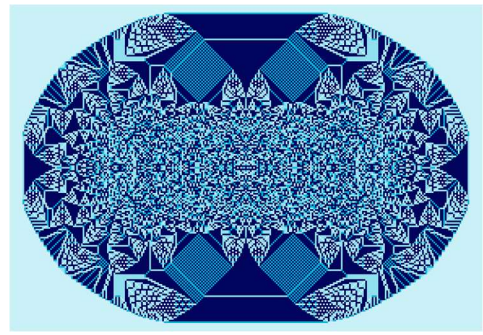




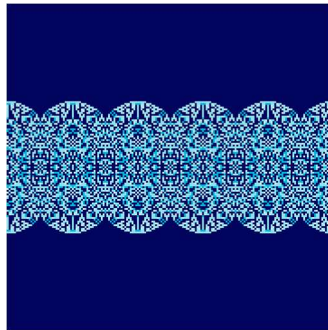
(a) 3 drop spots



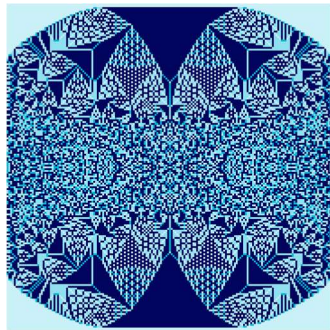
(b) 3 drop spots with sink



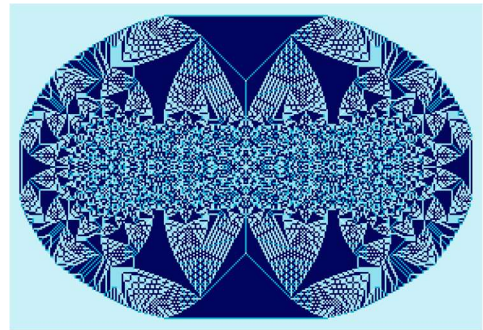
(c) 3 drop spots, unbounded



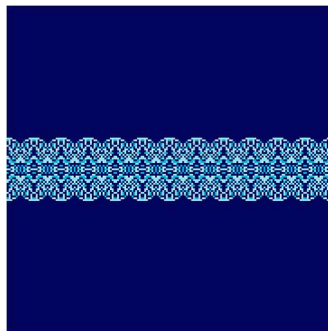
(d) 5 drop spots



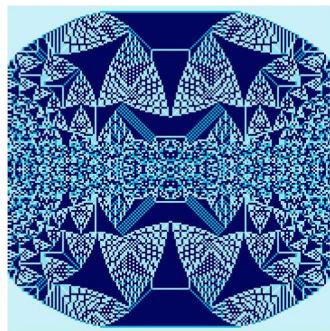
(e) 5 drop spots with sink



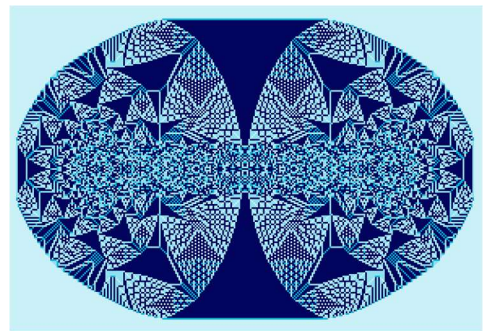
(f) 5 drop spots, unbounded



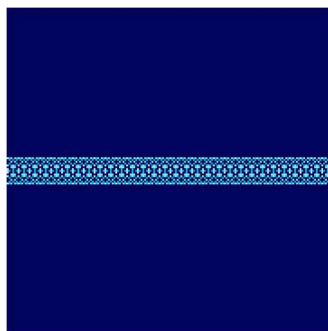
(g) 12 drop spots



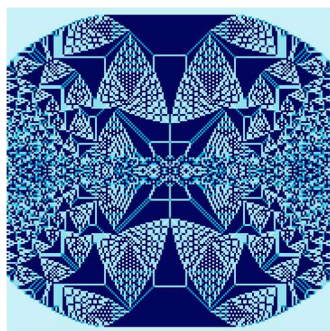
(h) 12 drop spots with sink



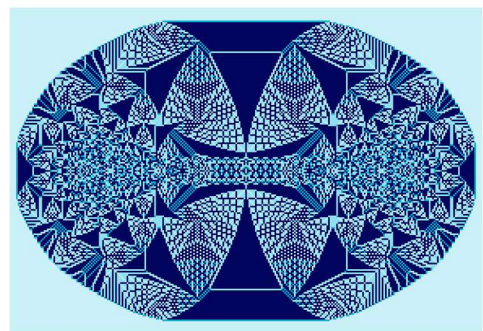
(i) 12 drop spots, unbounded



(j) 30 drop spots



(k) 30 drop spots with sink



(l) 30 drop spots, unbounded

**Figure 5:** 100,000 grains of sand divided equally between equidistant drop spots on a  $180 \times 181$  grid. Left column: Two-dimensional rendering of cylinder sandpile. Middle column: Equivalent sandpile with a bounding sink on all four sides. Right column: Unbounded sandpile.

Figure 5 shows the effects of increasing the number of drop spots around the cylinder. In each case, 100,000 grains of sand were added to the sandpile, divided into piles an equal distance apart. (In instances where 100,000 grains did not divide evenly by the number of drop spots, I rounded down to the nearest whole number of grains per drop.) Using multiple drop spots produces a regular periodic pattern, with the number of pattern repetitions corresponding directly to the number of drop spots. The number of vertices with 3 grains of sand also increases (seen visually as the amount of navy blue in each picture), so I measured the remaining sand throughout the pile to get a sense for the average sand density in grains per vertex.

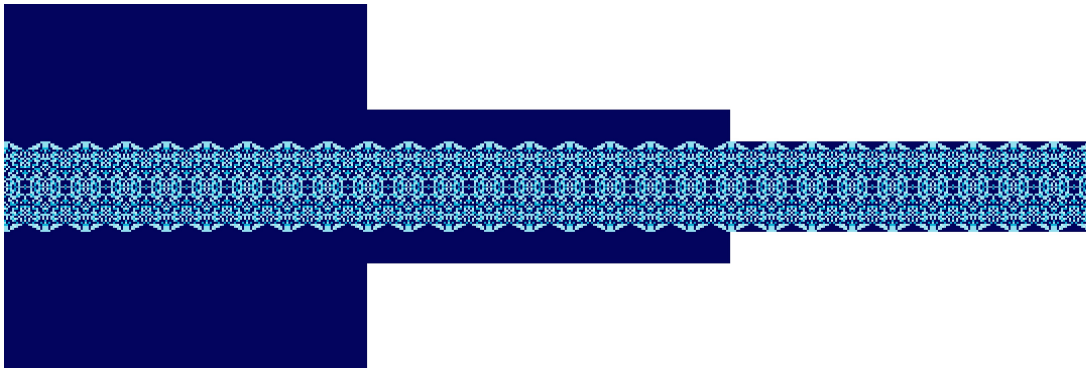
**Table 1:** *Average density of sand as a relation to the number of drop spots.*  
*Number of vertices = 32,580 for a  $180 \times 181$  sandpile.*

# of Drop Spots	Remaining Sand	Average Density
1	73,062	2.2425
3	81,639	2.5058
9	90,999	2.7931
30	95,310	2.9254
90	96,750	2.9696
180	97,740	3.0000

Table 1 shows the quantity of grains of sand remaining and the calculated average density per vertex after each cylindrical sandpile stabilizes. As the number of drop spots increases, the density of sand per vertex goes up as well until it reaches the maximum stable configuration of 3 grains of sand for every vertex in the sandpile. Contrast this with the average density of the unbounded sandpile, 2.125 grains of sand per vertex, measured for vertices within the spread boundary of the stabilized pile [3]. It's unclear why connecting the left and right sides of the pile seems to have this leveling effect, even with sinks at both ends. The more that sand is spread around, the more uniform (and denser) it becomes throughout the pile.

To confirm this is a property of the cylinder sandpile and not normal sandpile behavior, the middle column of Figure 5 shows the same amount of sand dropped at the same drop spots for a sandpile bound by a square sink. The right column shows the resulting sandpile on an unbounded lattice. In both cases the stable pile is an oval with a lower average sand density, and a horizontal band at its center from which the rest of the pile spreads.

On the cylinder sandpile, this horizontal band is a “groove” of vertices with 2 grains of sand or fewer, surrounded on the top and bottom by a flat sea of vertices with 3 grains. The height of this groove remains constant even if we change the height of the cylinder down to the height of the groove. Figure 6 shows 99,999 grains of sand dropped in 9 places with cylinders of varying height.

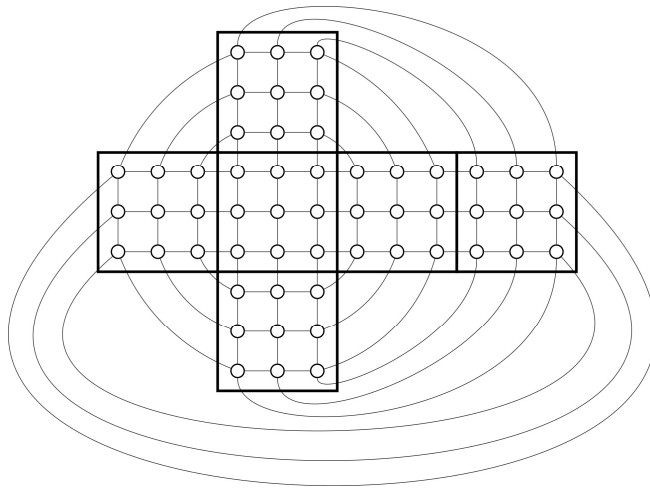


**Figure 6:** *Left: 99,999 grains dropped on 9 drop spots on a  $180 \times 181$  lattice. Middle: The same amount of sand dropped on a  $180 \times 77$  lattice. Right: The same amount of sand dropped on a  $180 \times 45$  lattice.*



### Surface Toppling on a Cube

Sand on the surface of a cylinder travels in two directions: along the curvature of the cylinder or toward one of the endcaps. But what if we were to try an object that can topple sand over multiple axes?



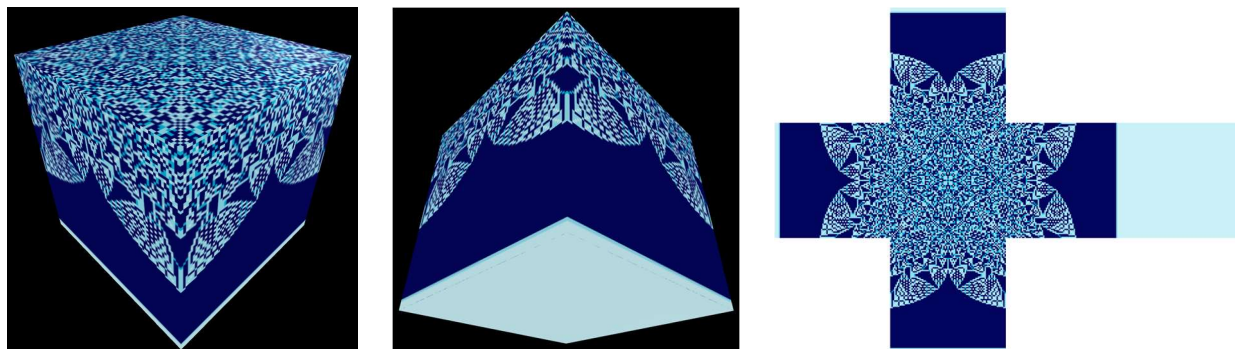
**Figure 7:** Two-dimensional representation of a cube surface graph with vertices, connecting edges, and cube faces.

A cube has six faces, each of which can be represented as a grid lattice and drawn in 2D with  $n \times n$  pixels. Figure 7 shows how vertices at the boundary of each face are connected on a planar representation. Importantly, the vertices are not placed on the border of each face. Rather, the connecting edge runs over the side of the cube. This avoids the complication of rendering vertices that are not on a single cube face. This also has the benefit of keeping all vertices degree 4, as the eight cube corners would otherwise be degree 3. Non-uniform surface graphs may be considered as an extension of surface toppling in future work.

Because the cube is a closed graph, there is a possibility that the sandpile won't stabilize. Fortunately, determining a safe amount of sand to add is a relatively simple calculation. Given  $n$  vertices and  $m$  edges, with  $N$  being a whole-number initial quantity of sand, we can determine if the cube will stabilize using the following rules [2].

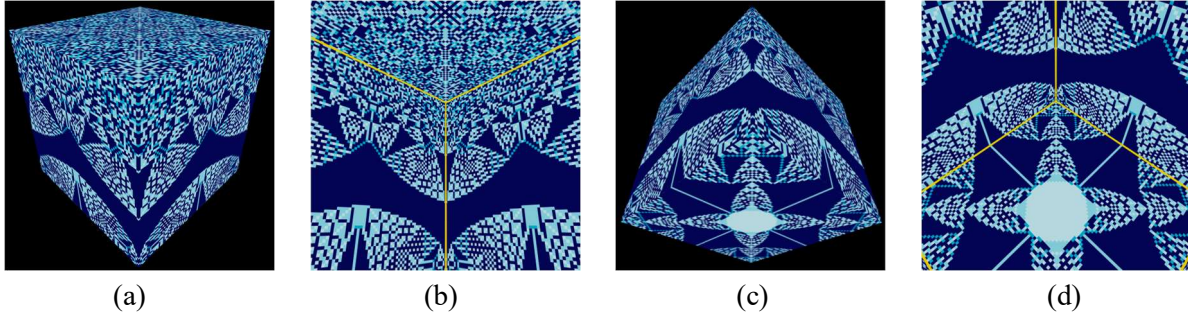
1. If  $N > 2m - n$ , the sandpile will never stabilize.
2. If  $m \leq N \leq 2m - n$ , then some configurations will stabilize while others will not.
3. If  $N < m$ , then the sandpile will always stabilize.

Consider a cube with  $87 \times 87$  vertices on six faces (I chose 87 vertices on a side since the expected density is close to the unbounded pile for 100,000 grains). The total number of vertices is  $n = 45,414$  and the total number of edges is  $m = 90,828$ . The pile will stabilize if we drop less than 90,828 grains of sand and may stabilize if we drop up to  $2m - n = 136,242$  grains; anything more will infinitely topple. Figure 8 shows a cube sandpile with 90,827 grains dropped in the center of the top face. The bottom of the cube is completely empty of sand, suggesting that more sand can be added before the pile will fail to stabilize.



**Figure 8:** 90,827 grains dropped in the top center of a cube sandpile with  $87 \times 87$  vertices on a face. Left: Top-down view. Middle: Bottom-up view. Right: Flattened cube, top in the center.

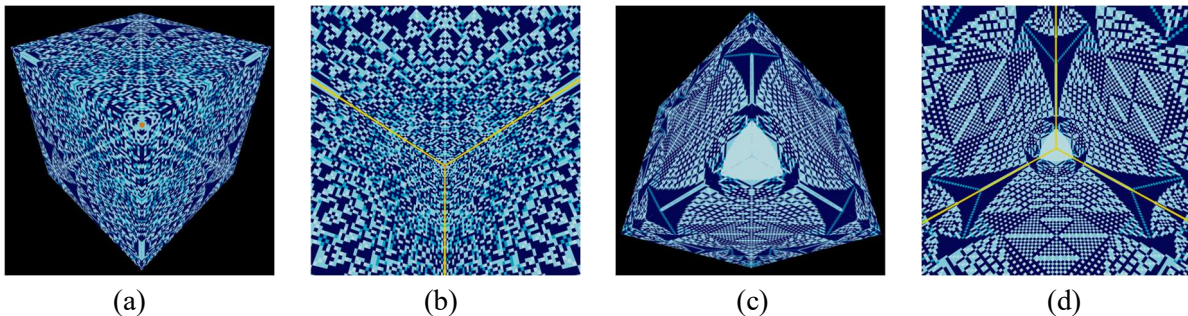
After some experimentation, it appears that the limit for this particular cube sandpile and configuration is 104,599 grains. That makes the average sand density about 2.303 grains per vertex, though some areas midway down the sides and around the bottom are denser, and the center of the bottom is empty. Figure 9 shows the stabilized pile from different perspectives. Because the cube is hollow, we can view the sandpile from both the outside and the inside.



**Figure 9:** 104,599 grains dropped in the top center of a cube sandpile with  $87 \times 87$  vertices on a face.  
 (a) Top-down view. (b) Internal view, looking at top (cube edges in yellow).  
 (c) Bottom-up view. (d) Internal view, looking at bottom (cube edges in yellow).

The four vertical faces of the cube are identical, as expected given the symmetrical spread. The sand spreading down the sides of the cube meets at the bottom to form a circular pattern. A barrier 3 grains tall separates the pattern on the top and bottom of the cube, but the cube is all one sandpile.

With three axes along which sand can topple, we can also produce three-fold symmetry that wouldn't be possible on a normal grid lattice. Figure 10 shows a cube sandpile with 99,999 grains split between three vertices that meet at one corner of the cube. The front, left, and top faces are all rotations of the same image, as are the back, bottom, and right faces.



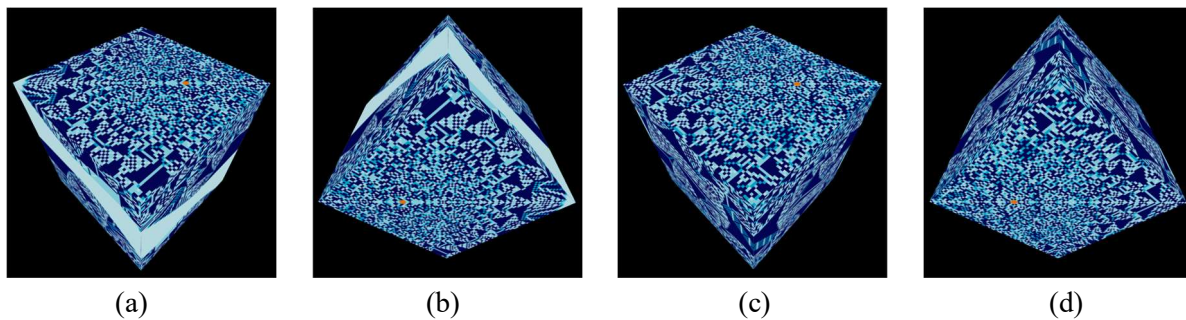
**Figure 10:** 99,999 grains dropped in one corner of a cube sandpile with  $87 \times 87$  vertices on a face.  
 (a) Drop corner view (drop spots marked in orange). (b) Internal view, looking at drop corner.  
 (c) Opposite corner view. (d) Internal view, looking at opposite corner.

The colloquial description of the sandpile model includes stacks of grains that topple, but from an abstract point of view, both the internal and external representation are valid ways of looking at the final pile. This is especially striking for our three-fold symmetry cube, where from the external perspective the sides are folding away from the viewer, while from the inside the faces bend toward the viewer, changing the appearance of the same formations (Figures 10(a) and 10(b), 10(c) and 10(d)). The images in Figures 8–11 were produced using software that allows the user to interact with the cube—turning it to see each face, or even flying inside (available at <https://github.com/RealFractalMan/toppling-sandpiles>).

With six faces, the cube surface opens many possibilities for drop spots in different combinations (more than can be definitively addressed in this paper). But while some possibilities have predictable results

(dropping in the center of all six faces produces a cube with six identical sides), we are not limited to drop spots at the center or edges.

Using the same  $87 \times 87$ -vertex cube we can drop sand equally on the top and bottom of the cube. On the top we drop sand a third of the way from one corner, and on the bottom we drop sand a third of the way from the opposite corner. Figures 11(a) and 11(b) show the stabilized sandpile with 80,000 grains dropped to illustrate the way the pile spreads, with 11(c) and 11(d) using 102,706 grains (which, from experimentation, seems to be the stabilizing limit). The stabilizing limit is less than our top-only drop, illustrating that configuration does play a role in the stabilization limit, as discussed in [2].



**Figure 11:** Skewed drops (marked in orange) on the top and bottom of the cube. (a) and (b) 80,000 grains (40,000 on top, 40,000 on bottom). (c) and (d) 102,706 grains (51,353 on top, 51,353 on bottom).

## Summary and Conclusions

Behaviorally, the cube surface sandpile is consistent with studies of closed graphs, but the patterns that can be produced have the potential for unique artistic and interactive representations. The cylinder surface, on the other hand, produced leveling behavior that seems atypical of most sandpiles. Fluid dynamics may offer some insight into this behavior. The surface toppling technique can be applied to other regular polyhedrons, with tetrahedrons, octahedrons, and icosahedrons possible using hexagonal lattices, and hopefully this introduction to surface toppling will inspire the reader to conduct their own experiments.

## Acknowledgements

The author would like to thank Brian Buckley for his editorial review and for fruitful discussions as this concept was being developed. Thanks to Martin Skrodzki and Ulrich Reitebuch for permission to reproduce their artwork [6]. Thanks to Bruce Sherwood for his timely guidance on how to apply textures with the VPython library [5]. And lastly, thanks also to my wife Hannah for all her encouragement.

## References

- [1] P. Bak, C. Tang, and K. Wiesenfeld. “Self-Organized Criticality: An explanation of  $1/f$  noise.” *Physical Review Letters*, vol. 59, no. 4, 1987, pp. 381–384.
- [2] A. Björner, L. Lovász, and P. W. Shor. “Chip-firing Games on Graphs.” *European Journal of Combinatorics*, vol. 12, no. 4, 1991, pp. 283–291.
- [3] J. Ellenberg. “The Math of the Amazing Sandpile.” *Nautil.us*, Oct. 6, 2021 (reprint). <https://nautil.us/the-math-of-the-amazing-sandpile-238320/>.
- [4] C. J. Klivans. *The Mathematics of Chip-firing*. CRC Press, 2018.
- [5] B. Sherwood. *VPython: 3D Programming for Ordinary Mortals*. 2023. <https://www.vpython.org>.
- [6] M. Skrodzki and U. Reitebuch. “Chip-Firing Revisited: A Peek Into The Third Dimension.” *Bridges Conference Proceedings*, Helsinki and Espoo, Finland, Aug. 1–5, 2022, pp. 221–228. <https://archive.bridgesmathart.org/2022/bridges2022-221.html>.
- [7] B. Trube. “Abelian Sandpile Quilting Blocks.” *Bridges Conference Proceedings*, Helsinki and Espoo, Finland, Aug. 1–5, 2022, pp. 437–440. <https://archive.bridgesmathart.org/2022/bridges2022-437.html>.