MATH 389: Earthquakes

Alex Cole, Ajay Pillay, Anourag Shah, John Gitterman

April 24, 2024

Abstract

We explore the Earthquake game with predetermined parameters and research routes that were chosen. The paper proposes conjectures on numerical patterns seen as the game is played out on different board sizes as well as exploring examples that helped form these conjectures. The analysis on the data is taken from model fits and tables for boards up to 20×20. Additionally, the paper analyzes the relationship between opening token placements and their end behaviour. Lastly, the paper offers a theorem that is used to calculate the capacity number, which is defined below in the paper.

1 Introduction

Consider an $n \times n$ tiled board. Each tile of the board may contain a pile of any number of tokens. For tokens to be redistributed to other tiles, the number of tokens on the tile must be greater than or equal to the number of adjacent tiles. An adjacent tile is a tile that shares an edge with the tile in question. An earthquake redistributes the tokens on each tile in the following ways:

- If the tile is on the corner and has at least 2 tokens, exactly one token is redistributed to each of its 2 adjacent sides, resulting in the tile losing exactly 2 tokens.
- If the tile is on the edge and has at least 3 tokens, exactly one token is redistributed to each of its 3 adjacent sides, resulting in the tile losing exactly 3 tokens.
- If the tile is anywhere else within the board and has at least 4 tokens, exactly one token is redistributed to each of its 4 adjacent sides, resulting in the tile losing exactly 4 tokens.

Throughout this paper, we assume that the above rules get applied to all tiles simultaneously in each time step. This process may repeat many times.

1.1 Example

Suppose we have a 3×3 board with the following initial start state:

0	**3**	**1**
0	**4**	**0**
2	**0**	**0**

Figure 1: Example start state

We can then apply the rules discussed above and result in the following sequence of transitions:

$$\begin{array}{|c|c|c|}\hline \mathbf{0} & \mathbf{3} & \mathbf{1} \\ \hline \mathbf{0} & \mathbf{4} & \mathbf{0} \\ \hline \mathbf{2} & \mathbf{0} & \mathbf{0} \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|}\hline \mathbf{1} & \mathbf{1} & \mathbf{2} \\ \hline \mathbf{2} & \mathbf{1} & \mathbf{1} \\ \hline \mathbf{0} & \mathbf{2} & \mathbf{0} \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|}\hline \mathbf{1} & \mathbf{2} & \mathbf{0} \\ \hline \mathbf{2} & \mathbf{1} & \mathbf{2} \\ \hline \mathbf{0} & \mathbf{2} & \mathbf{0} \\ \hline \end{array}$$

Figure 2: Example of transitions

At each transition, the rules get applied to all the tiles. In this example, the board reaches a state where applying the rules to each tile does not result in any of the tokens on any tile to be redistributed, hence the process ends.

1.2 Definitions

Definition 1. A **board state** is a configuration of tokens on a board, where each tile contains 0 or more tokens.

If we look back at Figure 2, we can see that there are three distinct examples of different board states. Board states can be equal to each other in the same way matrices can.

Definition 2. A **board transition** is when a given board state at time i, which we denote as B_i, undergoes all necessary token distribution in accordance with the specified rules such that it reaches a new board state at time $i+1$, which we denote as B_{i+1}.

Looking at Figure 2, we have two distinct examples of board transitions occurring between three different board states.

There are several different types of end behaviour of the earthquake process that we wish to classify. In particular, we will look at cases where the board state does not change.

Definition 3. A **stable state** is reached when the board state (ie: the amount of tokens on each tile) does not change in consecutive time steps.

5	4	5
4	5	4
5	4	5

Figure 3: Example of a stable state.

Figure 4 shows a stable state as an earthquake that would not cause any pieces to move. Since every tile has enough tokens to start shifting them out towards other tiles, any loss of tokens on tiles is cancelled out by a gain from neighbouring tiles. In a way, although it has reached a stable state, the game is not ending. The game will continue forever, cycling between the same board state turn after turn. However, this is not a behaviour that is observed in all types of stable states.

1	**0**	**1**
2	**3**	**0**
0	**0**	**0**

Figure 4: Example of a stable state.

In Figure 4 we also have a stable state, as no tile has a quantity of tokens that's greater than the number of its adjacent tiles. Yet, there is no further transition for the board to undergo, as none of its tiles can experience an "earthquake" or redistribution of its tiles. Unlike the case scene in Figure 3, the game will end once it reaches the board state scene in Figure 4. Both Figure 3 and Figure 4 are stable states, but we need further notation to distinguish between them.

Definition 4. A k**-cycle** is a pattern that the board reaches where it cycles between a minimal integer k board states. Formally, it is a sequence of board states $\{B_i, \ldots, B_{i+k}\}$ such that $B_i = B_{i+k}$ for all $i \in \mathbb{Z}_{\geq 0}$.

A k**-cycle** is a pattern that the board reaches where it cycles between some k board states. Formally, it is a sequence of board states $\{B_i, \ldots\}$ with $i \in \mathbb{Z}_{\geq 0}$ such that there exists some $j \in \mathbb{Z}_{\geq 0}$ with $j \geq i$ such that $B_j = B_{j+k}$. The choice of k here is minimal, such that the cycle does not contain some subset of states that also form a cycle.

$$
\begin{array}{|c|c|c|}\hline \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{13} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \end{array}
\rightarrow
\begin{array}{|c|c|c|}\hline \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \hline \mathbf{1} & \mathbf{9} & \mathbf{1} \\ \hline \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \hline \end{array}
\rightarrow
\begin{array}{|c|c|c|}\hline \mathbf{0} & \mathbf{2} & \mathbf{0} \\ \hline \mathbf{2} & \mathbf{5} & \mathbf{2} \\ \hline \mathbf{0} & \mathbf{2} & \mathbf{0} \\ \hline \end{array}
\rightarrow
\begin{array}{|c|c|c|}\hline \mathbf{0} & \mathbf{3} & \mathbf{0} \\ \hline \mathbf{3} & \mathbf{1} & \mathbf{3} \\ \hline \mathbf{0} & \mathbf{3} & \mathbf{0} \\ \hline \end{array}
\rightarrow
\begin{array}{|c|c|c|}\hline \mathbf{2} & \mathbf{0} & \mathbf{2} \\ \hline \mathbf{0} & \mathbf{5} & \mathbf{0} \\ \hline \mathbf{2} & \mathbf{0} & \mathbf{2} \\ \hline \end{array}
\rightarrow
\begin{array}{|c|c|c|}\hline \mathbf{0} & \mathbf{3} & \mathbf{0} \\ \hline \mathbf{3} & \mathbf{1} & \mathbf{3} \\ \hline \mathbf{0} & \mathbf{3} & \mathbf{0} \\ \hline \end{array}
$$

Figure 5: Example of a 2-cycle

Looking back at Figure 3, we can see that this game has reached both a 1-cycle and stable state. Thus, it is important to note that all 1-cycles are inherently stable states, but as shown in Figure 4, not all stable states are 1-cycles.

Definition 5. A **pure stable state** is reached when none of the conditions of the game is satisfied and the game terminates (ie: no tokens move). A pure stable state is a stable state but not vice versa.

By this definition, Figure 4 can be considered a pure stable state, as the game has terminated once it reaches its board state.

Definition 6. The **capacity number** of a board is the maximum number of tokens a board can have as its initial configuration such that it is able to reach a pure stable state.

For example, the capacity number of a 2 by 2 board is 4 tokens. This is displayed in the following example of a pure stable state on a 2 by 2 board.

1	**1**
1	**1**

Figure 6: Example of a pure stable state on a 2 by 2 board

In the above Figure, there is a way to place additional tokens on the board such that the board would be in a pure stable state, as each tile already has one less token than its number of adjacent tiles. A formal way to calculate the capacity number of a generic n-by-n board will be discussed later.

Definition 7. A **pure stable opening** is an initial placement of a board's capacity number of tokens such that the board eventually reaches a pure stable state

0	0	0
0	3	0
0	12	0

→

0	0	0
0	4	0
1	9	1

→

0	1	0
1	1	1
2	7	2

→

0	1	0
2	2	2
1	6	1

→

0	1	0
2	3	2
2	3	2

→

0	1	0
3	4	3
1	2	1

→

1	2	1
1	2	1
2	3	2

→

1	2	1
2	3	2
1	2	1

Figure 7: Example of a 1-cycle, pure stable state, and pure stable opening

Figure 7 shows the process terminating as we have the same table after every earthquake that happens after the final iteration displayed. The pure stable state is the final board state of this process. The pure stable opening is the initial placement of the board.

Definition 8. The **local saturation number** (LSN) of a tile on a board is the lowest number of tokens greater than the token capacity number such that if you place them on a single tile somewhere on the board it results in the board state eventually reaching a *1-cycle* after a finite amount of time steps.

24	0	0
0	0	0
0	0	0

→

22	1	0
1	0	0
0	0	0

→ ... →

2	3	2
3	4	3
2	3	2

Figure 8: The LSN of top-left corner tile on a 3x3 board

An important note to make is that each distinct tile on the board has its own LSN which could be greater, less than or equal to the LSN of another tile on the board.

Next, we will look at definitions that can be used to characterize the LSN's of an entire board with a single value for the whole board, as opposed to values for each tile.

Definition 9. The **global saturation number** (GSN) of a board is the maximum of all the *local saturation numbers* across all tiles on that board.

Definition 10. The **minimal saturation number** (MSN) of a board is the minimum of all the *local saturation numbers* across all tiles on that board.

1.3 Conjectures

We will now present several conjectures relating MSNs with GSNs, as well as which tiles can have the MSN and GSN on a given board.

Conjecture 1. *For a given $n \times n$ board, the GSN is equal to the MSN if and only if $n \leq 3$ and otherwise the GSN is strictly greater than the MSN for $n > 3$.*

Conjecture 2. *For a given $n \times n$ board with $n > 3$, the GSN is not equal to the LSN of a corner tile.*

These conjectures have been observed to be true for board sizes up to 17 by 17, the largest board that we were able to simulate with our computers in a reasonable computation time. For further exploration of the data gathered to support Conjectures 1 and 2, please see Section 3.

The next two conjectures pertain to the behavior of the values of MSNs and GSNs and their relationship to the size of the board they are for. Again, please see Section 3 for data that is able to support these conjectures.

Conjecture 3. *The $\lim_{n\to\infty} GSN(n)/n^2$ exists where $GSN(n)$ is the GSN of an $n \times n$ board.*

Conjecture 4. *The $\lim_{n\to\infty} MSN(n)/n^2$ exists where $MSN(n)$ is the MSN of an $n \times n$ board.*

The next and final conjecture pertains to the behaviors of pure stable openings.

Conjecture 5. *For a given $n \times n$ board, all pure stable openings contain at least $n-1$ nonzero tiles. In other words, any opening arrangement of a board's capacity number of tokens that has less than $n-1$ nonzero tiles will not result in a pure stable state.*

1.4 Overview of the Paper

Our paper will explore MSN, LSN, and GSN data, analyze such data through models, discuss pure stable states and their related openings, and conclude with further topics to explore as well as the computer data we used to get our results.

In section 2 we will explore data relating to LSN, GSN, and MSN's of boards of different sizes. Moreover, we will show our graphical approximations of these values across table size.

In section 3 we will explore pure stable openings, as well as our observations on how they can be found.

In section 4 we will mention topics for further exploration on the Earthquakes problem.

Lastly, in section 5 we briefly go over our algorithms and computer aid throughout this project.

1.5 Acknowledgments

We would like to thank Professor Aaron Pixton as well as the rest of the MATH 389 staff for their mentorship and guidance throughout this project.

2 Pure Stable Openings

In order for us to further explore pure stable openings, it is important to define a rigorous method to determine the capacity number of any n by n board. There is a closed-form solution for the capacity number of a board, as will now be laid out in the following theorem.

2.1 Capacity Number Theorem

Theorem 1. *The capacity number of an $n \times n$ board is $3n^2 - 4n$.*

Proof. For any $n \times n$ board, there are exactly 4 corner tiles, $4(n-2)$ edge tiles, and $n^2 - (4 + 4n - 8)$ remaining (center) tiles. In order to find the capacity number for this board, every tile must have the maximum number of tokens before qualifying for a valid transition. For a corner tile, it can have at most 1 token. For an edge tile, it can have at most 2 tokens, and for a center tile, it can have at most 3 tokens. This results in a total of:

$$M = (1)(4) + (2)(n-2)(4) + (3)(n^2 - (4 + 4n - 8)) = 3n^2 - 4n$$

tokens. If any tile has $\geq$ its maximum allowable tokens as stated above, this would result in a valid transition. Hence, there exists a pure stable state with exactly $3n^2 - 4n$ tokens, where the tokens are placed in accordance with the initial description of the proof, and this would result in a pure stable state since no valid transitions are possible. Furthermore, if a board contains $> 3n^2 - 4n$ tokens, by the pigeonhole principle, there must exist some tile with greater than its maximum allowable tokens as stated above, and hence results in a valid transition. □

2.2 Minimizing Non-Zero Tiles for Pure Stable Openings

In the process of discovering pure stable openings, we aimed to find pure stable openings which contained as few tiles with non-zero tokens as possible in an attempt to find any discernible patterns. On 3×3 boards, we found a pure stable opening with $n - 1$ tiles containing non-zero tokens.

0	**0**	**0**
0	**3**	**0**
0	**12**	**0**

Figure 9: 3×3 pure stable opening

For a detailed depiction of how this pure stable opening reaches a pure stable state, please reference Figure 7.

Furthermore, for 4×4, 5×5, and 6×6 boards, only pure stable openings containing n or even $n + 1$ tiles with non-zero tokens were able to be found. Examples of pure stable openings are shown in Figure 10.

These findings strongly support Conjecture 5, and we strongly suspect that Conjecture 5 holds for at the minimum 4×4, 5×5, and 6×6 boards. We know that 3×3 boards support Conjecture 5, as the board is sufficiently small to allow simulation of every possible opening on the board using 15 tokens.

0	**12**	**0**	**0**
4	**0**	**0**	**0**
0	**0**	**0**	**0**
0	**0**	**8**	**8**

0	**12**	**0**	**0**	**0**
4	**0**	**0**	**0**	**0**
0	**0**	**0**	**0**	**0**
0	**0**	**8**	**8**	**0**
0	**0**	**8**	**8**	**0**

Figure 10: Further Pure stable opening examples

In the process of exploring larger options, we were unable to find any pure stable openings that used less than $n - 1$ non-zero tiles. As we used a randomized placement of the board's capacity number of tokens, the problem quickly became intractable for large board sizes due to there being many parameters needing to be varied and the random nature of token distribution.

3 Data Exploration

As we explored $n \times n$ boards of different sizes, we observed that the LSNs across the board increase overall. More importantly, we noticed that the LSNs in certain squares tended to be lower than the LSNs for other squares. We wrote a script in Python to find the LSNs for $n \times n$ boards from $n = 1$ to $n = 20$. For $n > 20$, the computation time exceeded reasonably limits despite various optimizations we made.

3.1 MSN and GSN Data By Board Size

While running all of these simulations we were able to generate the MSN and GSN data for board sizes up to $N = 20$. The data is tabulated below.

Board Size	1×1	2×2	3×3	4×4	5×5	6×6	7×7	8×8	9×9	10×10
MSN	1	8	24	54	86	126	176	246	330	420
GSN	1	8	24	55	88	144	201	270	342	435
Difference	0	0	0	1	2	18	25	24	12	15

Table 1: MSN/GSN data for $N = 1$ to $N = 10$

Board Size	11×11	12×12	13×13	14×14	$15x \times 15$	16×16	17×17
MSN	500	610	688	856	980	1128	1244
GSN	530	633	755	883	1016	1164	1318
Difference	30	23	67	27	36	36	74

Table 2: MSN/GSN data for $N = 11$ to $N = 17$

Board Size	18×18	19×19	20×20
MSN	1412	1609	1792
GSN	1484	1655	1836
Difference	72	46	44

Table 3: MSN/GSN data for $N = 18$ to $N = 20$

The table above inspired our analysis into MSN and GSN values. While there was no recognizable pattern for the difference between GSN and MSN, the only $n \times n$ boards where the GSN and LSN were equal were for $n = 1, 2, 3$. Additionally, the MSN and GSN values were strictly increasing as the board size increased.

3.2 Visual Representation of MSNs, LSNs, and GSNs

Below is the LSN data for boards of size 1×1 through 8×8 boards. Values in teal are the MSNs and values in bold are the GSNs. Bolded teal values are values that are both the MSN and GSN of a given board. Unstyled values are values that are neither an MSN or GSN and only represent the LSN for that tile.

1

8	**8**
8	**8**

24	**24**	**24**
24	**24**	**24**
24	**24**	**24**

54	**55**	**55**	54
55	54	54	**55**
55	54	54	**55**
54	**55**	**55**	54

86	**88**	**88**	**88**	86
88	**88**	87	**88**	**88**
88	87	**88**	87	**88**
88	**88**	87	**88**	**88**
86	**88**	**88**	**88**	86

Figure 11: Tables of LSNs for $N \times N$ boards for $N = 1$ to $N = 5$

126	141	**144**	**144**	141	126
141	138	141	141	138	141
144	141	138	138	141	**144**
144	141	138	138	141	**144**
141	138	141	141	138	141
126	141	**144**	**144**	141	126

190	197	196	189	196	197	190
197	186	**201**	199	**201**	186	197
196	**201**	190	199	190	**201**	196
189	199	199	176	199	199	189
196	**201**	190	199	190	**201**	196
197	186	**201**	199	**201**	186	197
190	197	196	189	196	197	190

Figure 12: Tables of LSNs for $N \times N$ boards for $N = 6$ to $N = 7$

258	265	267	269	269	267	265	258
265	264	**270**	267	267	**270**	264	265
267	**270**	246	267	267	246	**270**	267
269	267	267	260	260	267	267	269
269	267	267	260	260	267	267	269
267	**270**	246	267	267	246	**270**	267
265	264	**270**	267	267	**270**	264	265
258	265	267	269	269	267	265	258

Figure 13: Table of LSNs for an 8×8 board

3.3 Analysis of MSN Data

By inspecting the data in Section 2.1 we observe that the MSN increase for each tile on a given board size as the size of the board increases. We found that a quadratic model fits the data well.

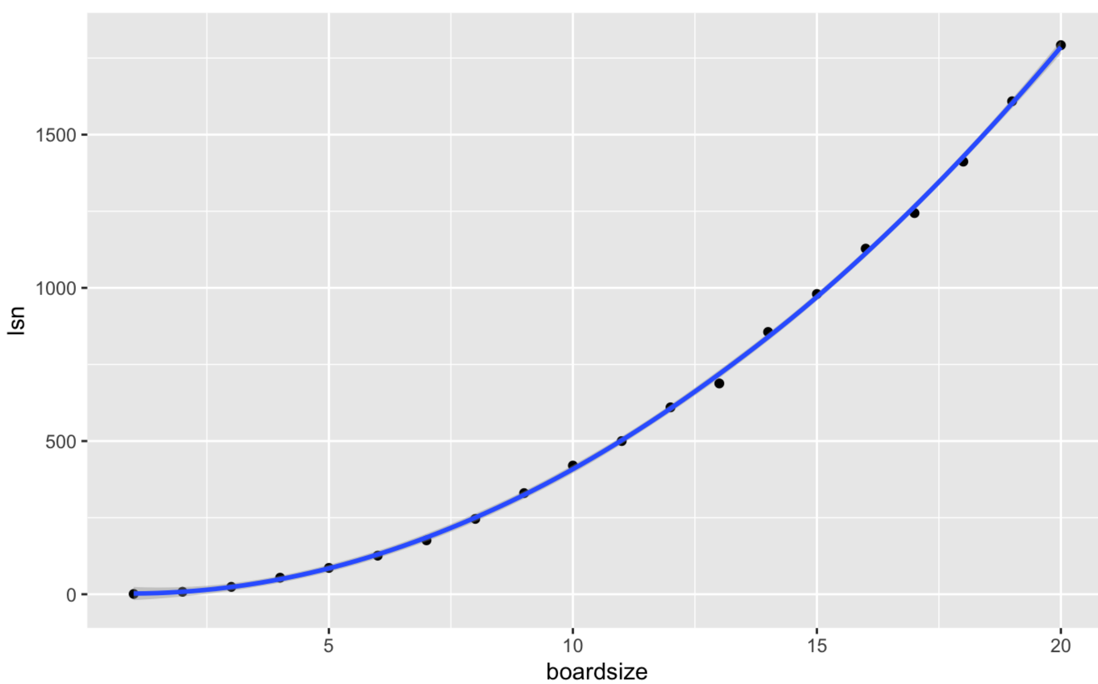

Figure 14: MSN against board size (quadratic fit)

The above figure shows the same MSN data but modeled with a quadratic fit. The fitted quadratic model is given as:

$$y = 6.30175 - 8.68462x + 4.87378x^2 \tag{1}$$

While attempting to find an appropriate model to fit, we settled on a quadratic model as we observed that this contains less of an error compared to the exponential model. By looking at the graph the model seems to fit well for all MSN points with a residual standard error of 12.59. This implies that the quadratic fit is nearly perfect and is a good predictor for finding the MSN of $n \times n$ boards with $n \geq 21$.

3.4 GSN Data

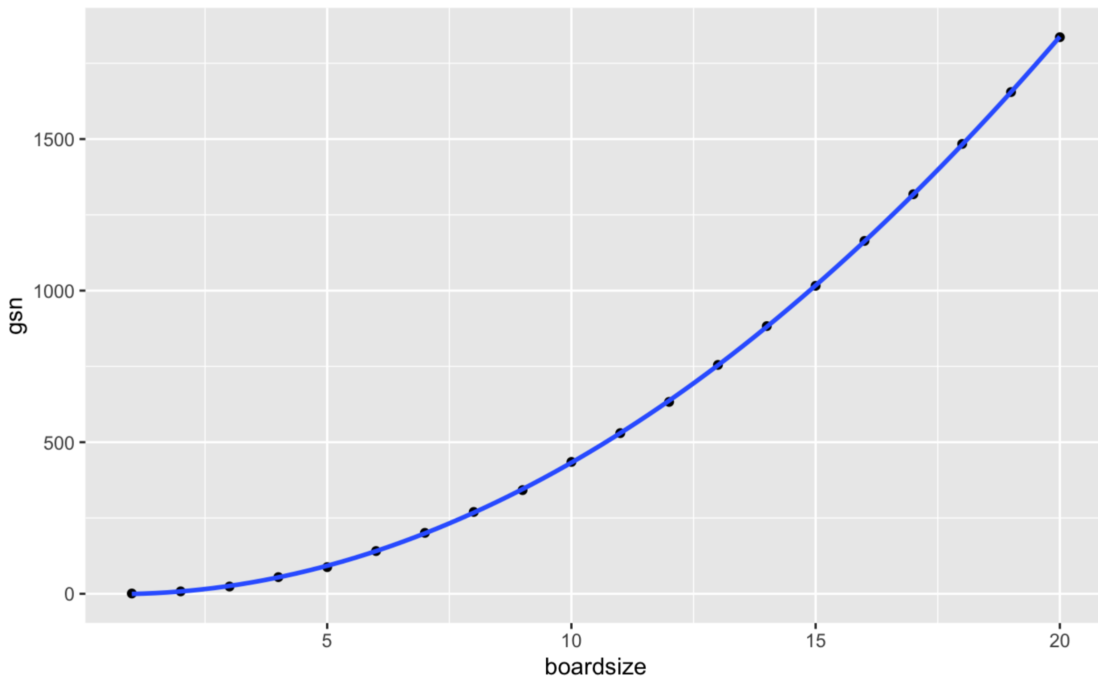

Figure 15: GSN against board size

The above figure shows the GSN with a quadratic fit. The fitted quadratic model is given as:

$$y = -2.82544 - 4.99634x + 4.85879x^2 \tag{2}$$

While attempting to find an appropriate model to fit, we found that the quadratic model is a perfect fit. This is implied as the 20 data points used, produced a residual standard error of 2.95. Thus, the quadratic fit is an exceptional predictor of the GSN value for a $n \times n$ board where $n \geq 21$.

4 Further Explorations

- Can we realize this problem in terms of $n \times n$ integer-valued matrices? What kind of states have well-defined linear transformations (matrices) between them?

There were some initial explorations to determine if board transitions could be defined in terms of linear transformations. They did exist for some transitions, but not all transitions. We pose this question to motivate ideas to formulate this problem in terms of matrices in a different, perhaps non-obvious way.

- Does the optimal number of pieces to reach a stable state that covers the board increase as we increase the stable state number?

  In general, we believe that this should necessarily be true.

- Is there a pattern in terms of how the GSN increases as the size of the board increases?
- If the GSN is applied randomly across the board, will it still reach a stable state?
- Does there exist a board and starting position of pieces that does not approach a *k-cycle* or a *stable state*?
- How can the maximum number of tokens be placed such that the board reaches a pure stable state? Is there a pattern between how these tokens can be placed based on the size of the board?
- Is there a reliable way to reverse engineer moves/turns on the board? What conditions would allow us to obtain a unique previous board state?

  This would help in being able to discover new pure stable openings, and hence allow for the generalization of previously discussed conjectures.

5 Appendix

5.1 Computer Aid

As part of our initial explorations, we wrote a script in Python to simulate the Earthquake process given an initial board state, of any $n \times n$ board.

We built off this script to generate the LSNs for each tile, which allowed us to obtain the GSNs and MSNs for each board size up to $N = 20$. We made several optimizations along the way, such as exploiting the symmetry of the problem to only compute the upper left quadrant of LSNs and reflecting them across both axes to obtain the LSNs of all tiles. We also experimented with various lower bounds as we observed that the new boards always had an MSN which was strictly greater than the GSNs of smaller boards, and hence computations did not always have to start at small numbers for each tile.

We also built off this script to simulate boards to find true stable openings. These modifications allow us to dictate how many tiles we wish to have for the simulated opening. For larger boards, the process of finding these pure stable openings has grown to be extremely computationally difficult.

One potential improvement that could be made in the future is to parallelize the script across multiple CPUs to reduce the overall computation time.

Looking at the figures and equations on LSNs and GSNs, we used R to graph and model the quadratic plots. In particular, the `lm` function produced the quadratic fits for the plots.