

Milestone 5 Report

Slime Team A: Abraham Hmiel, Eric Brooking, Matthew Thies, Ke Li

December 13, 2006

1 Introduction

Physarum Polycephalum is a large unicellular amoeboid organism that grows in the form of a thin sheet with wiry tubules. Numerous tests and experiments have been performed for years that show that slime mold has the capacity to move predictably and intelligently to maximize the efficiency of its nutrient distribution network. *Physarum Polycephalum* is remarkable for many reasons. It can find the shortest route through a maze [2] [5], all of its nuclei divide at the same time [7], the ectoplasm streams inside its tubes in an oscillatory manner [1] [7], it searches for food and avoids danger [3] [6], and produces an efficient network of communication within a closed environment [4] [5] [8]. The overall objective for this project is to create an efficient local model for the slime mold resource distribution network based on mathematical theory and our research.

Resource distribution networks play a major role in many fields including telecommunications, urban planning, nanotechnology, and shipping commerce. In all cases, finding an efficient (maximal cost-effectiveness) and robust (resilience to perturbations or unexpected events) network connecting all its constituents to the necessary resources is of prime importance.

We claim that through our efforts with the slime mold *Physarum Polycephalum*, it is possible to make a workable model for a generalized resource distribution network. We present a computer simulation of the time evolution of a petri dish with some initial slime cells and some food sources placed on it. Our model uses functions based on local rules that can simulate a colony as it grows and attempts to find food sources. We do this by considering two real physical processes: the conservation of mass and hydrostatic pressure.

2 The Problem

The problem we will attempt to answer is how a slime mold in a petri dish will organize itself when a certain arrangement of food sources are placed down upon it and the initial position of a small portion of the slime is known (The initial slime is grafted onto this petri dish from another more mature colony). We want to make a local model that uses hydrostatic pressure and the conservation of mass as primary rules for how the slime

forages for food. We are more interested in the foraging behavior than the long-term “choice” of most efficient tubes.

2.1 Modeling

We use Matlab version 7.0.4 for our computations. Our main simulation program is called `Slime.m`, with submodules `countnn.m`, `shift.m`, `maketubes.m`, and `look.m`. In our model we consider a square box as a model of the petri dish, with the origin at the top left. The box is subdivided into a number of cells. For the simulations that are in this document, the box is 42 sites in the x-direction and 42 sites in the y-direction. The structure we use to represent the universe of the slime is a rank 3 matrix, where the dimensions are y-sites, x-sites, and time steps. At each cell is a scalar value that represents the content of that cell:

Cell Value S	Description
$S = -2$	Food source not occupied by slime
$S = -1$	Food source occupied by slime
$S = 0$	Free space (agar gel)
$1 \geq S > 0$	Slime nucleus

A Slime nucleus’s thickness is therefore on a gauge between 0 and 1, with 1 being the maximum thickness that is only possible at the very beginning of a simulation for reasons that will be explained later. The thickness is roughly equivalent to the ‘yellowness’ of a small region of a real slime experiment. Connected food sources are considered to have a thickness of 1; their representation on the matrix as -1 is only to identify them as different than slime cells with thickness 1 that are not connected to a food source.

We use the conservation of mass principle to stipulate that slime cannot be born out of thin air; when a slime cell divides to conquer one empty cell, it must split its thickness between the existing cell and the new cell. The ratio of new cell thickness to old cell thickness is parameterized, but we keep it at 0.5 for simplicity (so that a cell splits its thickness evenly between new cell and old cell when it moves). The direction in which to add a new cell is chosen by the slime randomly among those cells with the highest pressure difference. For a thorough discussion of this procedure, see sections 2.4.5 and 2.4.6.

In addition to the slime moving its thickness (mass) from occupied cells to unoccupied cells, it can also shift its thickness from occupied cell to occupied cell based on the difference in pressure between cells. Slime thickness will always flow from high to low pressures (down-gradient). The mechanism by which a cell does this is explained later in sections 2.4.5 and 2.4.8.

2.2 Changes to the Algorithm Since Last Milestone

- The values of some of the parameters were changed to get better-looking results (namely, *tubetol*)

- The nature of the shifting algorithm was changed so that shifting among tubes and non-tubes follows different rules.
- Code was added so that a food source and tubes could be deleted from the matrix at a certain time, mimicking our ability in the lab to remove a food source and watch the changes that take place in the way the slime reorganizes itself.

2.3 Assumptions

Many assumptions are made in our simplified local model.

- First, we assume that the primary factors affecting the slime's organization are the positions of the food resources and the positions and thicknesses of slime cells.
- We assume that the thickness of a slime cell in our matrix is proportional to its mass, and therefore since the two factors scale with each other, we can use the general terms mass and thickness interchangeably.
- We assume that the mass of the slime is conserved except for zones very close to occupied food sources where food is transformed entirely into slime and then passed throughout the colony.
- We assume the time scale we use for the slime's movement is much longer than one period of sol-streaming oscillations, therefore, we neglect sol-streaming behavior.
- Temperature, calcium ion concentration, the rate at which slime nuclei can digest food per nucleus, and also waste production are all ignored.
- The atmospheric pressure is chosen to be 0, which is the value of the pressure of an empty cell or unoccupied food cell.
- The slime behaves on a local level and each nucleus can only be affected by its nearest neighbors.
- We assume that the direction for adding a slime cell is chosen randomly among those directions that satisfy the pressure and tube conditions.
- We assume that the importance of nearest-neighbors in swarming is high. Generally, our experimentation and modeling for both groups in class solicit the fact that swarms of autonomous animals do what others nearby them are doing. However, this assumption may have its disadvantages because it is hard to determine what a nearest-neighbor of one unit of slime in a petri dish is.
- We assume the slime never makes fruiting bodies or changes phase; it perpetually remains in the plasmodium state.

2.4 Techniques

We employ a variety of different programming techniques for our simulation of the time-dependent behavior and prediction of the tube structure of a slime mold colony. We want the model to be as accurate as possible, so there are many factors we have to consider. We use computer modeling because it would be extremely time-consuming to calculate the growth of a slime mold colony by hand.

2.4.1 Parameter estimation

Two parameters we will attempt to estimate are distance and time. In the lab, the petri dish is about 3.4 inches in diameter. One square of graph paper (1/4 inch to a side) is modeled by a 3×3 matrix, so each cell in the overall slime matrix is $\frac{1}{12}$ of an inch and we use a matrix that is 42×42

Time is tougher to estimate because our algorithm evolves by a number of integer timesteps and not a value of time, but it is possible to estimate the length of a time step by comparing our groups's pictures on the repository. For a full discussion of how we calculate the length of a time step, refer to section 4.1. According to Sauer [7], a mitotic cycle of *physraum polycephalum* is about eight hours.

The rest of the parameters we use in our model (twonnchance, shifting parameter, etc.) are chosen to: first, prevent crashes and unnecessarily long computations, and second, to give results that are in general agreement with what we see in the lab. Here is a description of the most important parameters we use in our model:

Parameter	Description	Value
twonnchance	Probability of add if cell has two nearest neighbors	0.40
threennchance	Probability of add if cell has three nearest neighbors	0.20
fournnchance	Probability of add if cell has four nearest neighbors	0.12
β	Stiffness coefficient	$e^{-0.05R_{cm}}$
split	Fraction of old cell added to unoccupied cell	0.5
shiftamt	Fraction of difference in cell pressure equalized	0.5
tubetol	Minimum amount of add that will form a tube	0.08
addlimit	Minimum possible amount of add	0.002
susfactor	Mass of slime that 1 occupied food source supports	8
S_b	Basal thickness	0
xsteps	Extent of matrix in x-direction	42
ysteps	Extent of matrix in y-direction	42

2.4.2 Hexagon Lattice

To better represent a real-life slime mold, our simulation takes place on a hexagonal lattice represented by a square matrix. This is done by redefining the nearest neighbors of a cell to include two extra neighbors (6 in the hexagon lattice as opposed to 4 for a square lattice). This gives the slime mold another degree of freedom in movement

and mitigates the annoying effect of the square lattice that a diagonal path has the same length as an L-shaped path that extends in one direction for a certain number of cells, and then turns to run perpendicular to that path until the endpoint is reached. A pictorial description of the indexing of nearest-neighbor cells follows in Figure 1, with the ii denoting the column and jj denoting the row:

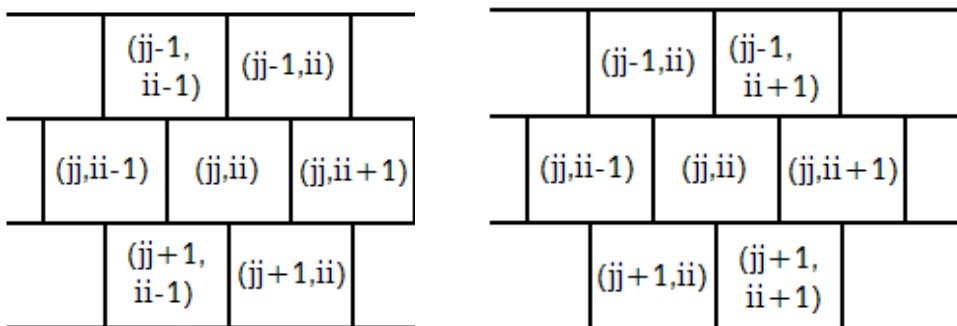


Figure 1: Left: Odd row indexing. Right: Even row indexing

2.4.3 Initial Arrangement

An initial arrangement consists of a matrix of mostly zeroes with food sources and initial slime cells present. Initial slime cells in our trials have a thickness of 1, but in general they can be any number S where $0 < S \leq 1$, or, they can be connected food sources ($S = -1$) if one wanted to start with slime underneath a food source. The initial position of slime is important because it will try to forage for food in places nearby first.

2.4.4 Counting Nearest Neighbors

For each timestep, our algorithm checks over the whole slime matrix. When it encounters a cell that is not either 0 or -2 , it counts the number of cells that are not 0 or -2 that are adjacent to it. This value determines what chance a cell has to add a new cell and whether or not it can shift some of its thickness to a neighbor with lower pressure. Surely, if a cell only has one nearest-neighbor, it is on the frontier of the network. Such a situation will always cause the cell to divide. If there are more than one nearest neighbors, the chance of adding a new cell to an unoccupied cell is governed by the parameters $twonnchance$, $threennchance$, and $fournnchance$. The probability of an add with that many neighbors is equal to the parameter itself. We have observed that, in general, increasing the three parameters makes the slime distribution more diffuse and cover more cells, and lowering those parameters make the distribution more sinewy and thin, and forces the slime colony to take more timesteps to find food.

2.4.5 Calculating Pressure

We use the model developed by Kobayashi [1] to determine the pressure at every cell once per timestep near the beginning. The pressure is given by:

$$p(\mathbf{r}, t) = \beta \frac{S - S_b}{\bar{S}} \quad (1)$$

β is a stiffness coefficient given by the pressure when the nondimensional thickness S is increased from S_b by \bar{S} . \bar{S} is the mean thickness of the entire organism. For our purposes, we equate β with a decreasing function of the distance to a cell to the center of mass of the colony:

$$\beta = e^{-0.05R_{cm}} \quad (2)$$

Where R_{cm} is the distance from the cell to the center of mass. Also from the Kobayashi paper, we have the equation:

$$S_b = \bar{S}(1 - a \cos \theta) \quad (3)$$

which calculates the basal thickness of the slime with theta being a phase field. Since we are not taking into account the oscillatory streaming of the sol but instead a hydrostatic approach, we can approximate S_b by averaging its over one cycle, which is simply \bar{S} . Subsequently, since all pressure cells are modified by $\frac{-S_b}{\bar{S}} = -1$, we redefine the atmospheric pressure to be 0, not -1 . It is the relative amount of pressure that matters for our model to calculate the difference in pressure, not the absolute pressure. Finally, our model equation for pressure is simply:

$$p(\mathbf{r}, t) = \beta \frac{|S^*|}{\bar{S}} \quad (4)$$

Where

$$S^* = S, S \neq -2 \quad (5)$$

$$S^* = 0, S = 2 \quad (6)$$

The absolute value accounts for the possibility of the occupied food source slime cell and S^* treats the unoccupied food source cell as 0. The gradient of p which determines the direction of mass flow has the discretized form:

$$-\nabla p = \text{maximum} = \frac{\Delta p}{\Delta r} = p_{jj,ii} - \text{minimum}(p_{neighbor}) \quad (7)$$

Where Δr will always be a constant equal to unity, for it is the distance between two neighboring cells for any of the six neighbor cells.

2.4.6 Moving to Unoccupied Cells

To split a cell into to a nearest-neighbor empty space or food source, the slime at a particular cell, if the chance parameters allow, picks out a direction randomly chosen from those that have the greatest difference in pressure with the splitting cell. If a

food source is nearby (within a one-cell radius), the command given will be to go in the direction of a nearby food source, approximating the 'feelers' of a real slime mold. The proportion of the cell that remains to that which existed before the split is chosen to be 0.5 for simplicity.

2.4.7 Tubules

When a cell divides, it creates a tube linking the two cells together in both cells if the add amount is above the parameter *tubetol*. The tube direction is given by a set of prime numbers corresponding to the six nearest-neighbor directions, and if more than one tube exists for a cell, these numbers are multiplied together. This makes it possible, by factoring, to find all the tube connections for the whole organism. If tubes exist in a cell that is marked for shifting, the shift can only occur in directions where the tubes lead. If no tubes exist for either cell, the shift may proceed in any direction corresponding to the greatest pressure difference.

2.4.8 Shifting Between Cells

During each time-step, after the total pressure matrix is calculated, each cell with more than just one nearest-neighbor will shift some of its mass to other cells if the pressure is lower there. We reason based upon our model so far that a slime cell with one nearest neighbor will never have its neighbor at a lower pressure if the splitting parameter is 0.5 or less. The shift is in the direction of the greatest pressure difference among slime cells, and the amount of the shift is proportional to the difference between the cells' pressures.

Using this approximation, an occupied food source slime cell (with a thickness of 1 but S of -1) will keep shoveling out mass to its nearest neighbor cells and those cells will pass nutrients along the line. This is especially true when the slime colony spreads out greatly and a very thin tube finally connects to the food source. The pressure difference between the thin, spread-out slime and the occupied food source slime will be very high, prompting a deluge of mass in the reverse direction that the slime took to get to the food source.

A schematic of how the look and shift functions actually work follows in Figures 2.4.8-2.4.8. In the actual program all of the initial cells may move, but we are simplifying it to illustrate the process for a few cells at a time:

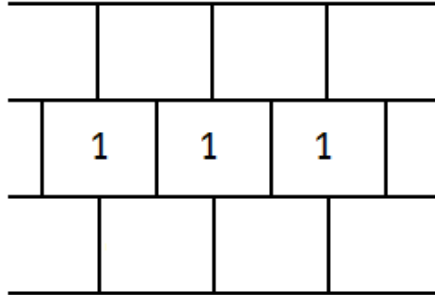


Figure 2: A slime arrangement at a certain time. S is shown. In empty cells, $S = 0$

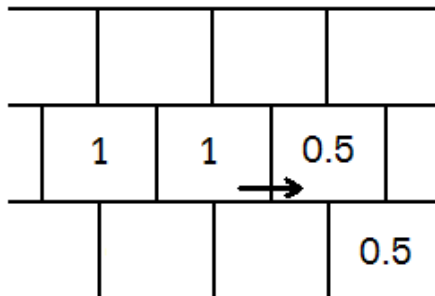


Figure 3: The slime arrangement after the rightmost cell moves in a randomly chosen direction. The arrow shows the direction of greatest pressure decrease for the middle cell. The splitting parameter is 0.5

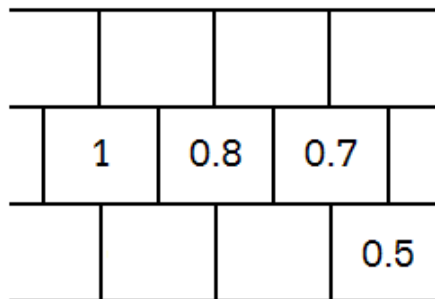


Figure 4: The slime arrangement after the shift function is called for the middle cell. The shift proceeds in the direction given by the arrow in the preceding figure and the magnitude of the shift is proportional to the difference in pressure between the two cells, with shifting parameter = 0.4. The total mass of the cells is still the same as it was in the initial layout.

3 Results

We present initial and final mass tables for a colony simulation, as well as pressure diagrams for intermediate timesteps. It is easier to see the slime overall using the log of pressure diagrams.

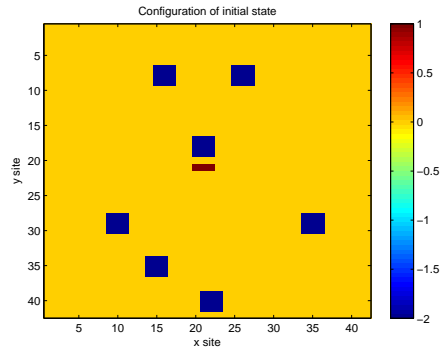


Figure 5: Initial state of slime (red) and food (blue)

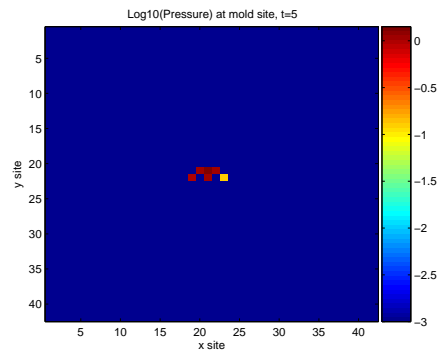


Figure 6: Log10 of cell pressure at t=5 (7.563 hours)

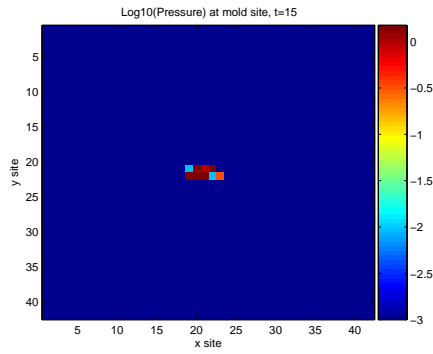


Figure 7: Log10 of cell pressure at t=15 (22.689 hours)

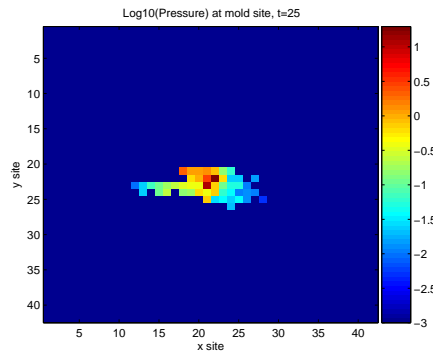


Figure 8: Log10 of cell pressure at t=25 (37.815 hours)

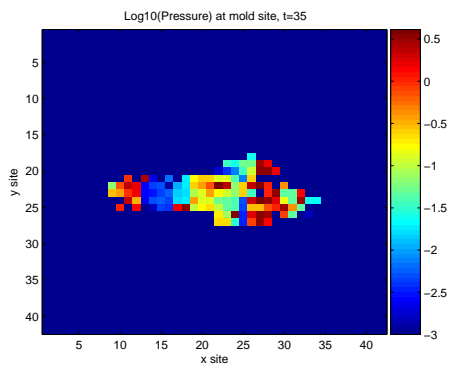


Figure 9: Log10 of cell pressure at t=35 (52.941 hours)

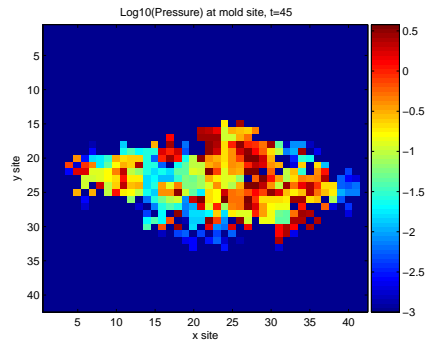


Figure 10: Log10 of cell pressure at $t=45$ (68.067 hours)

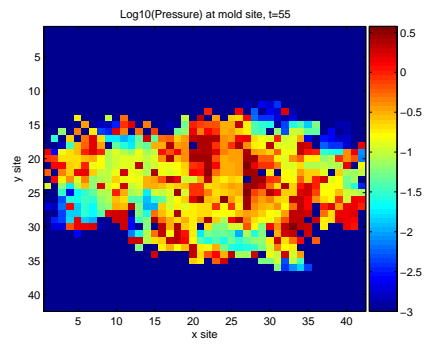


Figure 11: Log10 of cell pressure at $t=55$ (83.193 hours)

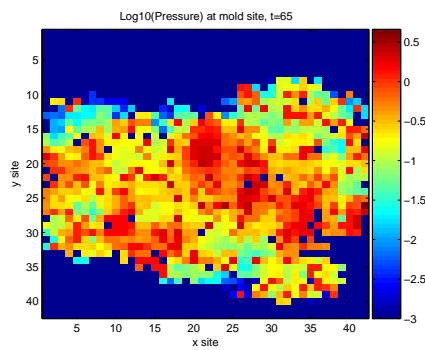


Figure 12: Log10 of cell pressure at $t=65$ (98.319 hours)

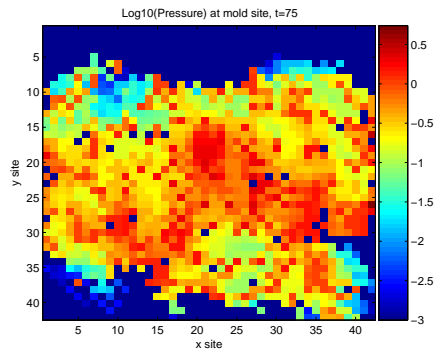


Figure 13: Log10 of cell pressure at $t=75$ (113.445 hours)

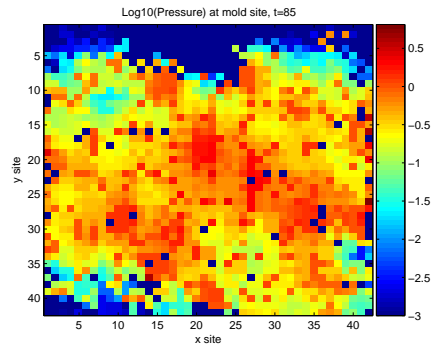


Figure 14: Log10 of cell pressure at $t=85$ (128.571 hours)

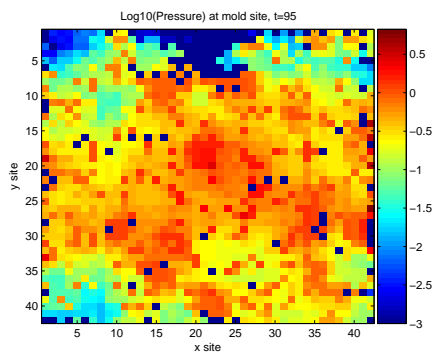


Figure 15: Log10 of cell pressure at $t=95$ (143.697 hours)

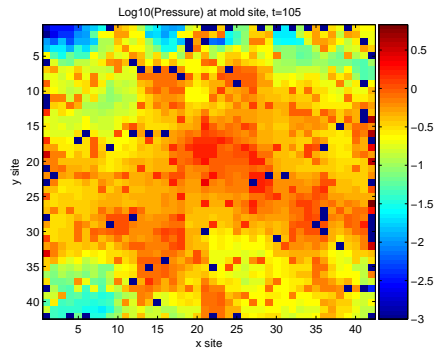


Figure 16: Log10 of cell pressure at $t=105$ (158.823 hours)

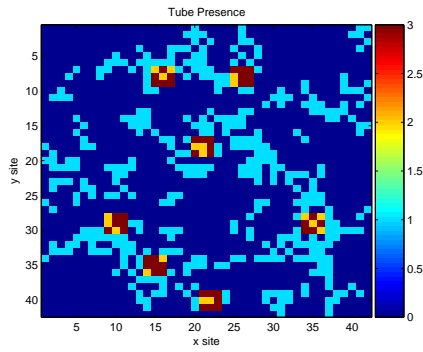


Figure 17: Tubes Present in Overall Structure at Finish

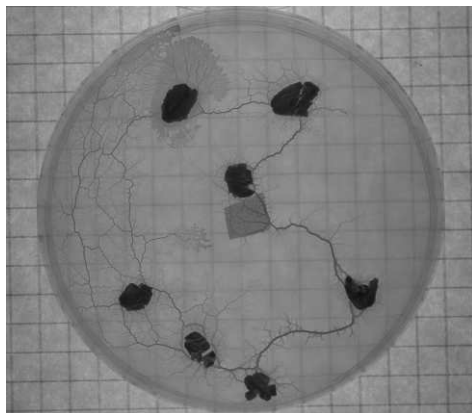


Figure 18: Snapshot of Lab-Grown Colony at $T=96$ hours

4 Analysis

The original configuration of slime for one simulation is depicted in Figure 5, plotting the slime’s thickness at each site. The food sources are the blue squares and the slime cells at the onset are the red squares. Most of the rest of the graphs are shaded logarithmically with pressure so that it is easier to see the contrast between different cells; If we plotted the slime thickness only a few cells would be differentiable from the agar medium.

In the first few pictures, one can see how the slime will grow radially outwards from its initial position in search of food. When it encounters a food source, it will produce a lot of slime very rapidly (see figures 3 and 3) and move out to find more food. The “fanning” behavior of the slime mold is ably duplicated by our model while the time steps remain small.

Because the pressure scale is logarithmic, for most snapshots of the slime growth, the only cells that would be visible to the human eye in a real tabletop experiment would be yellow-orange to red. One cannot see the tubules in the pressure matrix, but one can guess where they might be by the strong presence of orange-to-red cells that signify high pressure and thus good likelihood of high thickness.

4.1 Convergent Limit

One item that has discouraged our group in the past is that our model definitely seems to work better for systems of certain size than other sizes. When the number of cells in the grid changes, the slime still behaves by nearest-neighbor rules, and can only move by one cell at a time. This indirectly changes the length of one single unit of time in our algorithm because we consider that the slime always takes about six days (144 hours) to fully explore a petri dish. We wanted to investigate the relative changes in the value of the timestep for different system sizes and also see if we could obtain an agreement with the continuous model of [8] for very small cell sizes.

Several data points of number of time steps required to “fill” the grid with slime are plotted in figure 4.1. Only square systems were studied. The curve-fit is linear, and is given by:

$$N_{steps} = 1.413 * L + 35.8 \tag{8}$$

Where N_{steps} is the number of time steps required for the mold to fill the grid and L is the edge length of the system in unit cells. When considered that it takes about 144 hours to fill the grid in real-life, the expression for the length of a time step, T_{step} , as a function of the edge length, L , is:

$$T_{step} \approx \frac{144}{\sqrt{2L + 25.8}} \tag{9}$$

These expressions are only valid for systems with edge length between 5 and 131 units. Systems outside this bound will crash our algorithm. For the 42×42 systems we have previously discussed, the length of a time step is about 1.513 hours.

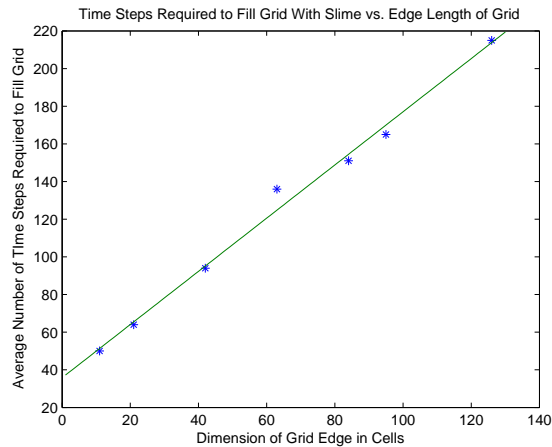


Figure 19: Result of several trials of how long it takes to fill a grid with slime, with linear curve-fit superimposed.

Unfortunately, as the size of the system grows, our prediction of where the tubules of the slime mold lie become very poor. The tubes become more or less randomly dispersed and cease to look like the pictures of the mold in the lab. For instance, a 126×126 system is shown with the tube structures at the final time step in figure 4.1. Comparison to figure 3 is quite disparate.

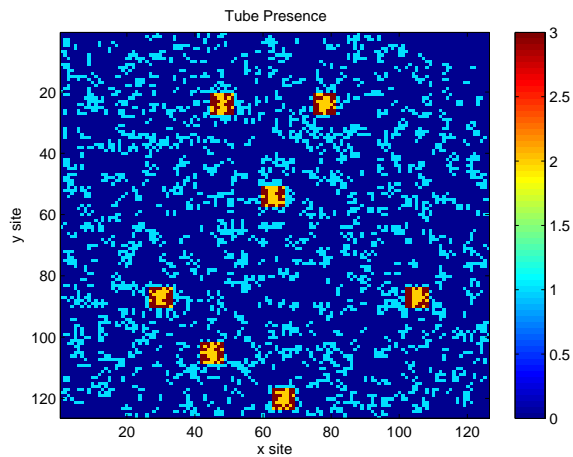


Figure 20: Result of tube structures of a 126 by 126 colony.

4.2 Image Processing

We have created an algorithm that highlights the tube structure of a picture of a slime mold colony from the lab. With the Matlab image processing toolbox and some clever programming, the original pixel values of the slime cells inside the tubules do not change,

while agar pixels, food sources (which may be overlaid with a program in development), and even most lines on a piece of graph paper underneath the petri dish are ignored. When viewed in the HSV colormap, the presence of blue lines in the modified image might even seem to suggest dying tubules while brighter greens and yellows imply that the tubules are thriving

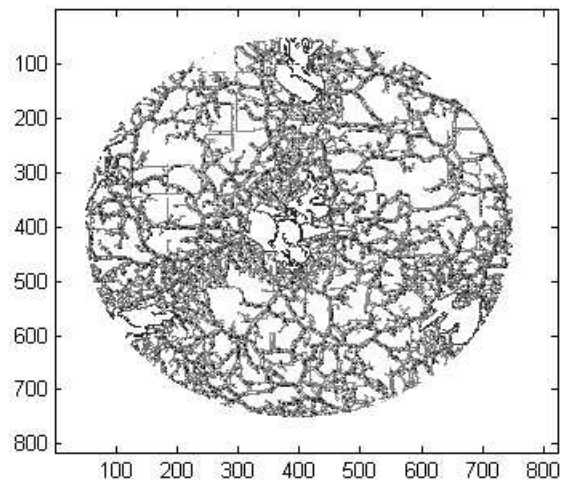
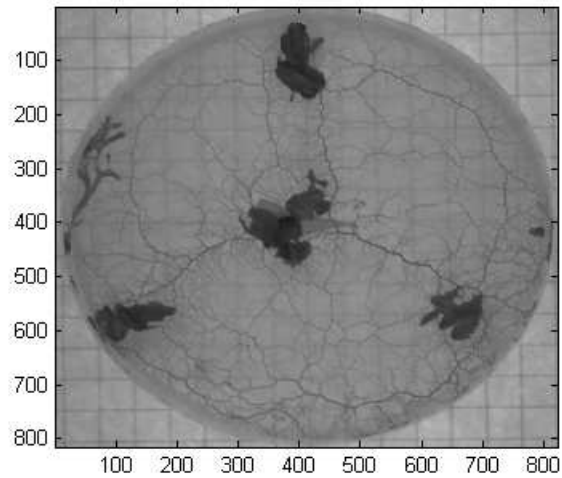


Figure 21: Result of edge detection algorithm on a picture of a slime arrangement (forgive the strange appearance)

5 Strengths and Weaknesses

- There are two anomalies that we use in our code that are inconsistent with building a *strictly local* model. The first is that the whole colony knows what the allowed maximum mass of the colony is based on the number of connected food sources. This information is conducted instantaneously across the whole network. To make a better local model, there should be some other kind of way that knowledge about how much food there is available gets transmitted. The other nonlocal structure in our programming is the calculation of the mean mass of the organism, which appears in the pressure calculation. To make this more localized, perhaps the slime should only count the average mass of slime cells in its vicinity, perhaps a 5×5 matrix. This would be relatively easy to implement, but will undoubtedly stretch out the computation time.
- Another limitation is that our model does not allow slime cells on the border of the matrix to move or shift, because our functions routinely address neighboring indices of a specific cell. If a cell on the left boundary, called, for example, nearest-neighbor counting routine, the program would crash because it would be unable to access the matrix element(jj,0). For a 42×42 matrix, 162 cells lie on the edges, out of 1764 total cells. About one-tenth of the total cells being faulty is a statistically significant problem, but increasing the number of sites on the matrix mitigates this effect. Also, the nearest-neighbor counting may be changed to allow cells on the border to access the routine and count neighbors inside the matrix.
- Besides those shortcomings, our model's success is hindered by the fact that many parameters are set in the beginning and do not change, like `twonnchance`, `threennchance`, and `fournnchance`. Perhaps a way to create better-looking tubes is to have these parameters change over time so that two nearest neighbors eventually fail to create new slime (a straight, infinite pipe of slime has two nearest neighbors at any point). Along the same lines, it neglects the S_b parameter and phase field that is present in Kobayashi's model [1].
- Our implementation of a line of code that would delete a food source and the tubes surrounding it at a certain time did not work very well. What it did, in effect, was delete the tubes and slime, but at the next time step, more slime would just run in and repopulate the region, the only difference being there no longer being a connected food source there but still a strong slime presence. For this reason, our solutions to the other groups' milestone challenges
- Overall, our steady-state solution of what the slime will look like after a large amount of timesteps is fairly good: the majority of the tubes lie close to food sources and others branch out to surrounding regions. However, there is still no routine for deletion or reduction of undesirable slime cells based on flux maximization.
- The image-processing program we have written to take a picture of slime from the lab and emphasize the tubes on it works very well, but we have not taken the next

step as to using it to calculate which tubes would be diminished and which would thicken by applying equations or another program to the created images.

- Looking past the limitations, it is clear that our simplified local model provides a good basis for modeling the slime’s foraging behavior and a predictor of major tubules.

As a local model, our computer code could theoretically be able to take an existing colony with networks, receive input as to a change or evolution within its environment, and predict with an experimentally possible outcome. Once the code predicts correct behaviors, further revision of the code will be able to account for large scale events through comparison with laboratory tests, allowing for continued development of the model. Running the model multiple times is also not computationally intensive for the user. This is very important because as an organism of study with very microscopic and exact interactions, it is very difficult to measure quantitatively what exactly is occurring with each nucleus. The real organism is not necessarily impacted by random events, but impacted by events which are too difficult to measure. This is different from computer networks for example. Computers have very interpretable features. A broad model, such as what we are developing, could be applied to computer networks with these detailed features included. In this sense, what we are hoping to accomplish has very definite applications outside of simply slime.

6 Summary and Conclusions

Our model provides a good approximation of how the slime grows and makes tubes. However, after a while, the colony will reach a steady-state behavior and no true optimization of the resource-distribution network will occur. Our model is therefore best at predicting initial foraging behavior and tube formation and should be reworked to give better predictions for long-term optimizing behavior.

7 Future Work

The biggest task we have in front of us is maintaining unity in our model over varying phases of the organism’s life cycle as they are integrated. Our model may not be 100% accurate in its current form, but it has enough flexibility to be tuned as needed. The best models are ones which keep the same structure while only changing the value of certain variables used in the behavior and not operating in a piecemeal construction. This looks promising in regard to slime probing and fruiting bodies.

Under our current assumptions, the model does not look at calcium ions nor flux whatsoever. With further research, we can assess implementing these as perhaps a third and fourth layer to our matrices construction affecting the probability of division and nutrient transportation.

One direction that is possible would be implementing image processing directly into our cellular model. This will allow us to identify the status of current slime cultures and with ease run simulations on their development.

If we can fix all the limitations of our program, we are confident that our ability to model long-term behavior of the slime will improve. One thing that we have not rectified is our model's reduction to the sol-streaming model of Tero [8] for small cell sizes and timesteps because our system can only go so far into that limit without crashing.

References

- [1] R. Kobayashi, A. Tero, and T. Nakagaki. Mathematical model for rhythmic protoplasmic movement in the true slime mold. *JOURNAL OF MATHEMATICAL BIOLOGY*, 53(2):273–286, August 2006.
- [2] T. Nakagaki. Smart behavior of true slime mold in a labyrinth. *RESEARCH IN MICROBIOLOGY*, 152(9):767–770, November 2001.
- [3] T. Nakagaki, R. Kobayashi, Y. Nishiura, and T. Ueda. Obtaining multiple separate food sources: behavioural intelligence in the physarum plasmodium. *PROCEEDINGS OF THE ROYAL SOCIETY OF LONDON SERIES B-BIOLOGICAL SCIENCES*, 271(1554):2305–2310, November 2004.
- [4] T. Nakagaki, H. Yamada, and M. Hara. Smart network solutions in an amoeboid organism. *BIOPHYSICAL CHEMISTRY*, 107(1):1–5, January 2004.
- [5] T. Nakagaki, H. Yamada, and A. Toth. Maze-solving by an amoeboid organism. *NATURE*, 407(6803):470–470, September 2000.
- [6] T. Nakagaki, H. Yamada, and A. Toth. Path finding by tube morphogenesis in an amoeboid organism. *BIOPHYSICAL CHEMISTRY*, 92(1-2):47–52, August 2001.
- [7] Helmut W. Sauer. *Developmental Biology of Physarum*. Cambridge University Press, 1982.
- [8] A. Tero, R. Kobayashi, and T. Nakagaki. Physarum solver: A biologically inspired method of road-network navigation. *PHYSICA A-STATISTICAL MECHANICS AND ITS APPLICATIONS*, 363(1):115–119, April 2006.