Justin Salamon

Project Report

5/19/2023

Background

The Honeybee is a eusocial insect that plays a vital role in pollinating fruits, nuts, and vegetables throughout the U.S (Torres et al., 2015). A eusocial population is one that has a division of labor decided by an age-based caste. The exact definition requires fulfillment of three features, cooperative brood-care, an overlap of generations that allows the colony to sustain itself, and a reproductive division of labor (Bee Health, 2019). Basically, this means that the bees split up into different groups based on age, and all devote themselves to a specific task while the queen bee lays eggs to sustain the population. The caste division is egg, larvae, pupae, hive bees and foraging bees, with hive bees usually being further divided into nurse bees and processing bees. Eggs, larvae, and pupae are the stages of life before a fully grown bee emerges and are cared for by the hive bee's caste. The forager bees are the oldest caste, with all bees eventually ending up there and leaving the hive daily to search for food. The queen bee is the sole source of eggs within the hive, and there is only one at a time. While the queen can die and multiple new queens can compete for the spot of new queen, this is not considered in the model. The success of the colony is due to a variety of factors, including food acquisition rates, weather, and even invasive species such as hive mites (Torres et al., 2015). These factors naturally lend themselves well to simulation and have been included alone or together with others in many prior models.

Motivation

The honey bee plays an important part in the U.S. economy because of its contributions to pollinating crops. There has been a steady decline of honey bee colonies in the U.S. from 6 million in 1947 to 2.5 million in 2015. The rate of loss of colonies also rose in general during the 2010's from a historical rate of 10-15% to an average loss of 30% (Torres et al., 2015). The honey bee is in danger, much like many other insect populations. Bee keeping offers up one way to reduce loss and maintain populations. For this reason, modeling colonies and collecting data shows how factors such as food acquisition, weather, invasive mites, and the spread of disease can impact the population is particularly important. Colony models have been around for a very long time, Fukuda and Sakagami (1968) being an example from the 1960's. Work on these models inevitably has and will continue as the damage to the environment by climate change and other harmful phenomena has only gotten worse as time goes on. Data on bee populations in the U.S. is collected every year, and new models have been published as recently as last year (2022). The amount of material I found on this topic and the number of factors that could be used to create even a simple model is what motivated me to pick it. Even with the comparatively small number of features implemented by me, we can still see the beginning of how more advanced analysis of hive colony models can produce useful results that can help save the bee population.

Simulation Description

For this model, I have taken inspiration mostly from the steady state model and partially from the transient model implemented by Torres et al. (2015). The simulations that I run are between 1 and 2 years, so at least 365 days. Each day the total population of a phase of bee life is recorded, along with the amount of food available. The bees have 5 phases of life that will be tracked in order, egg, larvae, pupae, hive bee and forager bee. Each phase lasts for several days according to data presented by Torres et al. (2015). The basic idea behind the simulation will be to first calculate what amount of the population in each phase stays in that phase or moves on to the next. Each phase has a survival rate that determines what percentage of the population on any day will make it to the next day. The next step calculates how much food is required for each phase, which has again been set according to data provided by Torres et al. (2015). The consequences of not having enough food manifest in the survival rate of each phase. The survival rates of hive bees and forager bees can potentially be reduced by a lack of food. If there is enough food again the survival rates will go back up to normal. The forager bees are then allowed to go out and collect food, in this model they only go out after eating. Foraging before eating will have an impact on whether there is enough food to eat, so it is better to have foraging occur after eating. At the end of the day the total population of each phase is recorded, the total amount of food that is available for the next day is also recorded. The final check will be to see if the colony is still able to continue. The colony will be unable to continue if the only bees left alive are eggs.

The additional features that are implemented are cannibalism and weather effects.

Cannibalism occurs when there is not enough food for all the bees that require food to eat. In this case the hive and forager bees will then instead start eating the larvae to make up for the food deficit. The larvae have an experimentally determined nutritional value given by Torres et al (2015). If there's enough larvae to make up for the food deficit then everything is fine, and every bee will continue along like normal. If there are not enough larvae, then the survival rates of the adult bees will sharply decrease.

The effects of weather are simulated by tracking the season. The seasonal changes here only occur during the winter. There is some level of fluctuation in things such as egg laying rates, food acquisition rates and overall survival in general depending on the season. The easiest one to show is the winter changes, which show the greatest fluctuations in all the rates. Over a span of a year, winter takes place from day 151 to day 240. During this time no foraging will take place, and no egg laying will take place. The hive effectively goes into a hibernation state, where all the bees who are currently alive will reach maturity and stay as a hive bee. All these hive bees will live longer during the winter since their only activity is eating. Each day a very small percentage of them will die, however. While there is enough food stockpiled to survive, every day there will be no changes. When the food runs out more of the hive bees will perish every day, their survival rate decreases. Once winter is over business resumes as normal, egg laying and foraging all begin again.

Parameters and Code

Here I will explain some details about the parameters that need to be used and how the code is set up. The NumPy library is of course used for its array data type, as I must use many large arrays that all contain floating points.

```
# number of days simulation is run
days = 365
# Minimum Egg laying rate of queen
# queen should lay at least this much and at most twice as
# much per day
# during winter months drops to 0
# during fall drops to half
egglaying = 1500.0
E_0 = egglaying
# Hive capacity - Called capacity but might just end up being some magic
# number to get random starting points for populations.
hive_capacity = 49000.0
```

The days parameter is used for how many days the simulation will run. The simulations should be run for at least a year given that stuff like cannibalism or seasonal effects needs time to influence the population. The egglaying parameter is set as constant here and should be between one to two thousand. Egglaying can be increased to a larger range if the hive grows to a sufficient size, but I did not implement this here. The queen will usually lay a constant amount eggs per day, though later I will use the NumPy uniform function to generate random amounts of eggs per day. The hive_capacity is not actually meant to limit the growth of the hive. Instead, the hive_capacity is used to determine the initial population of the hive and to divide up the population into each caste. It only has this name because it was initially going to limit the capacity of the hive, but this ended up being unnecessary in the end.

```
# days spent as an egg
days_in_egg = 3
# days spent as a larva
days_as_larvae = 5
# days spent as a pupae
days_as_pupae = 12
# Lifespan of hive bees
hive_bee_span = 21
# Lifespan of forager bees
forager_bee_span= 14
```

```
# Egg survival rate
S_egg = 0.94
#Larvae survival rate
S_larvae = 0.917
# Pupae survival rate
S_pupae = 0.985
# Hive bee survival rate
S_hivebee = 0.985
# Forager bee survival rate
S_forager = 0.9
```

Here are the parameters for the days spent in each phase and survival rates of each phase. The days spent parameters should remain constant throughout the run of the simulation. Despite having to extend the life span of the hive bee phase due to weather, the hive_bee_span parameter remains constant, and a different method is used instead. Survival rates determine what percentage of a phase lives when transitioning to the next day. As an example, the eggs have a survival rate of 0.94, the number of two-day old eggs will be 94% of the number of eggs that are 1 day old, the same goes for the three-day old eggs. Survival rates can also be interpreted as mortality rates, where we instead consider (1 – S_egg = 0.06) for example, which means 6% of eggs die every day. The survival rates are subject to change of course, be it from initial conditions, varying during seasonal changes, lack of food or even just day by day. These rates have a significant impact on how the population grows, and what maximum values it can reach.

```
# egg consumption
C_egg = 0
# larvae consumption
C_larvae = 0.018
# pupae consumption
C_pupae = 0
# hive bee consumption
C_hive = 0.007
# forager bee consumption
C_forager = 0.007
```

```
# nutritional value of larvae, 0.025 g - experimentally found
nutrition_L = 0.025
# Minimum Forager food acquisition rate - nectar and pollen
# can be influenced by seasonal changes or by illness.
# Initially constant rate
food_per_day = 0.1 # gram/(bee*day)
initial_food = 20000.0
```

Here are the parameters for food required per day, indicated by the C prefix, and followed by the nutritional value of the larvae, initial food stores and amount of food gathered per day. Food is measured in grams/bee*day. The food consumption rates and larvae nutrition were decided upon by experimental data provided by Torres et al. (2015), thus they should remain constant in the simulation. These parameters would probably need to be changed if we wanted to use a different kind of bee. The food gathered per day can remain constant or be randomly determined. Constant of course helps isolate the relationships between population and any other effect or parameter. Randomizing the food gathered is more realistic though and can naturally lead us to different outcomes where there may be a food deficit. The initial food unintentionally ends up being one of the most important parameters, as it generally determines whether the population will live past the first 30 days or so. This is not ideal though, some possible solutions that I was not able to implement and test fully involved generating exactly enough food for the initial population, or even twice as much. In the end though it is very difficult to simply create a hive with initial conditions that don't directly cause an immediate collapse of the population or allow them to grow to their peak easily.

```
# Eggs per day
Eggs = np.zeros(days, dtype=float)
Eggs[0] = init_eggs*initial_pop
# eggs will last 3 days, so they belong in egg_life
# initially randomize the number of eggs in the these 3 slots to match
# the initial egg amount
egg_life = np.zeros(days_in_egg, dtype=float)
for i in range(0, days_in_egg):
    egg_life[i] = Eggs[0]/days_in_egg
```

Here is an example of how the phases of life are stored. I use a NumPy array of floats that has the initial population of eggs divided up into each index. The other phases are stored the same way, but their corresponding array is of course if the days they spend in that phase. With this method of storing the population, I can easily track the total population for each phase and the individual population of a phase that is x days old. The unfortunate thing about using this method is that it creates too much difficulty when trying to change the time spent per phase mid simulation. One way this could be improved is to distribute the bees more randomly between each of the indices of their phase of life array, as splitting them evenly like this is not as random as nature really is.

A better way that could be used to store the population in general would be one long array that is divided up by the parameters of days spent per phase. For example, if we sum up the total days spent in each phase, then a bee can live for 55 days. An array with 55 spaces can be used to store all the bees at once, where indexes 0-2 store eggs, 3-7 store larvae, etc. This way I can easily shift the time spent in a phase mid simulation, and easily accelerate or decelerate the bee's maturation rates.

There are three important functions that get used for the simulations. The first is initialize(), which clears any collected data after a simulation has been run. The second is Right_Shift(arr, arg) which takes an array called arr, and pushes every element one space over, but scaled by arg. This lets me easily advance each bee to the next day of its life in the current phase it is in, by passing the relevant array as arr and the survival rate for the phase as arg. The last is Cannibalism(i: int), which is used to simulate cannibalism. This function will calculate the amount of larvae that get eaten and adjust the population accordingly. It has a return value of True if there was not enough larvae to baalnce the food deficit, and False if there was enough. Using this lets me easily tell when survival rates need to decrease because of starvation.

```
# take hive bees from end of hive bee life and put into forager bee life 0
forager_bee_life[0] = hive_bee_life[hive_bee_span-1]

# Hive Bees
Right_Shift(hive_bee_life, S_hivebee)
# take pupae transitioning to hive bee
hive_bee_life[0] = pupae_life[days_as_pupae-1]
```

Here is an example of how the bees get processed on any day. This example focuses on hive bees. The first step is to add all the hive bees on their final day of the phase into the first index for the forager bee phase. Then I can just use the right shift function on the hive bee array and discard the last element that already went to the forager bee array. Lastly, the hive bee array then takes all the bees who've finished their time spent as pupae into its first index.

This shows how the bees are all processed in general, the Hive Bees 0 days old = Pupae 12 days old, the Hive Bees i > 0 days old = Hive Bees i-1 days old * Hive Bee survival. For the eggs it is slightly different, where the eggs receive their first index population directly from the queen. All the simulations follow this same basic method for processing the bees and the food too.

```
# queen lays eggs into egg_life[0]
if(laying):
    egg_life[0] = random.uniform(500,2000)
```

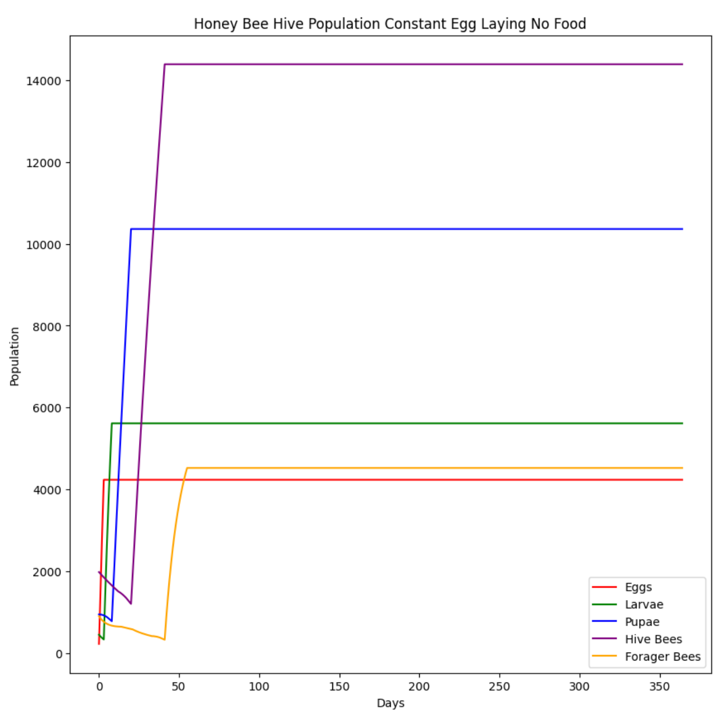
```
food_today = 0.
for j in range(0, int(Forager_Bees[i])):
    food_today += random.uniform(0.,0.1)
```

Some differences between simulations include the introduction of random egg laying rates, which add some fluctuations to the egg population, and by extension the rest of the population. I can also introduce randomness in the food gathering, by picking some amount of food for every forager bee, and then summing that up to add to the total food available the next day. The last difference is in determining the season, which is simply just an if statement on how many days have passed since the beginning of the simulation. If it's winter then nothing really happens, but otherwise everything previously described occurs.

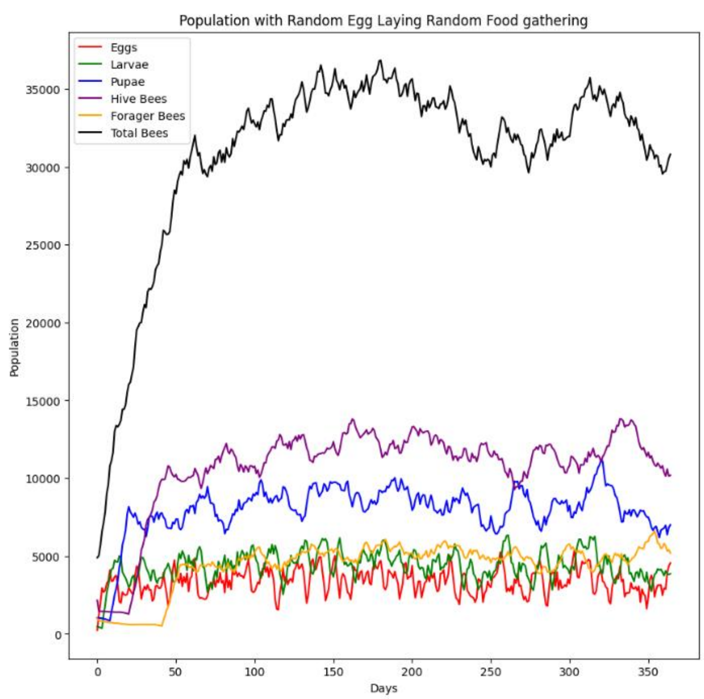
Overall, this code was mostly easy to write but I did have some problems with it. Cannibalism was very difficult to implement properly since I had to figure out some way to drain the larvae storage of enough of the larvae at a time. In addition, making sure that the survival rates would change correctly was kind of shaky, but it seems to be correct now. Implementing the seasonal change was a lot more difficult than it seemed it would be. The problem comes from the fact that the hive bees all need to live during the winter. I also need all the forager bees to die and for all the bees below hive bee age to transition into hive bees. In addition to this, the queen needs to stop laying eggs, which ends up colliding with the problem of getting the hive to die when the population consists of only eggs. Eventually this all ended up working mostly correctly, though there are assuredly still some bugs left behind which I haven't found yet. More test values need to be considered to find the rest of the bugs.

Results

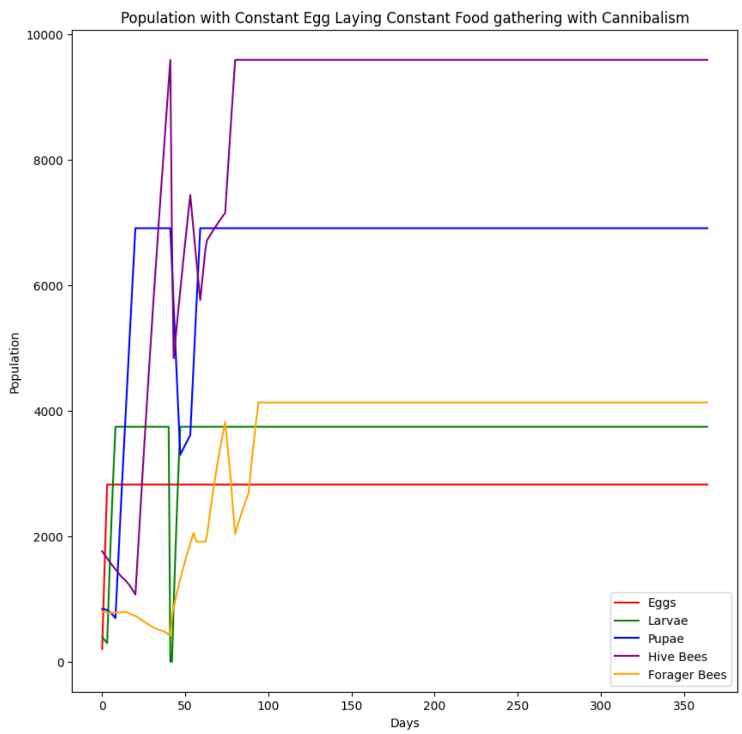
Here are some charts showing the results of the simulation when run multiple times. Each time I included one additional feature such as randomness, food, or season. All but one of the plots is days vs population, with each phase separated into an individual color. I don't include any of the initial conditions here since it's mostly irrelevant anyway, the only thing that it would show is how the initial food and initial population can end up deciding if the hive lives or dies right away, which is an unintended consequence of how the code is written.



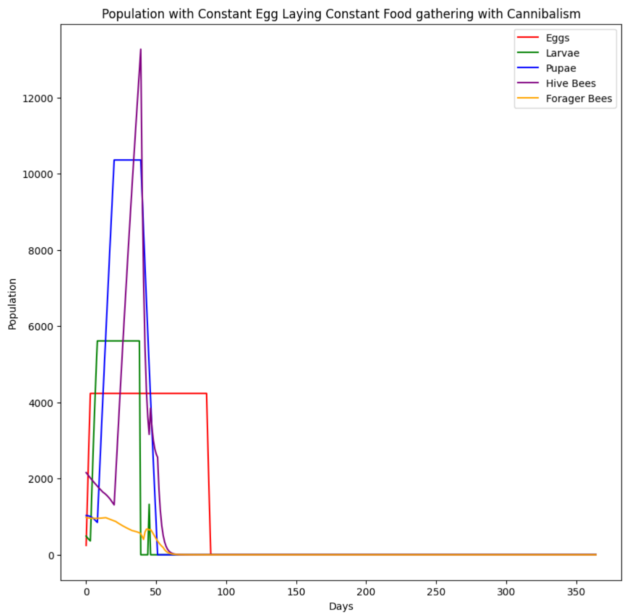
This is the result of a simulation where the population is allowed to grow without concerns over food or anything else. The initial dip in population is caused by the fact that the initial population has many bees at the end of their life as a forager. This will always occur due to the way the initial population is set up. Once the population is allowed to live for around 50 days, each phase has reached a maximum value. The sum of the heights for each of the phases equals the total population of bees. This is not very interesting, as the bees have no threat to their lives, and are just free to grow however they please. One possible way to unlock the maximum limit of the bee population here would be to use the hive capacity parameter as an indicator for when to increase the egg laying rate.

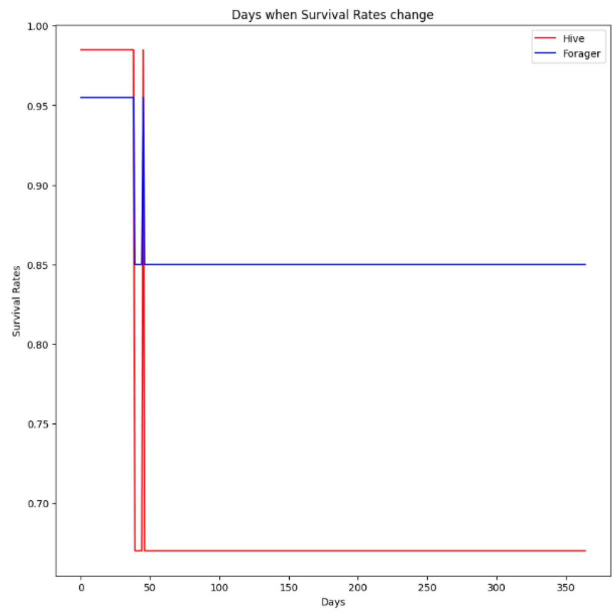


This is the result of including randomness in the simulation. Random food and egg laying produce this scratchy uneven plot. The black line which only exists here represents the total population of bees overall. Each population tends to hover around the same maximum values that they attained before. It's clear here how the initial conditions like egg laying and initial population can still limit the population growth. There could possibly be some extra condition added that could somehow allow the queen to grow the hive past these limits. The effect of randomness does introduce some more noise that could make it hard to see the effects of additional features. Food gathering can affect the chances of cannibalism occurring, but it's more accurate to real life if we don't have every bee gather the same amount of food every day. For seasonal changes, food gathering and egg laying both have an effect. The population during the winter is obviously based on how many eggs were laid overall, and the chances that they can survive is also based on how much food they gather. It may be better to not have randomness, and I've only used it for adding seasonal effects.

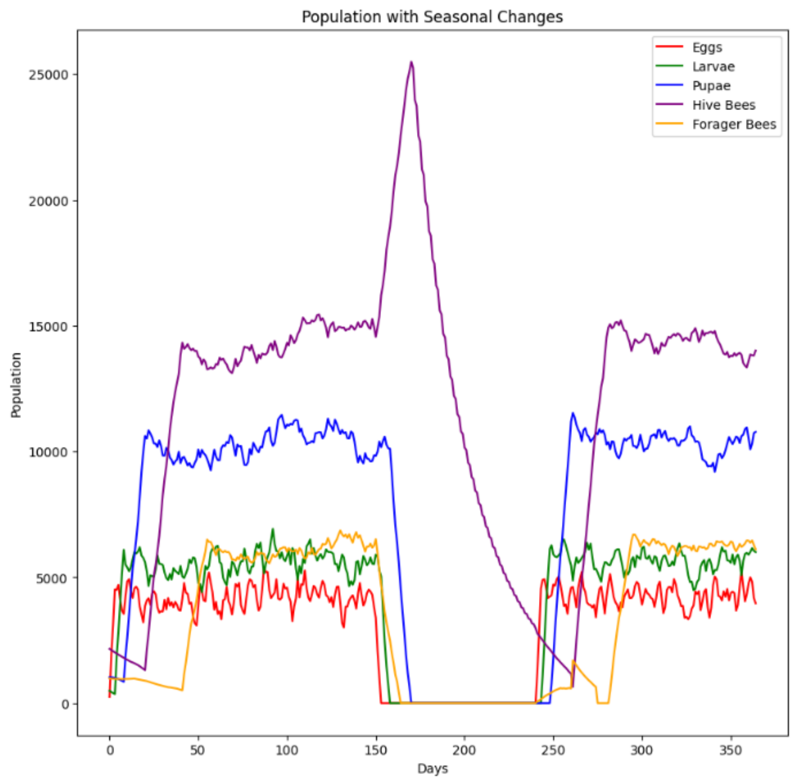


This is the result of including cannibalism to the simulation. The bees start out fine but end up reaching a point where they run out of food. They then begin to cannibalize the larvae for a very short period of time and actrually end up surviving and rising back up to their steady population levels. This is actually an outlier that happened very rarely. The majority of the time the hive just ends up dying competely. This is one place that matches what Torres et al. explained in their paper. Cannibalism typically does end up killing the colony in their model, and they even comment on the benefits of it only being seen in their model which considers not just days as steps of the simulation, but steps within days too. Steps within days could be hours for example, or a division like morning, noon and night. If we are able to look more closely at what happens during a day, then maybe cannibalism can happen but be stopped with the arrival of more food from foragers who left during a prior step during a day.





This is the more typical result when cannibalism is included. On the left we can see that they run out of food somewhere around day 30 and start cannibalizing. They repeat this but are never able to recover and gain enough food. The combination of no larvae being able to grow into adult bees and not enough foragers being left as their survival rates decreased ends up reducing the population enough to cause the queen to cease laying eggs. On the right is a simple plot that shows when the survival rates decrease, they do recover for a few days but then immediately drop back down to running out of food. So, cannibalism failed to save them this time, which is what usually happens.



This final chart shows the typical effects of implementing seasonal changes. This simulation also included some randomness. The population can rise to near maximum levels for the first 150 days. After this winter starts and we can see that the hive bee population quickly explodes to encompass the entire population of bees. During the winter the population decreases slowly for a very short while, but then quickly starts decreasing until the hive nearly dies out. The hive is then able to recover and head back to its near maximum levels. This simulation only considers extreme cold; however, it could be expanded to include extreme heat or other weather conditions such as rain. The heat or rain can also impact the food gathering rates too, which may further push the bees to engage in cannibalism.

Conclusions

Overall, I was not able to implement as many of the features that would simulate potential threats to the hive. There are hive mites, which are invasive bugs that sneak into hives and infect bees with a virus that damages their wings and can considerably damage the hive. This would end up being very complicated though, as it requires not only implementing the simulation of mites on their own, but the interactions between the mites and the bees. This would most likely be some kind of host-parasite model and would not really fit well with what I've already set up to simulate. There were also problems with my code, where the conditions to have the hive die were very difficult to implement and sometimes may not even work correctly. Eventually it worked properly, as evidenced by the cannibalism portions ending up in dead hives often. The difficulty, and by extension the time needed to implement the more advanced form of the model described by Torres et al. was also a problem. The version that I have implemented is very basic compared to their version which uses differential equations. To implement the differential equations-based model would require implementing the single array method of storage for the hive population that I described previously, which would in turn introduce a lot of problems in getting it to work correctly too. It was also possible to include pheromones in the simulation, specifically ones that would influence the survival rates of the larvae, and the time spent in the pupae and hive bee phase. There could actually be some interesting results produced when all of these effects are put together, as they all inevitably end up affecting each other in some way. In the end I chose to stick with the model that I implemented because it does show accurate results of the seasonal effects, and the effects of cannibalism. With more time I would have been able to expand upon the seasonal effects and include some more results from the multitude of other papers on the subject referenced in Torres et al. There is a massive amount of work on this subject, so further research into it would be possible and further improvements could be made to what I already have, even entirely remaking the model.

Sources

Bee Health. (2019, August 20). *Bees and social insects*. Bee Health. https://bee-health.extension.org/bees-and-social-insects/#Definition_of_eusociality

Fukuda H, Sakagami S. Worker brood survival in honey bees. *Res Popul Ecol*. 1968;10: 31–39. 10.1007/BF02514731

Torres, D. J., Ricoy, U. M., & Roybal, S. (2015, July 6). *Modeling honey bee populations*. PloS one. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4493160/