

Chip-Firing with Walls and Dying Vertices

Benjamin R. Trube

Marion, Ohio, USA; bentrubewriter@gmail.com

Abstract

I experiment with chip-firing models with a non-uniform graph degree, using fixed boundaries to block or redirect the flow of chips within the system. I explore some ways in which these boundaries can be used to affect the stabilized pattern. I then introduce vertex death, which changes the graph degree of the surrounding vertices as their neighbors die, before the chip-firing model has stabilized. I observe that consistent final configurations can be achieved despite these changing dynamics, provided vertex firings and death are handled uniformly.

Chip-Firing Behavior

Chip-firing models or games involve taking a fixed quantity of chips, coins, or grains of sand and moving them between vertices of an undirected graph G . Edges connect neighboring vertices, and the number of edges determines the graph degree of each vertex. A vertex contains a non-negative integer quantity of chips. If a vertex has a quantity of chips greater than or equal to the graph degree, it will pass an equal amount of chips to each of its neighbors through a process called *firing*. This firing continues until each vertex's chip count is less than its graph degree. If all vertices have stopped firing, the chips remaining in each vertex can be colored according to their quantity. On larger graphs this produces intricate patterns. In this paper I experiment with chip-firing on non-uniform grids by removing vertices ("creating walls") from uniform grids when defining the initial graph configuration, or dynamically as the game is being played. Previous Bridges contributions have considered one-dimensional [3], three-dimensional [6], and surface [7] chip-firing models.

Figure 1(a) – (d) demonstrates firing behavior with vertices of varying graph degrees on a 3×3 grid. Figure 1(a) shows an initial configuration of randomly placed chips. The center vertex is fired in Figure 1(b) because it contains 5 chips, which is greater than its graph degree. It sends 1 chip to each of its neighbors equally; 1 is left over since the model does not allow fractional chips. If the center had 8 chips, it would send 2 chips to each neighbor (a firing can send more than 1 chip). Figure 1(c) shows a degree 3 vertex firing since it now contains 3 or more chips, again sending chips equally to its neighbors. The four corners of the grid are degree 2 and will fire if they have 2 or more chips, as we see in Figure 1(d). After this final firing, the chip-firing model is *stable*; no more vertices can be fired. Some configurations, such as Figure 1(e), will never stabilize (5 chips on a 2×2 grid), resulting in an infinite cyclical loop of firings.

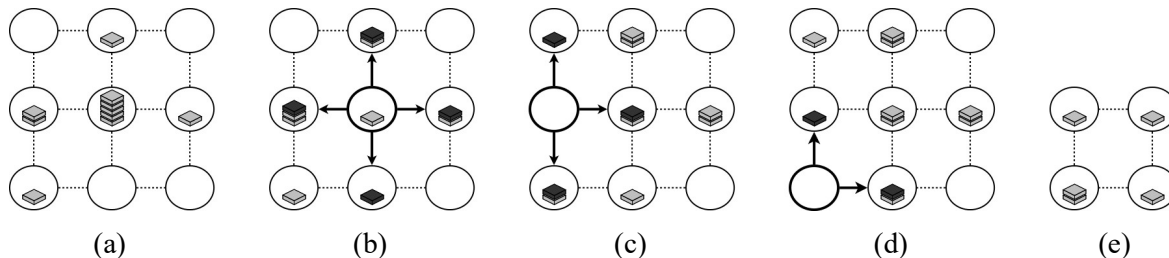


Figure 1: Demonstration of chip-firing on a 3×3 grid.

Two adjacent vertices have a property called *local confluence* [4] if they can be fired in any order with the same resulting configuration. On the wider graph, chip-firing models that stabilize will have a unique final configuration unaffected by the order in which the vertices are fired. This is called *global confluence* [4]. Chip-firing systems whose vertices all demonstrate local confluence will also have the property of global

confluence. Felzenswalb and Klivans have also explored chip-firing systems that demonstrate global confluence without satisfying local confluence in [2].

Adding Walls to Chip-Firing Models

On a uniform grid, interior vertices are degree 4, vertices on the outside of the grid are degree 3, and the four corners are degree 2. Within this grid we can choose specific vertices to be *walls* into which no chips can be moved. This changes the graph degree of vertices adjacent to the walls, as we see below.

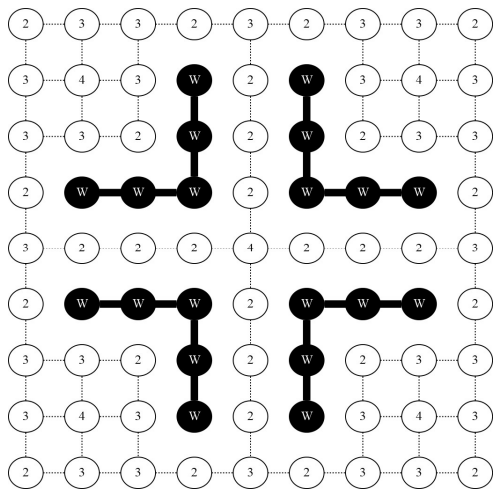


Figure 2: Chip-firing model with walls.

Figure 2 shows an example of such an arrangement. Four L-shaped walls have been introduced to create a one-vertex-wide cross through the center of the grid (represented by the black vertices with a white W). Edges can connect around but not through these walls. The graph degree of each vertex is shown in the white vertices. The central vertex remains degree 4, as it still can be fired in four directions. The vertices on the inside of the cross are degree 2, and the surrounding vertices are of varying degrees between 2 and 4. We can also think of the outside of the graph as functioning like a wall, as no chips can be fired outside the graph.

Using walls of varying shapes, we can affect the resulting configuration and the ways the chips move. In Figure 2, the chips at the center will move along narrow channels instead of spreading in all directions. Sand outside the cross will also be blocked on some paths.

To see the wall’s effect on final configuration, let’s consider a simple chip-firing model, the *Abelian Sandpile*, first described by Bak, Tang, and Wiesenfeld [1]. The sandpile involves placing a large quantity of chips (or sand grains) at the center of a large uniform grid. When the chip-firing process stabilizes, we color each vertex according to the quantity of sand remaining: 0, 1, 2, or 3 sand grains. This produces a fractal-like pattern (Figure 3 Left). Figure 3 Right shows the result of dropping these grains into a larger one-vertex-wide cross like the one in Figure 2 (this cross is 161×161 vertices placed in the middle of a 401×401 grid). The bounding shape has changed, as well as the interior detail.

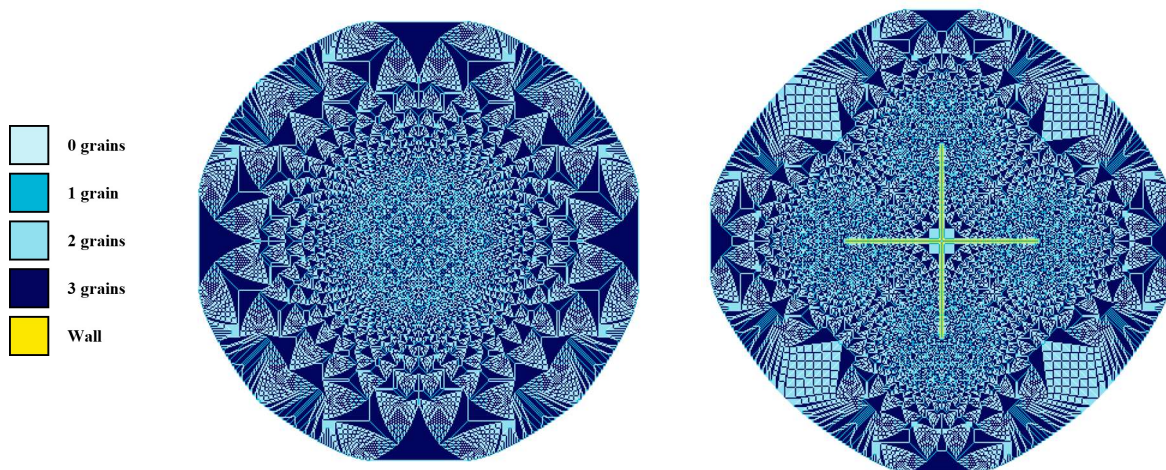


Figure 3: Results after 250,000 are chips added at the center and fired. Left: *Abelian Sandpile*. Right: Reshaped sandpile with one-vertex-wide cross at the center (highlighted in yellow).

Alternative wall arrangements can redirect the chips in all sorts of aesthetically pleasing ways. Chips can be bounced off walls, clumped, or pushed into corners. Figure 4 shows a few examples. Figure 4(a) drops 250,000 chips at the center, with the flow of chips affected by two diamond walls with slits at the middle. Figure 4(b) is similar, with an inner square wall with slits on each side, and a second outer set of walls placed outside those slits. Figure 4(c) uses 250,000 chips dropped into an alternating grate arrangement of diagonal walls. Figure 4(d) looks like the classic Abelian Sandpile, though in this case the four quadrants are completely separated by vertical and horizontal walls that cross at the center. 62,500 chips were dropped into each of the four corners at the center of the model, with no interaction between the four quadrants.

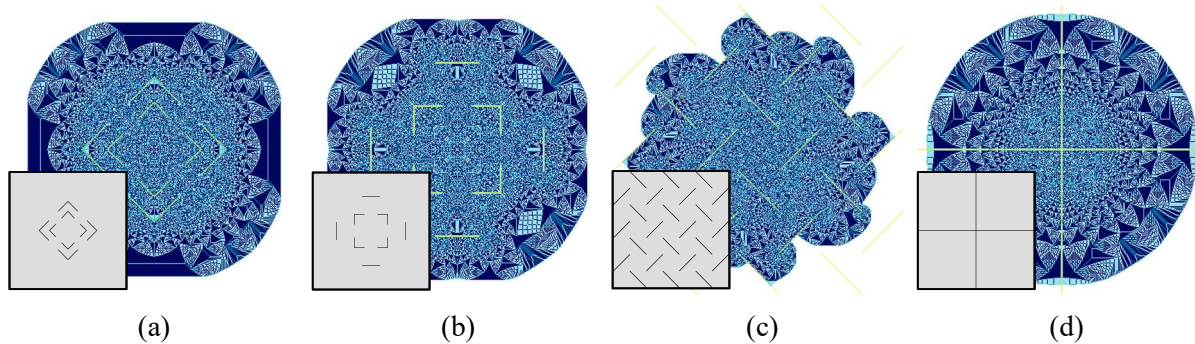


Figure 4: (a) – (d) *Chip-firings with walls in different configurations.*

In these arrangements (and others we might imagine), the global confluence property holds, allowing for efficient parallel calculations without affecting the final pattern. Depending on the arrangement of walls, we can tweak the rotational symmetry, affect the spread of chips, or change the final bounding shape. Figure 5 shows a chip-firing system with a bounding wall 8 vertices thick at its smallest point, formed by the 32-segment quadratic Koch curve [5]. The figure is the result of adding 764,820 chips. I chose this curve because the path of the chips is not direct; the curve is gnarled. Figures 4(c) and 5 suggest that adding more walls that force the sand to change directions can produce a noisier result.

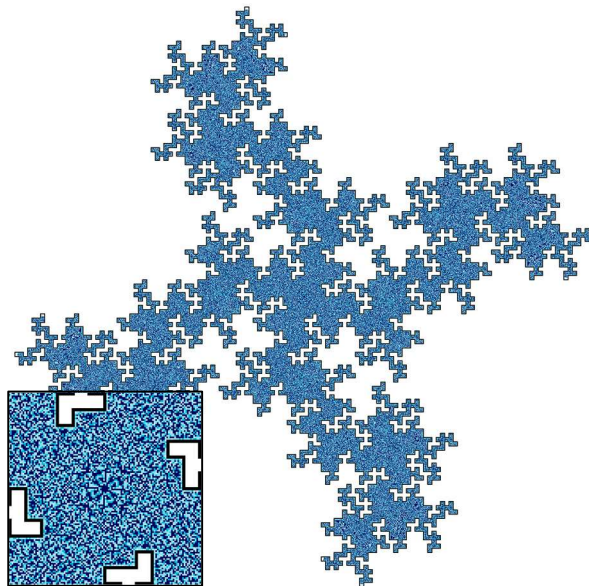


Figure 5: *Chip-firing inside a 32-segment quadratic Koch curve (detail zoom on center).*

Adding Vertex Death

A vertex in a chip-firing system can fire hundreds, thousands, or even millions of times, depending on the initial configuration. But what if the vertex “wore out” and died, turning into a wall upon its death?

When a vertex fires, it sends equal amounts of chips to each living neighbor, leaving behind an amount of chips less than the graph degree of the vertex. With vertex death, after firing a set number of times, the vertex dies, and any chips remaining are colored when the system stabilizes. Neighbors cannot send chips to dead vertices.

Unfortunately, death order matters in this system. If two neighboring vertices die after 1 firing and both are above the firing threshold, then the choice of which to fire first affects the final configuration; the system does not have local confluence.

To remove this dependency on firing order, I instead fire every “ready to fire” vertex at the same time. Moving chips are recorded in a cache and only added to their destination vertices at the end of each time slice. Each vertex has a lifespan that decreases by 1 with each firing. Any vertex with a lifespan of 0 is dead and becomes a wall; any chips in that vertex are kept, but no chips may be added. For all neighbors of the newly dead vertex, I decrease the graph degree by 1. A side effect of this is that dead vertices may have more chips than their graph degree if their neighbors sent them chips right before the vertex died. However, this can only occur if both neighboring vertices fire in the same time slice. While not satisfying local confluence, this method does result in a consistent final configuration.

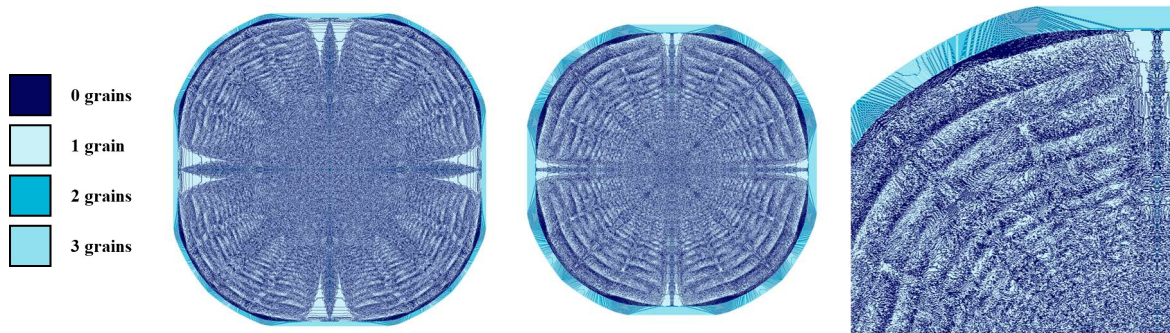


Figure 6: One million chips added at the center. Vertices die after a set number of firings. Left: Death after 512 firings. Middle: Death after 2,048 firings. Right: Detail of middle image.

Figure 6 shows the behavior of vertex death with increasing lifespans. As in the Abelian Sandpile model, a high quantity of chips is added at the center and allowed to stabilize, with vertices dying if they are fired a set number of times. In the stabilized examples above, the density of chips in the dead vertices is lower than in living vertices. Most vertices die with 0-2 chips, whereas many live vertices have 3 chips. Circular bands are visible in the dead vertices, and these become thicker as the lifespan is increased. The final configuration retains rotational and four-fold symmetry. The patterns in the outer ring of live vertices contain motifs from the outer edge of the Abelian Sandpile, while the interior is more spread out and intricate.

Summary and Conclusions

The introduction of walls offers a new tool for artistic exploration of chip-firing models. Global confluence is maintained when the graph degree does not change from the initial configuration. When introducing vertex death, the pattern converges to a unique configuration if vertex firings and deaths are handled uniformly within each time slice. This has remained consistent for all experiments in this paper, though this has yet to be formally validated. Future explorations may include study of spread at increasing death thresholds, and the combination of fixed walls and dying vertices to produce novel configurations.

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] P. Felzenswalb and C. Klivans. “Flow-Firing Processes.” *Journal of Combinatorial Theory, Series A*, vol. 177:105308, 2021.
- [3] G. Greenfield. “Generative Art from One-Dimensional Chip-Firing Automata.” *Bridges Conference Proceedings*, Linz, Austria, Jul. 16-20, 2019, pp. 115-122.
- [4] C. Klivans. “Chapter 2 – Chip-firing on Finite Graphs.” In *The Mathematics of Chip-firing*. CRC Press, 2018.
- [5] B. Mandelbrot. *The Fractal Geometry of Nature*. W. H. Freeman and Company, 1983, pp. 54–55.
- [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.
- [7] B. Trube. “Surface Toppling on Cylindrical and Polyhedral Sandpiles.” *Bridges Conference Proceedings*, Halifax, Nova Scotia, Canada, Jul. 27-31, 2023, pp. 105-112.