

# **Glider Dynamics on the Sphere: Exploring Cellular Automata on Geodesic Grids**

JEFFREY VENTRELLA  
[Jeffrey@ventrella.com](mailto:Jeffrey@ventrella.com)

## **ABSTRACT**

This paper describes the dynamics of mobile structures (gliders) in 2D cellular automata (CA) on geodesic grids. 2D CA are typically arranged on regular grids with periodic boundary conditions – equivalent to the topology of a torus. This paper describes an alternative topology – the sphere, with an underlying agenda to better understand natural closed systems. The positive curvature of the sphere as manifested in geodesic grids is described as a rich environment for CA. The necessary grid discontinuities are accepted as integral components of the environment. They are not considered as defects but rather as environmental features to be exploited. To explore the potential for a uniquely spherical computational space, a novel XOR gate built on Conway’s Game of Life is demonstrated, utilizing the double-crossing of glider paths following geodesic great circles.

*Key words:* glider, geodesic grid, sphere curvature, topology, torus, logical gate, XOR, polyhedra, irregular grid

## **1 INTRODUCTION**

Cellular automata (CA) are used for modeling many kinds of biological processes. Some CA rules can give rise to complex dynamics, self-replicating patterns, and the primary components of Boolean logic. CA-based models sometimes accompany discussion on the origins of life [1]. CA are typically modeled on 1D, 2D, or 3D grids. To avoid boundary artifacts, it is common to create periodic boundary conditions – so that opposing boundaries wrap around. In the case of a 2D cellular automaton on a rectangular grid, like Conway’s Game of Life (GOL) [2], this periodic boundary creates the topological equivalent of a torus. The advantage in doing this has nothing to do with the donut shape per se: simply that it allows the dynamics to be uninterrupted across boundaries. A 2D grid can be mapped onto a torus without introducing cell neighborhood discontinuities. A 3D torus mapping introduces geometrical distortion, but no *topological* artifacts – no discontinuities in grid connectivity. This is not the case when mapping a grid onto a sphere.

For any two points on a sphere, if they are set to motion and follow a straight path, staying on the surface of the sphere, they will come back to their original starting points. But more importantly: the two paths will intersect - *twice*. This is because all lines drawn on a sphere are actually geodesic arcs that close to form great circles. All great circles intersect all other great circles. Euclid’s parallel postulate works fine when you’re drawing figures in the sand, but not if you are considering large trajectories on a sphere, where curvature matters.

Let us celebrate the curvature of the sphere, and learn what happens when we model CA dynamics on this wonderfully non-Euclidean surface. CA implemented in non-Euclidean spaces, such as hyperbolic

surfaces, has been studied in depth recently [3][4][5][6]. It is sometimes useful when modeling real-world biological systems or urban modeling in GIS to use irregular grids [7][8][9][10][11]. Besides real-world modeling, the artificial life agenda is not fundamentally concerned with building models with perfect regularity and predictability. Sometimes it is desirable to have a model with irregular environments, and to explore the ways that emergent structures (such as gliders) evolve and adapt within these environments.

CA models that utilize periodic boundary conditions on a rectangular patch implicitly simulate the torus as an *object* – the particular shape that affords this kind of periodicity. This is often not considered relevant to the subject at hand. But in the case of the present exploration, it is the main subject. It grounds the notion of periodicity as a meaningful property of the environment. With this in mind, the sphere was chosen as the *object* of study, along with all the disruptions in grid continuity that come with it.

Given a CA rule that supports gliders with complex interactions, as in Wolfram’s Class IV CA dynamics [12], and given the discontinuities that result from mapping a regular grid onto a sphere, what new and different dynamics emerge?

### 1.1 Grid Distortion

Any implementation of a CA requires a grid: a regular array of cells of equal shape and size – but more important than geometrical regularity is topological regularity. The square grid in the Cartesian plane is used for GOL, and employs the 9-cell Moore neighborhood. Hexagonal grids have been studied as well, in which there are six neighbors surrounding each cell. As long as a Euclidean plane is used, this connectivity-scheme will have no discontinuities. The sphere changes that. Figure 1 shows four examples of how the curvature of the sphere introduces distortions in the topology of a grid, concentrated at “pinch-points”.

The top example shows a square grid stretched onto the sphere, resulting in two pinch-points located at the north and south poles – the entire top and bottom edges of the grid are each reduced to a single point. The other examples show the generation of geodesic grids based on the three triangle-faced Platonic solids. Grid distortion from spherical curvature is distributed to the vertices, corresponding to the number of pinch-points  $P$ . Grid connectivity  $G$  is 6 (based on the hexagonal grid). The number of neighbors per pinch point  $V$  is equal to polyhedral vertex valence. The value  $C$  represents the amount of distortion per pinch-point. It normalizes to 2 when multiplied by  $P$  in all three examples. One implication from this comparison is that for implementing geodesic CA, there is a trade-off between fewer pinch-points with greater distortion per pinch-point vs. more pinch-points with less distortion per pinch-point. This is related to the concept of *balance* in tiling [13]. Observe also that the only possible regular spherical tilings (having no pinch-points) are represented by the five Platonic solids. But these have far too few tiles (i.e., cells) for useful glider dynamics to emerge.

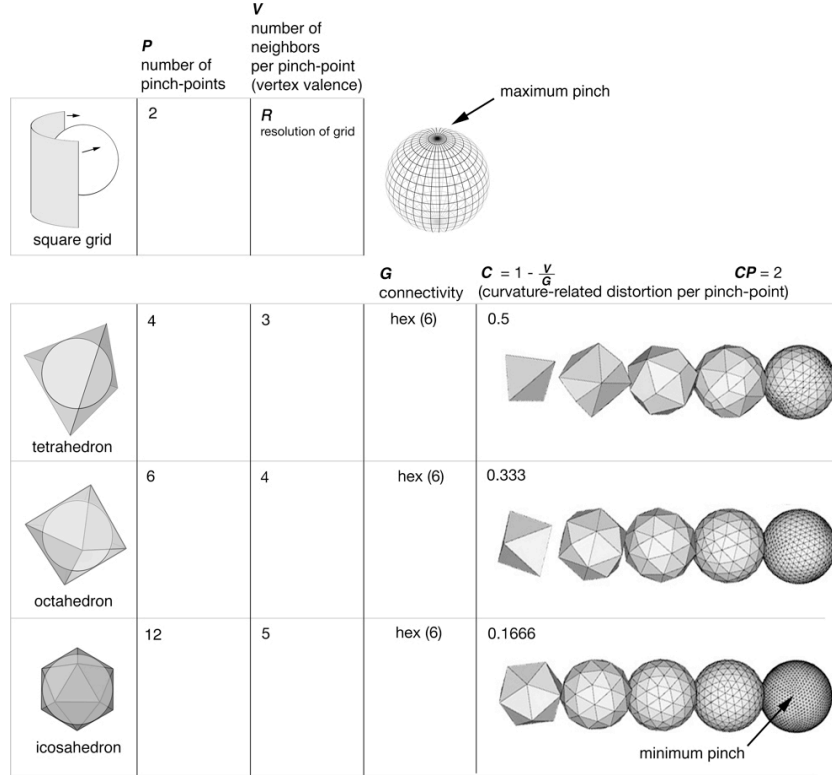


FIGURE 1  
Comparisons of geodesic grids and distortions to grid topology at pinch-points

## 2 ICOSAHEDRAL MODEL

The progression of the icosahedral geodesic at the bottom of Figure 1 is now described. The technique is borrowed from an interactive animation developed by the author to celebrate Earth Day [14]. To generate the grid, twelve points are arranged on the unit sphere in a regular icosahedral pattern. Then a method for iteratively increasing the geodesic frequency is applied, whereby every pair of neighboring points gives birth to a new point lying in-between the pair, increasing the number of points to 42. This process is then repeated five times to make a total of 10242 points. This process is equivalent to the recursive subdivision of the triangles of an icosahedron to generate a geodesic dome. The number of pinch-points (twelve) remains the same no matter how many subdivisions are applied.

Cell neighborhoods are calculated based on proximity of points after each step of increasing the geodesic frequency, as follows: the shortest distance between any two points is calculated, and this is multiplied by a number slightly greater than 1 to determine a distance threshold. For each point, if another point is closer than this threshold, it is tagged as a neighbor (this technique works equally well for irregular, randomized grids, using an arbitrary constant as the threshold). For each geodesic subdivision, the local neighborhood of each cell is stored in an array. This is used when applying the cellular automata rules.

Connecting the dots reveals the triangular grid at the bottom of Figure 1. The inverse of the triangular grid (the *dual* polyhedron) consists of all hexagonal faces, except for twelve, which are pentagons, as shown in the right sphere in Figure 2. In this illustration a lower-frequency geodesic was chosen so that the cells are more legible. The cells of the CA are best

visualized as these hexagonal/pentagonal tiles. Near the top of the left sphere is a pinch-point. The corresponding pentagon can be seen on the right sphere – it is one of the twelve cells with five neighbors.

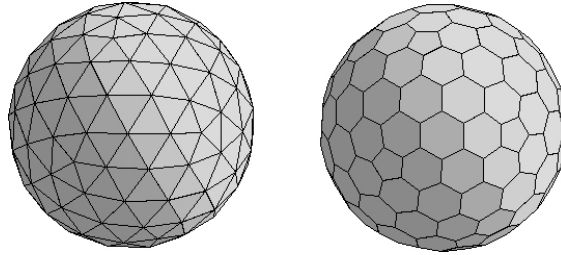


FIGURE 2

A triangulated geodesic grid and its dual polyhedron (the cells of the CA).

Similar techniques for icosahedral grid generation have been developed for geographical mapping and climate modeling, dating as far back as 1968 [15], and recently becoming more sophisticated, including schemes for generating hierarchical partitioning of hexagonal cells [16].

### 2.3 Evolving Rules That Support Gliders

The CA rules determine how the cells change their states from one time step to the next, as a function of their neighborhood states. The rules in this experiment are evolved using a variation of the genetic algorithm (GA) borrowed from a previous research project by the author [17]. This scheme permits an arbitrary number of states. A variety of interesting glider dynamics can emerge when the number of states is greater than 2.

A cell can assume any state in the range from 0 to the number of possible states  $K$ . State 0 is the *quiescent* state. At each time step  $t$  the state of each cell can be changed to another state at time  $t+1$  according to the transition function. The transition function for a given cell is described as follows: first, the cell's state at time  $t+1$  (its next state) is set to quiescence as a default. Then,  $R$  "subrules" are applied in sequence. A single sub-rule can be expressed as follows: the cell's state at time  $t$  (its current state) is compared to a reference state (the *referenceState*). If the cell's state does not match *referenceState*, then the sub-rule is not applied. Otherwise, the sub-rule compares the number of neighbors having a specific state (the *neighborState*) to a specific number (the *neighborCount*). If there is a match, then the sub-rule sets the cell's state at time  $t+1$  (its next state) to a specific result state (the *resultState*). Thus, four integer parameters are used for each subrule. They are *genetic*, i.e., they can have a range of possible values, and can be changed by a genetic operator. The ranges are:

1. *referenceState* can be any state in  $K$
2. *neighborState* can be any state in  $K$
3. *neighborCount* can be any number from 0 to  $M$
4. *resultState* can be any state in  $K$

In this experiment,  $M=3$ ,  $K=4$ , and  $R=20$ . The number of genes for the transition function  $KR$  is thus 80. The set of all genes for a transition function is referred to as the *gene array*. The pseudocode below illustrates how the sub-rules are applied to determine the value of the next cell state.

```

set next cell state to quiescent

for each subrule
  if current cell state equals subrule.referenceState then
    if number of neighbors of state subrule.neighborState
    equals subrule.neighborCount then
      set next cell state to subrule.resultState
    endif
  endif
end for

```

The gene array specifies a single individual in a population evolved using the GA. The GA uses an interactive evolution component – a human provides the fitness values while viewing CA dynamics. The population is initialized with random gene values, and then a tournament scheme is used: for each iteration, two individuals with statistically high fitness values are chosen to mate and generate an offspring, using crossover and mutation. This individual replaces the least-fit individual in the population. It is then viewed and given a fitness value. The purpose of this interactive scheme is to find gliders (and proto-gliders), using human vision – a very efficient system for identifying objects and tracking motion against a chaotic background.

One rule for a 4-state CA was found using the technique that generates gliders that exhibit annihilation, reproduction, and reflection (kick-back) upon colliding with each other. The 80 genes (arranged to show the 20 subrules) are shown here:

<b>1223</b>	<b>0013</b>	1223	1012	0213	1110	<b>0313</b>	0010	2211	<b>1103</b>
2110	0320	1102	2120	0313	2023	0100	2303	3111	1310

The purpose of this scheme is not elegance or optimal performance, but evolvability – hence the seemingly large number of subrules. Some of these sub-rules are redundant, and some of them over-write the states of previously-applied sub-rules. For instance, the first subrule (1223) is redundant with the third, and so it can be eliminated. Also, the result of the second subrule (state 3) will always be over-written by the result of the eighth subrule (state 0), and so it can be eliminated. Subrules indicated by solid gray or rectangle outlines can be eliminated.

A typical period 2 glider that emerges is shown in Figure 3. It is moving downward and to the right. Black cells show state 1 and gray cells show state 3. While this traveling glider only uses these two states, glider collisions evoke state 2 and thus more subrules. Analogous to real organisms, some genes are only expressed under certain circumstances.



FIGURE 3  
A 2-state period-2 glider that emerges from the above rule.

### 3 EXPERIMENTS

Figure 4 shows gliders that emerge from the rule described above as they interact on two grids shortly after the CA is initialized with random states. On the left is an array of 10241 (133x77) cells arranged with hexagonal connectivity, with periodic boundary conditions and so it corresponds to the torus topology. This has one less than the number of cells in the

corresponding geodesic grid (which is 10242, as determined by its geodesic frequency) – close enough to make reasonable comparisons between the two grids. Both of these grids are topologically equivalent at the local scale, with the exception of the locations of the twelve pinch-points in the geodesic grid.

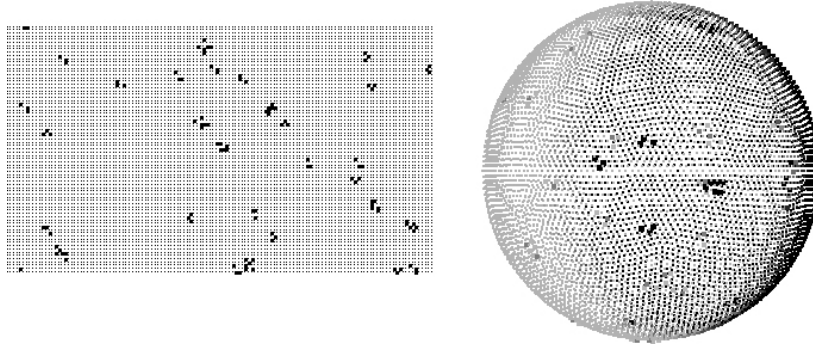


FIGURE 4  
A rectangular domain with 10241 (133x77) cells (left); and a geodesic grid with 10242 cells (right). Both show gliders after being initialized with random states.

One experiment was done to compare the rates at which the dynamics settle to a relatively stable, mostly-quiescent state after random initialization. Results did not show a notable difference in dynamics over time between the torus and the sphere. It was suspected that there may be differences in the way gliders interact. If a CA rule supports gliders that exist against a mostly quiescent background, and if those gliders live long lives and travel great distances, then there is more opportunity for them to experience the impact of spherical curvature and thus interact with each other differently than if they were traveling on the torus.

To test the global effect of sphere curvature on glider movement, seven gliders were initialized in a vertical row on the grid. Figure 5 shows, in three stages, the paths traced by these gliders as they move leftward. An icosahedron is superimposed onto the grid to indicate the locations of the pinch-points. As this example shows, the curvature of the sphere causes some gliders to collide – specifically near the top, where one of the pinch-points exists. Like a gravitational field, one can see the effect on the paths of the two top gliders, they swerve downward, and collide with the other gliders, causing a cascade of more collisions. The glider at the bottom nearly escapes the effects of a pinch-point, and is able to follow the same general direction as the gliders immediately above it.

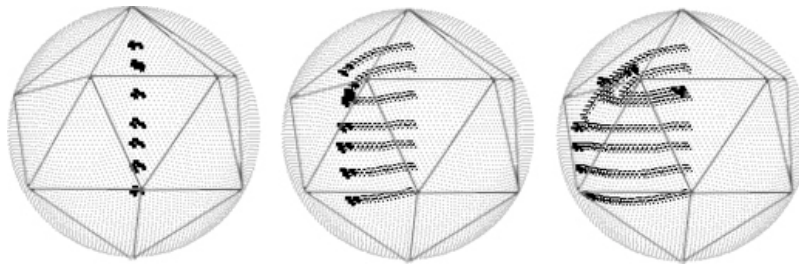


FIGURE 5  
Glidars moving leftward pass over a vertex and experience collisions

Another way to explain this collision is by considering that the icosahedron has six strips of regular grid connectivity that wrap around the six equatorial bands. This is shown in Figure 6. The strips overlap such that each face is visited by three such overlapping strips. The collisions shown in Figure 5 are the result of gliders moving along two of these strips, which overlap on the triangular face at the left.

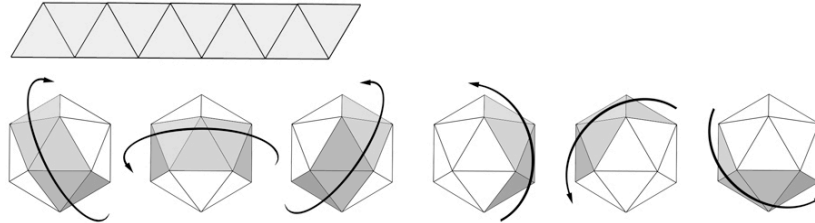


Figure 6. Six overlapping equatorial bands with regular grid connectivity on the icosahedron

### 3.1 Signals on Spheres

Are there uniquely spherical CA computational machines? Logic gates have been simulated in CA using glider guns and other still-life's and short-period structures which, when arranged in just the right way, can simulate the primary elements of computation, such as AND, NOT, and OR gates. A Turing Machine was built using patterns of GOL [18]. These are typically implemented on 1D or 2D grids. Imagine a CA with logic gates that not only rely on periodicity (as a torus affords) but also on a uniquely spherical kind of periodicity.

The sphere shown in Figure 7a illustrates GOL implemented on a unit cube and then projected onto a sphere. Unlike the triangulated geodesic grids shown above, this uses a square grid – six of which are sewn together such that the CA dynamics carries over across the twelve edges. Projection onto the sphere does not introduce any changes in grid connectivity – it only introduces geometrical distortion. Gosper's glider gun is shown on the top face in 7b. The path of gliders generated by the gun moves diagonally, and therefore visits all the faces of the cube (and unless it is dealt with on another face, it will come back around and destroy the gun!). Imagine alternatively a continual stream of gliders (no gun) and a gun in another location on the cube sending a perpendicular signal of gliders that interact with this stream. Or imagine four continuous glider streams with no guns, as illustrated in 7c. This wasn't tested, but it is suspected that there are spherical implications to what emerges. These signal paths correspond to the edges of the cuboctahedron (7d), which can be realized as four intersecting great circles (7e).

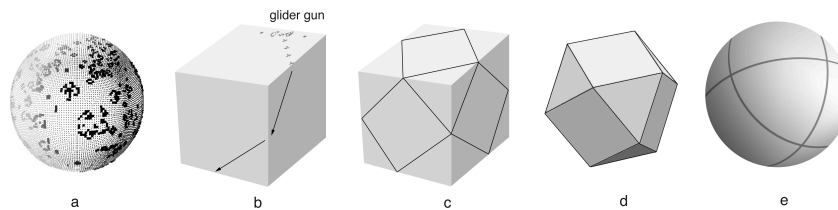


Figure 7. (a) Game of Life implemented on a cube and projected onto a sphere. (b) glider gun signal crossing midpoints of cube edges; (c) overlapping glider streams; (d) cuboctahedron; (e) there are four cuboctahedron great circles – each with six crossings.

### 3.2 A Spherical XOR Gate

To explore the effect of spherical curvature on CA computation, an XOR gate was implemented based on the geometry shown in Figure 7. It employs similar techniques as demonstrated by Rendell [19]. This is illustrated in Figure 8, showing GOL with 60x60 cells per cube-face (21600 total cells). On the left is a view of the two inputs, and on the right is a view from the opposite side, showing the output. Two period 30 Gosper guns are used to generate the glider streams for inputs A and B. These streams are terminated by two eaters on the opposite side of the sphere.

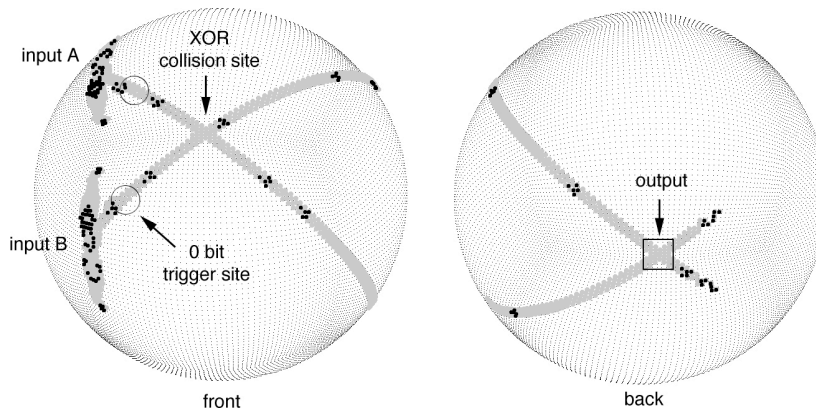


Figure 8. An XOR gate using the Game of Life implemented on a sphere takes advantage of crossings of geodesic great circles which provides a single output location.

Glider streams represent the one's and zero's of a binary signal, and the absence of a glider in the stream represents zero. A zero value is created by destroying a glider in the stream soon after leaving the gun. The site for this action is shown in Figure 8 as the "0 bit trigger site". The input clock value  $c = \text{time step } t \bmod 30$  (the period of the gun). A new value is fed to the input stream every  $c = 0$ . If the input signal calls for a zero, the cell at the trigger site is set to true ("on", or "1") and causes the glider to dissolve within a few time steps. If the input signal calls for a 1, the glider is allowed to pass through. If gliders pass through from both inputs A and B, then they collide and annihilate each other at the XOR collision site.

The binary output value is determined by reading the state of a specifically-chosen cell at the output site when  $C = 10$ . This cell is on only when a glider is passing over it, and so the output is 1. This cell detects the passage of a glider from either stream, as shown. With this implementation, there are two critical crossover points, corresponding to the intersection points of two great circles. The first one is responsible for the XOR collision, and the second one is responsible for reading the output.

### 3.3 Exploiting Grid Discontinuity

The above experiment specifically avoids the eight pinch-points of the cube. As Figure 9 shows, if a glider passes over one of these points, it may dissolve or get converted into two blinkers and a six-cell block (which normally changes to a beehive, but because of the abnormal neighborhood at the pinch-point, it remains as a still-life).

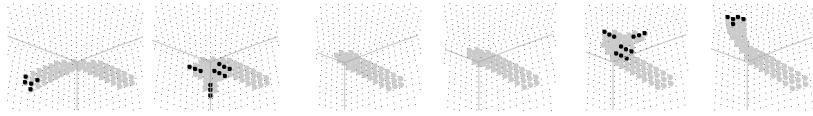


Figure 9. Game of Life gliders passing over a cubical vertex

In this illustration, each glider is offset vertically by one cell. The first and last examples in the illustration show gliders escaping the outlaw nature of the pinch-point and veering off, following the grain of the grid.

A CA machine engineered to live on a geodesic grid could take advantage of these regions as termination points for gliders, or other computational purposes. These would normally be considered *defects* littering an otherwise regular grid. But it is well-known that material defects can be catalysts for complex and characteristic growth, such as with crystals. Evolutionary simulations for studying emergent complex behavior might discover emergent CA structures that exploit these regions.

Imperfections, boundary artifacts, and errors in physical simulation are routinely exploited by artificial organisms, and they can reveal something about adaptability and emergent complexity within environments with arbitrary features.

#### 4 CONCLUSION

*Sphereness* will be experienced by an agent to varying degrees depending on how much of the sphere the agent traverses in a given event. A sailor on the ocean looking through a pair of binoculars will not be able to see a distant ship until it is close enough to rise above the horizon, due to the curvature of Earth. Two ants fighting on one side of an apple may give up and part ways, only to find each other again on the other side of the apple, having traveled only half-way around. On a local scale, and safe from any pinch-points, CA structures will experience the same dynamics as with any regular planar grid. When CA dynamics evolves around the vicinity of a pinch-point, the behavior is different. But is this difference unique to spheres? No, because the grid discontinuities of a pinch-point could be implemented on a plane. It is only when the geodesic grid as a whole system comes into affect that the true nature of spheres makes its impact on the CA.

The mobile structures we are discovering and designing are just the beginning of a fascinating journey as increased computing power affords more complex CA, with more states, larger neighborhoods, more detailed rules, higher dimensions, and more complex grid topologies. This paper describes a cursory exploration into geodesic geometry, and how this environment affects glider behaviors on a global scale. It is an exploration into the nature of *gliderhood* juxtaposed against *spherehood*. There may exist CA machines which can be built more efficiently using the artifacts of geodesic grids. It may suggest solutions to problems that either cannot be solved in the plane, or else come out more efficient, elegant, or aesthetic, on a geodesic grid.

#### REFERENCES

- [1] Langon, C. Life at the Edge of Chaos. Artificial Life. Addison Wesley, 1992.
- [2] Berlekamp, E. R.; Conway, John Horton; Guy, R.K. (2001 2004), Winning Ways for your Mathematical Plays (2nd ed.), A K Peters Ltd.

- [3] Margenstern., M. Cellular Automata in Hyperbolic Spaces. Old City Publishing, 2007.
- [4] Yukita, S. Cellular automata in non-euclidean spaces. Proceedings of the 7th WSEAS International Conference on Mathematical Methods and Computational Techniques In Electrical Engineering. Pages 200-207. World Scientific and Engineering Academy and Society. 2005
- [5] Kiestler, R.A., and Sahr, K. Planar and spherical hierarchical, multi-resolution cellular automata. Computers, environment and urban systems. Elsevier. 2008
- [6] Yukita, S. Dynamics of cellular automata on groups. IEICE Trans. Inf. & Syst., vol. E82-D, no. 10, pp. 1316-1323, 1999.
- [7] Flache, A., Hegselmann R. Do Irregular Grids make a Difference? Relaxing the Spatial Regularity Assumption in Cellular Models of Social Dynamics. Journal of Artificial Societies and Social Simulation vol. 4, no. 4. 2001
- [8] O'Donoghue, Diarmuid P. and Mullally, Emma-Claire. Extending Irregular Cellular Automata with Geometric Proportional Analogies. Proceedings of the Geographical Information Science Research UK Conference, 11th - 13th April 2007, NUI Maynooth, Ireland. 2007
- [9] Stevens, D, Dragičević S, 2007, "A GIS-based irregular cellular automata model of land-use change" Environment and Planning B: Planning and Design 34(4) 708 – 724
- [10] Johnston, P., Kelso, J., Milne, G. J. Efficient simulation of wildfire spread on an irregular grid. International Journal of Wildland Fire. 2008
- [11] O'Sullivan, D. Exploring Spatial Process Dynamics Using Irregular Cellular Automaton Models. Geographical Analysis, 2001.
- [12] Wolfram, S. Universality and Complexity in Cellular Automata. 1984
- [13] Grunbaum, B. and Shephard, G. C. Tilings and Patterns. W. H. Freeman and Company, 1987 ("balanced tilings explained in pages 129-134).
- [14] Ventrella, J. Earth Day 2009 – A Spherical Cellular Automaton (web site published online at <http://www.ventrella.com/EarthDay/EarthDay.html> 2009
- [15] Sadourny, R., et al. Integration of the Nondivergent Barotropic Vorticity Equation with an Icosahedral-Hexagonal Grid for the Sphere. Monthly Weather Review. Article: pp. 351–356
- [16] Sahr, K., White, D., Kimerling, A. J. Geodesic discrete global grid systems. Cartography and Geographic Information Science. American Congress on Surveying & Mapping. 2003
- [17] Ventrella, J. Gliders and Riders: A Particle Swarm Selects for Coherent Space-Time Structures in Evolving Cellular Automata. Stigmergic Optimization. Springer, 2006.
- [18] Rendell, P. Turing Universality of the Game of Life. Collision-Based Computing, ed. Adamatzky, Springer, 2002
- [19] Rennard, J. P. Implementation of Logical Functions in the Game of Life. Collision-Based Computing, ed. Adamatzky, Springer, 2002

