

# Paper Title

Student Name (1st Author)  
School of Student (1st Author)

Student Name (2nd Author)  
School of Student (2nd Author)

URECA Supervisor Name  
School of Supervisor

Collaborating Supervisor Name  
School or Organisation of Collaborating Supervisor

**Abstract.** Self-assembly pervades nature, from the coalescence of iron oxide nanoparticles to the collective motion of flocks of birds. To understand self-assembly, we devised a set of simple local rules to simulate the swarming of Western honey bees (*Apis mellifera*), thereby giving rise to a spectrum of interesting emergent states characterised by their morphology and behavior. Parameterised by two variables, a *crawl rate* (how fast bees attached to the wooden board advance towards the queen) and a *climb rate* (how vigorously bees at the peripheral of the structure attempt to scale the inverted mount), we were able to yield structures akin to those recorded in the physical world. These findings were then corroborated with empirical data established in the literature. Our simulation reveals that at lower-density regions, a diagonally-centered square lattice is the prevalent structure, while at higher-density regions, this lattice breaks down into an amorphous mesh of bees with a higher degree of connectivity.

**Keywords:** self-assembly, simulation, physics

## 1 INTRODUCTION

Nature is complex, even seemingly simple problems such as the double pendulum have proven near intractable. Yet within this chaos, scientists have always found ways and means to quantify and model this universe. The best formulations are not necessarily the most complex ones. Consider Newton's laws of motion or Maxwell's equations of electromagnetism. Despite their compactness, they've proven useful time after time again, suggesting that Occam's razor still prevails. In this paper, we'll be exploring the collective motion of Western honey bees (*Apis mellifera*). In the same spirit, we hope to develop a set of simple rules to describe this fascinating phenomena.

During reproduction, there are four distinct phases that honey bees go through [5]. The first is the swarm exodus phase where they leave their current nest, followed by the migration phase where they initiate their search for a new site. When a suitable site is found, the bees form a temporary cluster. This inverted mount is known

as a **bivouac**, and is the subject of study in this paper. Finally, when an alternative has been found, they enter the final phase (swarm liftoff) where they disassemble the structure and head towards the new location.

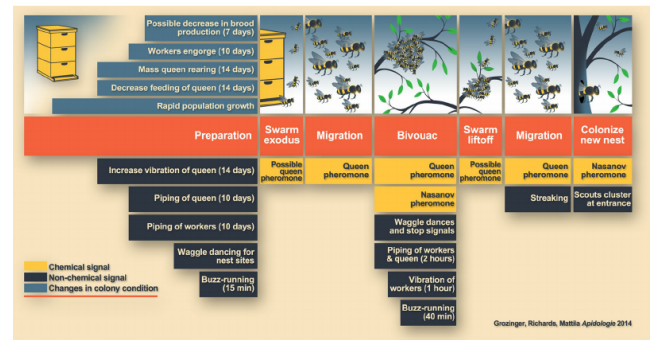


Figure 1: An overview of the reproduction phases [5]



Figure 2: Bivouacs in nature [8]

There have been several attempts in the literature that aim to demystify self-assembly in Western honey bees. At the exodus phase, [3] explores how several factors such as the colony size and the worker age distribution may be incorporated into a holistic model that predicts when swarming is triggered. On the contrary, [7] argues that swarming is initiated not by the queen, but by the individual decisions of thousands of bees, each

trying to optimize the honey that they may contribute to the hive.

Moving on to the migration phase, [4] explores the use of 2D ODE-based models rooted in the *streaker bee hypothesis* that try and simulate the flight of honey bees towards the new nest. There are also robotics-based approaches such as *BEECLUST* that use simple qualitative principles to programatically determine the next step of each bee [10].

Unfortunately, the bivouac phase has been largely left untouched. This is unsurprising as there is little data available on honey bee bivouacs. A notable success in recent times is the establishment of a strength-mass power scaling law by [11]. However, there haven't been any attempts to establish a mathematical framework that describes this formation process, thus offering a gap that our paper aims to bridge.

## 2 METHODOLOGY

In designing these rules, we heavily referenced the timelapse video in [8] and tried to get a sensing of not only how the bivouac's overall structure morphs over time, but also how the individual bees are reacting and the general direction in which they're heading towards. It's important to stress that our focus is on *local* rules, rather than global ones, as we're keen to discover beautiful emergent states that stem from simple behavior at the individual level.

Drawing inspiration from [1] and [2] that explore self-assembly in other social insects such as *Eciton* army ants and *Solenopsis* fire ants, we also adopt a similar state machine approach, where each bee has a series of finite states that it operates within, moving in and out from one state to another depending on certain conditions, as well as exhibiting behavior/rules particular to each state.

To test the validity of these rules, we've chosen to implement them in a real-time 2D simulation using TypeScript and HTML Canvas Graphics. This allows for convenient debugging and easy access to anyone with a web browser. It's also easy to then layer graph components over the simulation using the Recharts package to understand how certain quantitative variables are evolving over the course of the simulation. We adopt the use of pixels (as defined on the Canvas plane) as the standard unit throughout this paper.

Naturally, one may think of a bee as oscillating between one of two states: *hover* or *attached*. A bee has several legs which it uses to cling onto one or more objects. As long as it's clinging to the board or another attached bee, we consider a bee attached. Note that a bee can be clinging to *both* the board and another bee, and because of this recursive definition, we

avoid the situation of a bee clinging onto another bee in flight.

In addition, there are several assumptions that we have to impose.

- Bees are treated as point masses, they may be viewed as tiny circles of radius  $r_n$ .
- Two bees may not overlap.
- A bee has a "reach", represented by a circle with an outer radius  $r_b > r_n$ .
- A bee may cling on up to *two* objects.
- A bee only has *local awareness*, i.e., it can only sense coordinates of other bees that it has bumped into, or is attached to, or that it's supporting.
- A bee has access to the queen coordinates (so that it can move towards it).
- The queen is a stationary target positioned at the center of the board.

In practice, we found that setting  $r_n = 5$  and  $r_b = 15$  produces realistic looking results.

## 3 IMPLEMENTATION

### 3.1 COORDINATE SYSTEM

As is common in web programming, we define the origin  $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$  as the top left corner of the screen. Similarly, the bottom right corner is denoted by  $\begin{pmatrix} w_c \\ h_c \end{pmatrix}$ .

Intuitively, this implies that when we have velocities where  $v_y < 0$ , we're moving *towards* the board, not away from it like you might have been used to.

### 3.2 INITIALIZATION

To mimic the setup in [8], we design a board situated at a y-coordinate of  $\frac{h_c}{3}$ . The queen spawns at  $\begin{pmatrix} w_c/2 \\ h_c/3 \end{pmatrix}$ . It remains stationary throughout the simulation and serves as an anchor for worker bees to latch onto. For realism, the queen's dimensions are enlarged by a factor of 1.75.

Upon initialization, the simulation progressively introduces new bees into the canvas. Bees spawn from either the left or right bottom corners with equal probability. We add a small position offset from the uniform distribution  $U(0, 30)$  to make it less monotonous. To decide the initial velocity, should the queen be covered, we set it to  $\vec{v} = \begin{pmatrix} U(0, 0.5) \\ -2 \end{pmatrix}$ , otherwise, we direct it towards the queen bee. The acceleration  $\vec{a}$  is always initialized at zero.

Internally, we keep track on the number of frames that have elapsed. Every 50 frames, we introduce a new

bee, up till the point where the number of bees have reached the permissible limit.

### 3.3 COLLISIONS

There are two kinds of collisions to consider, namely, a bee colliding with a board/wall and a bee colliding with another bee. The first case is easy to settle, WLOG, consider a collision with the left wall; we may check if  $p_x \leq r_b$  (note that we're considering the outer circle). It's common during the simulation for the bee to clip into the wall slightly ( $p_x < r_b$ ). In this situation we immediately reset  $p_x$  to  $r_b$  to keep the bee within bounds. In the second case, let's denote the euclidean distance between the two bees as  $d$ . One may observe that a collision has occurred when  $d \leq r_{b_1} + r_{b_2}$ .

Instead of going through every pair of bees in  $\mathcal{O}(n^2)$  to perform the collision check, we choose to trade some memory in favor of a *collision grid*, where each bee has a corresponding square in the grid. Let the side of each square be  $s$ , then the x-coordinate of the square is  $\lfloor p_x/s \rfloor$ . Memory wise, this would cost us  $\lceil h_c/s \rceil \cdot \lceil w_c/s \rceil$ . We found that  $s = 100$  gives the smoothest performance. To check if a bee is colliding with another bee, we now only need to lookup those bees in the 8 adjacent squares, as well as the current square. At the end of each frame, we'll recompute the corresponding square of each bee in  $\mathcal{O}(n)$ .

## 3.4 STATES

### 3.4.1 HOVER

In the hover state, a bee is either flying around or colliding with objects. As it flies, we'd like to manipulate its velocity in such a way so that it mimics an actual bee flying. A popular solution is to use Perlin noise [9] often used to procedurally generate terrain. Leveraging several layers of Perlin noise sampled at different frequencies and amplitudes, we may derive Fractal Brownian Motion that results in extra realism.

There are two classes of objects a hovering bee may collide with. The first is attachable objects, comprising the board and other attached bees. The second is non-attachable objects; this includes walls and other hovering bees. By *walls* we're referring to the left, right, and bottom edges of the browser screen.

To resolve the first scenario, we'll immediately attach to the object without any reservations. Should there be multiple candidates, we'll simply attach to any one. Further refinements (i.e., selecting better attachment targets) would be made in the attached state so this suffices for now.

In the second scenario, we resolve the collision using the conservation of linear momentum. Suppose a bee touches the left wall, we'll have  $\vec{v}_1 = \begin{pmatrix} -x \\ y \end{pmatrix}$ , where  $\vec{v}_0 = \begin{pmatrix} x \\ y \end{pmatrix}$ . Alternatively, if it collides with another hover-

ing bee, we'll use  $m_1 v_1 + m_2 v_2 = m_1 v'_1 + m_2 v'_2$  to solve along each axis separately.

### 3.4.2 ATTACHED

In the attached state, we begin by reconsidering our attachment targets. The point on the board that's considered for attachment is the nearest point to the bee, i.e., the bee should never be holding onto the board at a slanted angle. We then consider all other bees within collision range, and filter those that are at least 2 pixels above our current position (it's only logical to grasp onto something above us for better support). We then sort all candidates based on their Euclidean distances. Special consideration is required for the board point which is allocated a value of zero so it will always be chosen. In other words, we prioritise hooking onto the board first before placing a burden on other bees.

At this juncture, we also allow the bee to detach from the structure if it has no bees that it's supporting, i.e., no other bees are clinging onto it. Let the probability of detachment be  $p_d$ . Should this value be high, bees will detach too quickly without allowing the structure to coalesce and develop over time, i.e., you'll observe a flat uninteresting one-layer structure. After some experimentation, we found that  $p_d = 2 \cdot 10^{-7}$  best replicates what we observe in the recording.

Let's proceed to consider the force interactions that affect an attached bee. We begin by exploring what happens if we only have a single bee attached to a board. Through observation, we notice that a bee that has to support more weight is situated closer to the board, i.e., it has to pull itself closer to the board. This implies that we need to exert a force on the bee directed *upwards* in proportion to the bee's mass, let's call this force vector  $\vec{F}_m = \begin{pmatrix} 0 \\ -mg \end{pmatrix}$ , where  $g$  represents gravitational acceleration. Within the simulation, we used  $m = 1, g = 0.04$ . Higher  $g$  values produce more compressed structures that deviated too far from reality, so we opted for a low  $g$  value instead.

This force vector alone isn't sufficient in producing the desired effect as the bee will merely stick to the board. Indeed, there needs to be some *counteracting* force. Consider an arbitrary point (board/bee) that a bee is attaching to, let it be situated at  $\vec{p}_1$ . Intuitively, the closer we get to this point, the harder it tries to push us away from it. This is reminiscent of Hooke's law. Consider the unit vector pointing from the attachment point to the bee  $\hat{v}$ . Additionally, observe that the separation distance  $d_n$  at which no pushing occurs is at either  $r_b$  (bee and board) or  $r_{b_1} + r_{b_2}$  (bee and bee). Letting the actual separation distance be  $d_a$ , we have  $\vec{F}_a = k \cdot (d_n - d_a) \cdot \hat{v}$ . Through trial and error, we found that  $k = 0.02$  was most suitable.

In the same vein, let's consider the bees that one has

to support. It turns out that the same analogy holds, and we may derive a similar expression for  $\vec{F}_s$ , representing the force exerted by a supporting bee.

Although it may appear that these forces are sufficient in producing the desired effect, our preliminary testing shows that overlaps can still occur near the base of the bivouac. At this high-density zones, a bee may be colliding with many other bees, yet it may not be connected to every one of them. Therefore, let us impose yet another force, coined  $\vec{F}_r$  that acts a repulsive force handling these miscellaneous bees. For simplicity, let this be equal to the unit vector pointing from the miscellaneous bee to our current bee.

Thus, the total force imposed on a bee is:

$$\vec{F} = \vec{F}_m + \sum_i \vec{F}_{a_i} + \sum_j \vec{F}_{s_j} + \sum_k \vec{F}_{r_k} \quad (1)$$

Using (1), we may solve for the acceleration  $\vec{a}$  accordingly. At the end of each frame, we'll add  $\vec{a}$  to  $\vec{v}$  and then update  $\vec{p}$  accordingly. At 60 frames per second (FPS), the time step is small enough to produce a smooth and realistic-looking simulation. A caveat is that as the number of bees increases, rendering overhead results in a slight degradation in FPS.

### 3.5 COALESCENCE

In spite of the existing interactions, we're still unable to replicate what we observe in [8], because we have yet to incorporate the concept of coalescence in our simulation, i.e., the bees aren't converging towards the queen. To resolve this dilemma, we propose two-parameter approach consisting of a *crawl rate* and a *climb rate*. The former affects bees that are attached to the board, while the latter affects bees that are only attached to other bees. In general however, there are several common principles that we rely on in crafting our subsequent equations.

- There should be an element of randomness in deciding whether to let a bee advance towards a queen.
- The rate at which a bee moves towards the queen should depend on how far it is from the queen; bees that are far away should strive to move closer as rapidly as possible, while bees in the close vicinity of the queen are content with their position and should remain near stationary.
- Bees that are supporting other bees shouldn't be able to advance as fast as those that are unencumbered.

In any case, let's denote  $p = \frac{x - w_c/2}{w_c/2}$ , where  $x$  is the x-coordinate of a point. Note that when a bee is far from the queen,  $p$  is large; and when it's near the queen,  $p$  tends to zero.

#### 3.5.1 CRAWL RATE

Let's consider the case where we have a standalone bee attached to a board that isn't supporting other bees. First, denote the unit vector pointing from this bee to the queen as  $\hat{v}$ . Next, let's compute a probability  $c$  that dictates whether to advance the bee. The crawl rate ( $\alpha$ ) is a value that ranges between 0 and 1 and determines  $c$  as such:

$$c = \frac{p + \alpha}{2} \quad (2)$$

Assuming the bee is able to advance, we'll alter the acceleration as such:

$$\Delta a_x = \frac{v_x}{|v_x| \cdot (4 \cdot (1 - \alpha) + 6)}, \Delta a_y = \frac{v_y}{30} \quad (3)$$

Similarly, we can derive similar formulas for the case when the bee has to support one or more bees:

$$\Delta a_x = \frac{v_x}{|v_x| \cdot (4 \cdot (1 - \alpha) + 9)}, \Delta a_y = \frac{v_y}{30} \quad (4)$$

By tuning the denominator slightly higher, we can slow down the advancement of the bee.

#### 3.5.2 CLIMB RATE

The climb rate ( $\beta$ ) affects bees that aren't attached to the board. Ideally, bees along the peripheral of the bivouac would aim to climb downwards towards an imaginary summit of the structure. It's convenient to thus define a new point  $\vec{p}_i = \begin{pmatrix} w_c/2 \\ h_c \end{pmatrix}$  that represents this imaginary summit, and let's redefine  $\hat{v}$  as the unit vector pointing from the bee's position to  $\vec{p}_i$ . For the case where we have a bee on the perimeter of the structure, i.e., those not supporting other bees, we have:

$$c = \frac{p + \beta}{3} \quad (5)$$

Just as we did before, we'll have two acceleration deltas:

$$\begin{aligned} \Delta a_x &= \frac{v_x}{|v_x| \cdot (3 \cdot (1 - \beta) + 3)} \\ \Delta a_y &= \frac{v_y}{20} + \frac{U(-1/2, 1/2)}{15 \cdot (1 - \beta) + 15} \end{aligned} \quad (6)$$

Note that adding that element of randomness in  $\Delta a_y$  allows the bee to climb aggressively at times instead of displaying a monotonous climbing motion.

The fourth (and final) case to consider are the remaining bees situated *within* the structure. They form the

bulk of the bees as well. For a start, we'll use a lower  $c$  value to slow down their movement.

$$c = \frac{p + \beta}{4} \quad (7)$$

From there, we have:

$$\begin{aligned} \Delta a_x &= \frac{v_x}{|v_x| \cdot (6 \cdot (1 - \beta) + 6)} \\ \Delta a_y &= - \left( \frac{v_y}{20} + \frac{U(-1/2, 1/2)}{30 \cdot (1 - \beta) + 30} \right) \end{aligned} \quad (8)$$

Note that we flip the sign of  $\Delta a_y$  as these bees ought to huddle closer towards the queen rather than move outwards.

### 3.6 SIMULATION

After implementing the aforementioned details, we now have a test bench to collect data and tune parameters accordingly. Typically, it takes around five to ten minutes for all bees to be introduced into the system. Another five minutes is then required for the structure to stabilize. We only take measurements on this stabilized structure. When a stable state is achieved, we take the data from the latest ten frames (managed using a FIFO buffer).

Because still images do not quite capture the gradual transitions in this self-assembly process, we also recommend trying out the simulation here. There are several input buttons that you may use to alter the behavior of the bees.

## 4 RESULTS

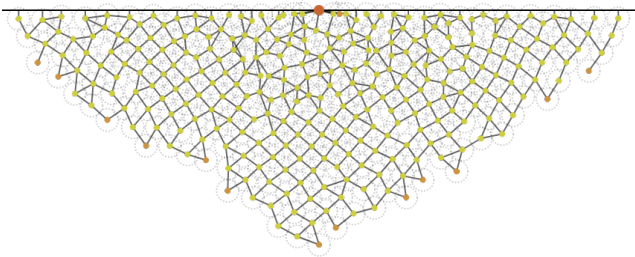


Figure 3: A 300 bee structure ( $\alpha = \beta = 3/5$ )

More often than not, a stabilized structure looks like the one in Figure 3. To offer some context, the small inner circles have radius  $r_n$  while the larger outer circles have radius  $r_b$ . The queen bee is the large glowing red circle at the core of the bivouac. Yellow bees are those that are supporting at least one bee, while red

bees are those without any bees to support, and are therefore prime candidates to climb downwards as described earlier.

One may observe that the outer regions are characterized by a diagonally-centered square lattice structure, whereas inner regions lack a discernible pattern. Additionally, outer regions are less dense compared to the tightly packed inner region. This is to be expected as you'd require more bees at the upper layers to support the weight of the layers below them.

### 4.1 BEE COUNT

To begin, we mirror the analysis in [11] by varying the bee count while keeping the other parameters constant ( $\alpha = \beta = 2/5$ ). In addition, we adopt the *binning method* by collecting samples at distinct, uniform intervals down the vertical axis, which we'll denote as  $z$ .

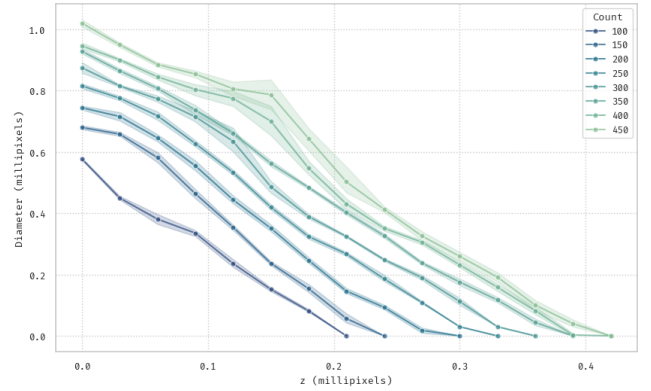


Figure 4: Diameter against depth  $z$

From Figure 4, it appears that the diameter decreases almost linearly with depth  $z$ . However, this linear trend doesn't necessarily hold true for other parameter configurations as we'll discover subsequently.

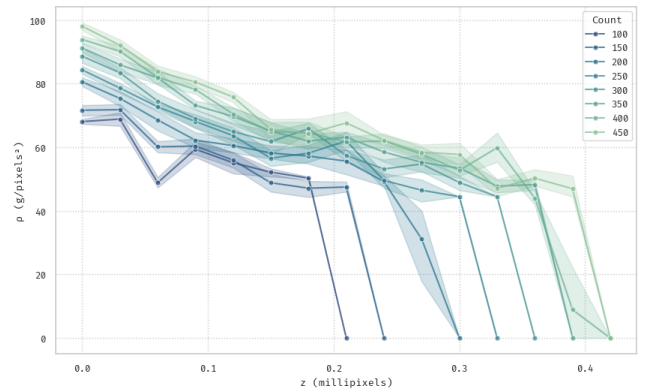


Figure 5: Density  $\rho$  against depth  $z$

Next, we examine how the density  $p$  changes with  $z$ . From Figure 5, a gentle linear downwards slope is prevalent at first. However, past a certain point, the density sharply dips to zero. As unintuitive as this appears, this result actually corroborate with those established in physical experiments [11].

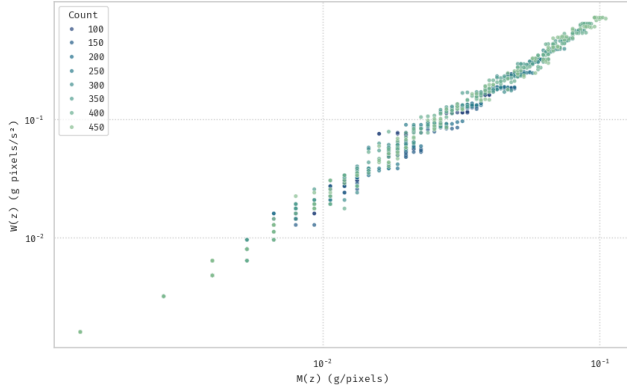


Figure 6: Supported weight  $W$  against layer mass  $M$

By defining the supported weight as the sum of the weight of the current layer along with all other layers below it, Figure 6 reveals that when a logarithmic scale is used, the supported weight appears to increase linearly with the mass of a layer. By fitting a power curve of the form  $W = C \cdot M^a$ , we are able to obtain Figure 7.

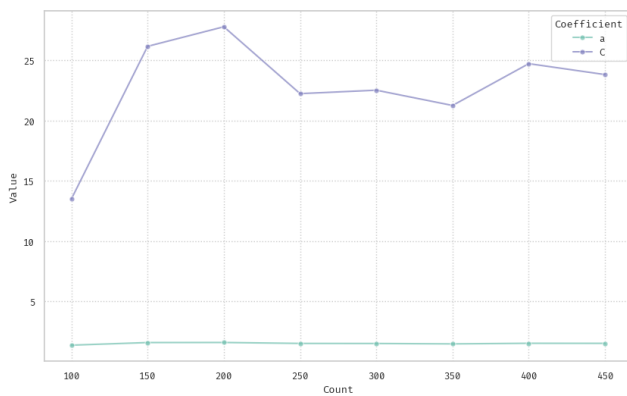


Figure 7: Coefficients of Best Fit

Noting that  $a$  is near-constant across different bee counts, we may deduce that our simulation does abide by the established strength-mass scaling law.

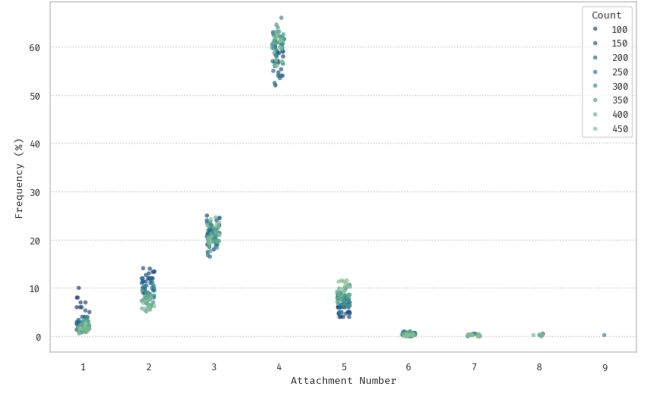


Figure 8: Frequency against attach number

We have consolidated the frequency distribution across attachment numbers. We define the *attachment number* as the sum of in-degrees and out-degrees, i.e., the total number of bees that a bee is attached to or is supporting. Referring to Figure 8, it is evident that the predominant attachment number is *four*, reflecting that the square lattice structure is the predominant structure.

Another subtle observation is that as the number of bees increase, there's a slight shift towards higher attachment numbers, i.e., those above four. This may be explained by the need to have a large densely-packed base layer to support the heavier load.

## 4.2 CRAWL RATE

In this section, we'll attempt to vary the crawl rate  $\alpha$  while keeping the climb rate  $\beta$  constant at  $2/5$  and the number of bees constant at 250.

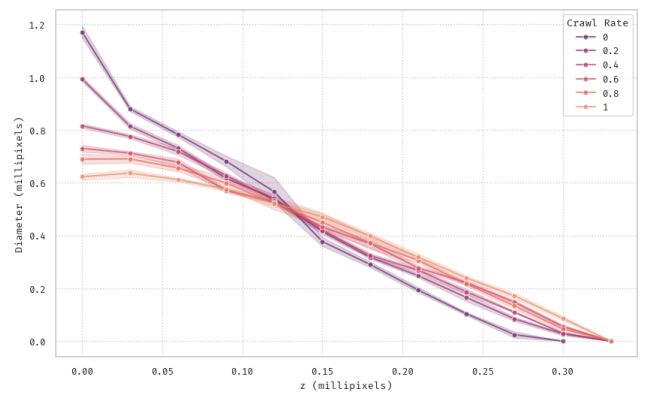
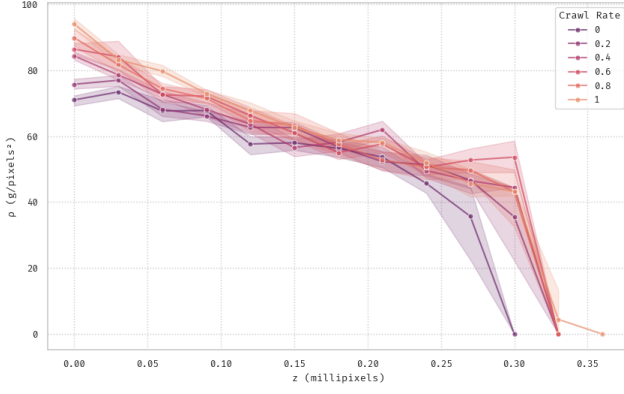
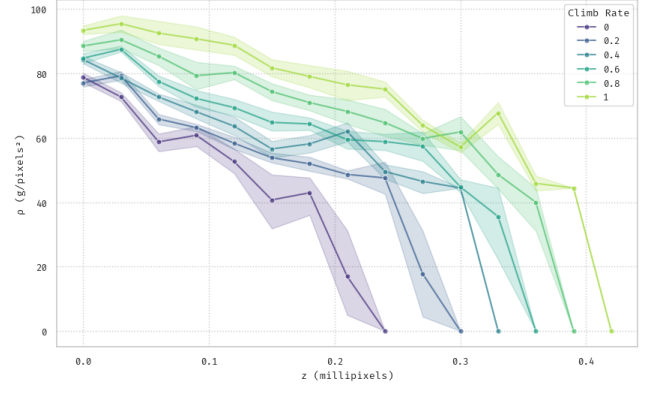


Figure 9: Diameter against depth  $z$

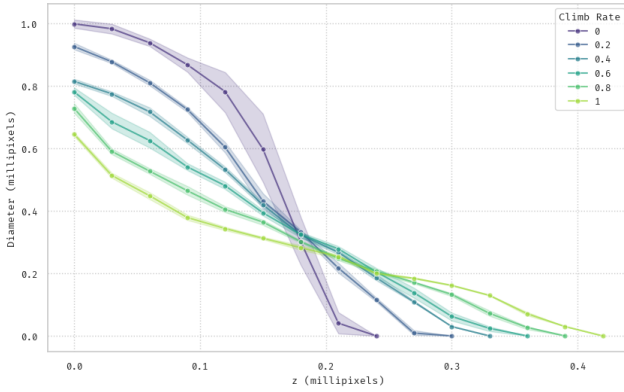



 Figure 10: Density  $\rho$  against depth  $z$ 

 Figure 12: Density  $\rho$  against depth  $z$ 

At first glance, Figure 9 and Figure 10 both present a similar overall structure that we've come to know from subsection 4.1. However, it's worth noting how muted the impact of  $\alpha$  is at influencing both the overall shape and density of the bivouac. Even, as we tune  $\alpha$  from zero all the way to one, it appears that the bees only react slightly.

This isn't surprising considering  $\alpha$  only affects those attached to the board, which is just a fraction of the overall swarm, so much of the impact would be concentrated at the base layer. By focusing on low values of  $z$  in Figure 9, it's clear that  $\alpha$  plays an important role at the bottom layers.

### 4.3 CLIMB RATE


 Figure 11: Diameter against depth  $z$ 

### 4.4 PHASE DIAGRAM

Thus far, we've only considered  $\alpha$  and  $\beta$  independently. However, that does not paint a complete picture of the full range of morphologies that our simulation is able to produce. As such, let's vary both  $\alpha$  and  $\beta$  at the same time and observe the resulting bivouac structures. Additionally, we'll be keeping the number of bees constant at 100.

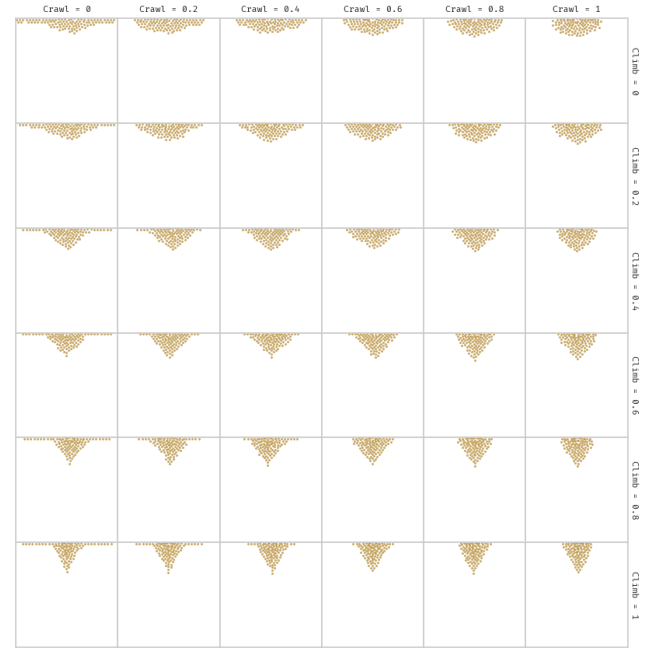


Figure 13: Phase Diagram

Through Figure 11 and Figure 12, we reveal a stark difference between the crawl and climb rates, in the sense that small perturbations in  $\beta$  have a massive impact on the morphology of the structure. Intuitively, as the bees try to climb the mount more vigorously, it's only logical for a pointy densely-packed formation to be the result.

With reference to Figure 13, we may now inspect the entire spectrum of states. As alluded to earlier, it holds that  $\beta$  is the dominant parameter in the simulation. Another observation is that an ideal parameter configuration would be around  $\alpha = \beta = 2/5$ , as that morphology resembles what we observe in physical experiments conducted in [8].

Another remarkable property is that the real-life parameter *temperature* may in fact be a linear combination of  $\alpha$  and  $\beta$ . From [6] swarm morphologies exist as means for temperature regulation. Low temperatures produce structures with a higher  $\alpha$  and a low  $\beta$ , whereas high temperatures work in reverse, producing more pointy structures with a low  $\alpha$  and a high  $\beta$ .

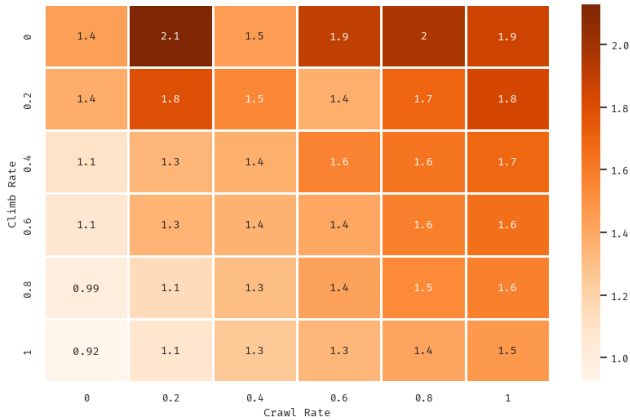


Figure 14: Heatmap of  $\alpha$  coefficients

Besides looking at the shape of the bivouacs, we may also examine how the  $\alpha$  coefficient changes when we fit the power curve. In Figure 14, it's evident that higher crawl rate configurations result in larger  $\alpha$ , suggesting that each layer is supporting far more weight in these pointy structures.

## 5 CONCLUSION

## ACKNOWLEDGMENTS

I would like to acknowledge the funding support from Nanyang Technological University – URECA Undergraduate Research Programme for this research project.

## REFERENCES

[1] Carl Anderson, Guy Theraulaz, and J-L Deneubourg. Self-assemblages in insect so-

cieties. *Insectes sociaux*, 49:99–110, 2002.

- [2] Daniele Carlesso and Chris R Reid. How to become one: the proximate mechanisms of self-assembly behaviour in social insects (hymenoptera: Formicidae, apidae). *Myrmecological News*, 33, 2023.
- [3] NH Fefferman and PT Starks. A modeling approach to swarming in honey bees (*apis mellifera*). *Insectes sociaux*, 53:37–45, 2006.
- [4] RC Fetecau and A Guo. A mathematical model for flight guidance in honeybee swarms. *Bulletin of mathematical biology*, 74:2600–2621, 2012.
- [5] Christina M Grozinger, Jessica Richards, and Heather R Mattila. From molecules to societies: mechanisms regulating swarming behavior in honey bees (*apis* spp.). *Apidologie*, 45:327–346, 2014.
- [6] Bernd Heinrich. The mechanisms and energetics of honeybee swarm temperature regulation. *Journal of Experimental Biology*, 91(1):25–55, 1981.
- [7] Chung-cheng Lin, Tze-wei Chen, and Ching-chong Lai. The economics of honeybee swarming. *Regional Science and Urban Economics*, 33(5):581–594, 2003.
- [8] Orit Peleg, Jacob M Peters, Mary K Salcedo, and Lakshminarayanan Mahadevan. Collective mechanical adaptation of honeybee swarms. *Nature Physics*, 14(12):1193–1198, 2018.
- [9] Ken Perlin. Improving noise. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 681–682, 2002.
- [10] Thomas Schmickl and Heiko Hamann. Beeclust: A swarm algorithm derived from honeybees. *Bio-inspired computing and communication networks*, pages 95–137, 2011.
- [11] Olga Shishkov, Claudia Chen, Claire Allison Madonna, Kaushik Jayaram, and Orit Peleg. Strength-mass scaling law governs mass distribution inside honey bee swarms. *Scientific reports*, 12(1):17388, 2022.