MATHEMATICS OF EVOLUTION

DAVID MUMMY

Abstract. The Adaptive Landscape (AL) provides biologists with a heuristic for modeling the way the forces of evolution affect a population of organisms. We investigate the possibility of creating a computer simulation of a population moving around an AL. First, we create a simple landscape using a Gaussian distribution, and then a sample population in phenotype space. Then we explore methods of simulating the population's motion through phenotype space over time.

1. Introduction

On the surface, biology appears to be one of the least mathematical of the three major sciences. While physicists and chemists benefit from many and varied applications of diverse branches of mathematics, the stereotypical biologist is canvassing the jungle for a rare species of howler monkey, mutating fruitflies, or scrutinizing bacteria under a microscope, not puzzling over equations on a whiteboard. Biology would seem to be a largely qualitative discipline. However, experience has taught us that there is very little under the sun that mathematics cannot be utilized to investigate, expand on, and model; and biology is certainly no exception.

It should come as no great surprise, then, that evolution, the single most unifying concept in biology as well as one of the most influential scientific ideas in human history, lends itself readily to such mathematical endeavors. The gradual flow and change of genes and species over time, and how those genes and species are molded, selected for, and driven out of existence by natural selection, is a fascinating subject to explore from an analytical standpoint. We intend to explore methods of modeling changes in population over time (genetic drift, speciation, extinction, and so on), using a concept called "The Adaptive Landscape" as a framework for my investigation. Several ideas in this paper, as well as the inspiration for the project as a whole, are due to Arnold *et al.*

This paper begins with a brief description of the development of the adaptive landscape as a tool for modeling evolution, from its inception in 1932 to current work in the field. We will also detail both the promise, and criticism, of the model. In Section 3, we'll explore the paper used as a basis for the investigation. After that, a more detailed exploration of the computer coding techniques used will be presented, followed by some conclusions regarding the potential of this particular model.

We will begin by answering the obvious first question: what is an Adaptive Landscape?

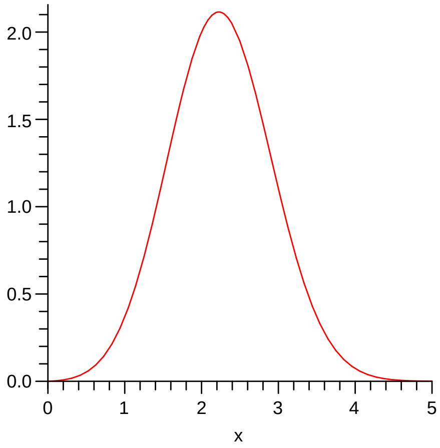

Figure 1. Sample distribution of leg size among individuals within a population. The scale is arbitrary.

2. The Adaptive Landscape

It is a central tenet of biology that a certain amount of genetic variability is common to all species (for if it were not, and all organisms in a species were exactly the same from one generation to the next, natural selection, and thereby evolution, could not occur). Humans, as an example, exhibit a wide range of heights, body types, eye colors. They also have other traits that are not so visually obvious, such as resistance to disease, metabolism rate, and so on. These differences, excluding various environmental factors such as diet and climate, are caused by the genes present in an individual's DNA. However, the interplay between genes is so complex that for the purposes of this investigation we will ignore genes and concentrate rather on the physical traits they cause in an organism. Natural selection seldom acts on genes themselves; it is, rather, these inherited traits, called phenotypes, that determine whether an organism dies young or lives to procreate and pass on its genes, and thereby its phenotypes as well, to another generation. Relating an organism's phenotypes to its propensity to procreate (in biological jargon, its fitness) forms the basis of the adaptive landscape that will be considered here.

To get an initial grasp of what the adaptive landscape represents, it is easiest to start very simply, with one phenotype only, and build up from there. Consider leg length (certainly a phenotype of sorts) in the roadrunner (*geococcyx californianus*). In order to survive and reproduce, the roadrunner must evade its main predator (which, as everyone knows, is the coyote, *canis latrans*). It isn't too surprising that 6-inch legs would prove highly ineffective, for such a mal-proportioned bird would quickly be outrun and devoured; however, 10-foot long legs would also be a disadvantage, as the unfortunate bird would have a hard time keeping any sort of gait under control, much less eating, from such a ridiculous height. Therefore it is reasonable to conclude that there is a certain leg length that provides the roadrunner with maximum getaway speed, and hence that phenotype is the one most favored

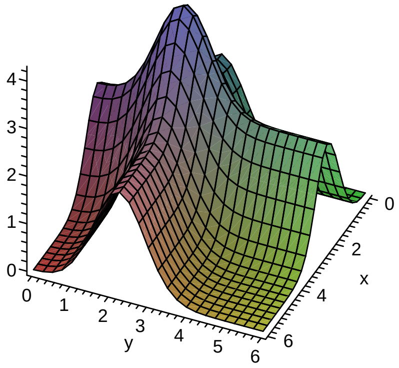

Figure 2. A sample Adaptive Landscape in phenotype-space. Note the fitness peak. Once again, the scale is arbitrary.

by natural selection (since those that deviate from the height are more likely to be killed off before they can reproduce). Phenotypic variation generally occurs in a bell-shaped curve, with the peak of the curve at the most "fit" sort of phenotype (there can be more than one peak, however; more on that later); see Fig. 1 for a sample leg-length distribution.

Now imagine another phenotype: beak size, to use an oft-cited example. Picture threespace with x, y, and z axes; let the x axis represent the leg length phenotype previously discussed and the y axis beak size. There exists a fitness function $f(x, y)$ that expresses the relative fitness of an organism in terms of its phenotypes; this surface in phenotype-space, shown in Fig. 2, is called the Adaptive Landscape. The peak on the surface, called a *fitness optimum*, represents the best combination of phenotypes from a fitness standpoint: natural selection encourages the proliferation of organisms that are closer to the fitness peak, for the tautological reason that they are more fit and hence create more progeny. Entire populations can be plotted on adaptive landscapes, in far more dimensions than the 3-D example outlined above.

We will investigate the possibility of plotting a population of organisms on such an adaptive landscape, and simulating the 'pull' of natural selection moving a population towards the peak. We will also explore the potential for simulating more complicated phenomena, as well. A landscape with several peaks, for instance, is certainly possible, leading to a population getting 'stuck' on a lower peak. We could also move towards a dynamic landscape rather than a static one, leading to events such as speciation (the divergence of one species into two) and extinction.

Before any of that is possible, though, we will need to create a simple model of the landscape itself and insert a population onto it. Our goal then is to create a numerical simulation of a population of organisms moving on the adaptive landscape. We will begin by creating the landscape itself, then move on to a population.

2.1. **The Quadratic Form.** Because of its simplicity, we're going to use a function called a *quadratic form* to create our sample landscape. Quadratic forms have the form $Q(\mathbf{x}) = \mathbf{x}^T A\mathbf{x}$, where $\mathbf{x}$ is a $1 \times n$ vector and $\mathbf{A}$ is an $n \times n$ matrix. They are so named since carrying out the matrix multiplication yields a term that is quadratic in every variable present in the vector $\mathbf{x}$. An example Adaptive Landscape can be described by the multi-variate Gaussian

$$W(\mathbf{z}) = \exp[-\frac{1}{2}(\mathbf{z} - \Theta)^T \omega^{-1}(\mathbf{z} - \Theta)]$$

where $\mathbf{z}$ is a point in phenotype-space, Θ is the optimum of the surface (the peak), and the matrix ω is made of vectors describing the selection surface itself[1]. This has an obvious parallel with a quadratic form; in fact,

$$W(\mathbf{z}) = \exp[Q(\mathbf{z})]$$

with $A = \omega^{-1}$ and $x = \mathbf{z} - \Theta$.

For our purposes, the actual values of A don't really matter, since we're attempting to create a qualitative model rather than obtain meaningful numerical results. All we need are parameters that describe a landscape that's 'nice-looking'.

In order to find how a population moves around the landscape, however, we need to find the derivative of the surface at a given point - the selection gradient. We will go ahead and derive the formula for the derivative, as it will come in handy later on when we write up our code.

2.2. **Differentiating the Multivariate Gaussian.** Consider the function

$$f(\mathbf{x}) = \exp[\mathbf{x}^T A\mathbf{x}]$$

where

$$A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}, \mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \end{bmatrix}.$$

This function f is what we'll be using for our Adaptive Landscape. It can be easily manipulated to have the correct shape and that's all we need at this point.

Doing the multiplication to expand out $f(\mathbf{z})$ yields

$$f(\mathbf{x}) = \exp[a\mathbf{x}_1^2 + (d+b)\mathbf{x}_1\mathbf{x}_2 + e\mathbf{x}_2^2 + (g+c)\mathbf{x}_1\mathbf{x}_3 + (h+f)\mathbf{x}_2\mathbf{x}_3 + i\mathbf{x}_3]$$

This is easy to differentiate in turns with respect to $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$.

$$\frac{\partial f}{\partial \mathbf{x}_1} = [2a\mathbf{x}_1 + (d+b)\mathbf{x}_2 + (g+c)\mathbf{x}_3]f(\mathbf{x})$$

and so on. A pattern of these derivatives quickly emerges, and we see that

$$\begin{bmatrix} \frac{\partial f}{\partial \mathbf{x}_1} \\ \frac{\partial f}{\partial \mathbf{x}_2} \\ \frac{\partial f}{\partial \mathbf{x}_3} \end{bmatrix} = \begin{bmatrix} 2a & d+b & c+g \\ d+b & 2e & f+h \\ c+g & f+h & 2i \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \end{bmatrix} f(\mathbf{x})$$

Therefore

$$\beta = (A + A^T)\mathbf{x}f(x)$$

yields a direction gradient vector (comprised of partials of $\mathbf{x}_1$, $\mathbf{x}_2$), etc. This result, it turns out, can be generalized to any number of variables (not just 3).

We have found that the derivative of $\mathbf{x}^T A\mathbf{x}$ is $(A + A^T)\mathbf{x}$, and applied this result to the Gaussian function we're using to describe the adaptive landscape. This will be helpful in the future in allowing us to calculate how populations will be "pulled" towards fitness peaks.

3. Adding a Population

To start off, we'll take a simple adaptive landscape (created in the manner described above) and plot a "trait mean point" and its motion through phenotype space. The trait mean point is the average value of all the phenotypes among individuals in a population; it's a way of collapsing an entire population into one data point for simplicity. We're going to take this mean point and plot its movement along a landscape. Naturally, since the landscape represents the viability of certain traits over others, we expect the point to travel up the landscape to the nearest peak (indicative of highest fitness), and then stay there. The way we'll go about modeling this is with the following iterative process:

1. Pick an arbitrary point in phenotype-space (call it $\bar{z}$)
2. Find the gradient vector β of the AL at that point
3. Find our new $\bar{z}$ with the equation

   $$\mathbf{\bar{z}_n} = \mathbf{\bar{z}_{n-1}} + c\beta$$

   where c is just a small scaling constant, to keep $\mathbf{\bar{z}}$ from jumping all over the place.
4. Return to step 2.

This is pretty easy to code, but it is not that interesting, as the trait mean simply follows a path directly up the gradient vector to the nearest peak, which is exactly what we'd expect. Thus far in our model, the population can never leave a peak, because doing so would be movement *against* the force of natural selection; and that is simply not allowed.

3.1. **Larger Populations.** The next step towards a more accurate model is to abandon the idea of treating a population as one representative point, and allow the model to portray an entire population of organisms spread out through phenotype space. We'll make the assumption that, with respect to each possible trait, the distribution takes on a normal curve (also called a Bell curve or a Gaussian function; refer back to Figure 1 for an example). We're going to create a population based off of a Gaussian, but instead of trying to model natural selection acting on the entire population at once, we're going to break it down into a matrix of values and have our 'selection gradient' act on each point separately. From this ground-up approach, we hope to create a usable model.

3.2. **The Population Matrix.** First, we'll create a single-peaked multi-variate Gaussian (in much the same way we made the AL, using an exponentiated quadratic form). Then, we'll create a matrix of points in phenotype space to use as a simplified representation of the population - a sort of low-resolution snapshot of the population in phenotype-space. Every point in that grid of points will also have a value associated with it: that is, the (x, y) values of that point in phenotype space fed into the original Gaussian function. So basically, the matrix will be comprised of points 1 to m , and each point will be associated with both an (x, y) value and the value of the Gaussian population function at that point - just a way of representing the population as a discrete rather than continuous distribution. It is important to note that these values - initially determined by the Gaussian function w - are fixed to that particular slot in the matrix from here on in. The (x, y) co-ordinates of that point can and will change, but if the jth entry in the matrix has a w 'population' value of 5 (representing the relative number of organisms with a certain phenotypic makeup), it will always have a value of 5.

3.3. **Selection Pressure.** The next step, then, is to superimpose the population down on the AL and watch it change. We'll find β - the gradient vector - in the same way as before, just taking partial derivatives. However, we'll use a slightly different process for iterating changes in population. This time, for a point $a_n = (x, y)$ we're going to use the formula

$$a_{new} = (x, y) + k\beta w(a_n)$$

where k is an arbitrary scaling constant and w is the population 'height' (the initial w value) at the point. So the outlying fringes of the population move more slowly, and the ones towards the middle, where the population is more concentrated, are pulled more strongly toward the peak.

3.4. **Change Over Time.** Initially, at least, the higher-density population points will pull out ahead, leaving the lower points trailing behind in a comet-like tail. However, as the denser points start getting close to the peak and the gradient vector begins to shrink, the lower density population points will start to catch up again. Unfortunately, however, as yet there is no restriction on the minimum variation present in population - with the inevitable conclusion that every point in the original population matrix will end up sitting on the peak, all clumped together!

3.5. **Inherent Variation.** Needless to say, this is not a desirable end result, nor a realistic one. In nature, with every population cycle (which is, of course, in reality a continuous process; we have merely modeled it iteratively for the sake of convenience) comes new varation. It is this variation, in fact, that gives natural selection fresh material to work with, and it is the lifeblood of the evolutionary process. (Obviously, it also prevents an entire population from ending up genetically identical, as our model would have it). Hence, a parameter needs to be added to represent this ongoing creation of population variation.

3.6. **Current State of the Model.** The basic idea here is to initially describe the population as a multivariate Gaussian, but model that Gaussian with a grid of points (the (x, y) values together with the values of the Gaussian itself, the 'density' of the population at that (x, y) value). These points would then take on a life of their own, moving across the adaptive landscape (AL) in the direction of the gradient vector at a rate proportional to their associated population density. Since this method results in a consolidation of all points as they head towards the AL peak, we will also need to program in a 'variation' routine that smears the population out, so to speak, with every generation (or, in our case, iteration).

4. Numerical Experiments

First off, it is necessary to create an array of points that we can use to model the multivariate Gaussian we're going to be using to describe population density. We create a Matlab script for this purpose. Here's roughly how it works:

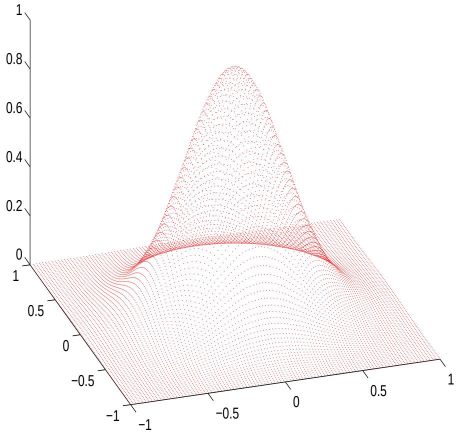

FIGURE 3. A simple multivariate Gaussian, modeled at a mesh of points.

(1) The user feeds in a number m that determines the size of the grid; the higher the number the better, but processing power rapidly becomes a problem. We will use $m = 50$, thus a grid of 2500 points, for our examples.
(2) Matlab's *meshgrid* function is used to create an appropriate array of evenly spaced out (x, y) points.
(3) An $m \times m \times 3$ matrix called *popgrid* is created. The x values from the meshgrid are used to fill up the 'bottom level' of the matrix, and the second level houses the corresponding y values.
(4) Each (x, y) pair is run through the multivariate Gaussian distribution, and the resulting value is placed in the slot 'above' that point in the matrix (i.e. it's at $(i, j, 3)$ for whatever i and j values correspond to that (x, y) pair. So now we have a matrix, $m \times m$ wide and 3 tall, where for a given representation point (i, j) $(i, j, 1)$ is the corresponding x value, $(i, j, 2)$ is the y value, and $(i, j, 3)$ is the value of the Gaussian evaluated at that (x, y) point.

Figure 3 shows the result of graphing all these points for $m = 50$.

4.1. **Putting them on the AL.** The goal, in order to better visualize how the population sits on the landscape, would be to plot a landscape and have this population distribution sit 'on top' of it. All that's necessary to do this is, instead of plotting (x, y, z) from the matrix, plot $(x, y, z + al(x, y))$ where al is the adaptive landscape fitness function. So basically we're keeping the x and y values the same for every point; we're just raising them all up so that the 'base' of the Gaussian distribution lies along the landscape itself. Coding this yields the result shown in Figure 4.1 (this is for a landscape with a peak at $(1, 1)$. Obviously the scaling factor is off (the population is much larger relative to the fitness peak than it should be) but that's just a matter of throwing in a constant to shrink down the population. The next step is to try to get the Gaussian samples to actually move relative to the fitness peak in the manner described earlier. This is done simply by treating

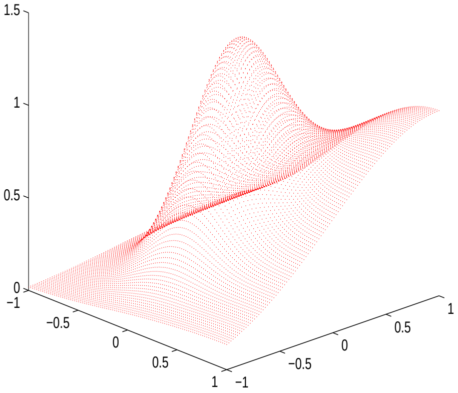

Figure 4. The same Gaussian, superposed over a (so-far invisible) AL.

each (x, y, z) 'slot' in *popgrid* (or vertical column, if you prefer to think of it that way) as an individual point in phenotype space. The gradient vector is evaluated at that point and used to move the point along the AL (except this time throwing in a scaling factor proportional to z, the population's 'density' at that point. Using a pair of nested for loops, the script cycles through all the points in *popgrid* and moves them one by one.

Starting with the Gaussian in Figure 4.1: Figure 5 is after five iterations, Figure 6 after 10 iterations, and Figure 7 after 50. The population is clearly converging towards the peak at $(1, 1)$.

4.2. **Iterations.** When we move the population across the landscape, we start seeing odd things happen. Figure 8 shows the result of 10 iterations, where an 'iteration' is simply the the addition of the vector

$$\frac{1}{5}\beta AL(p)^{(1/4)}$$

to the point p (where $AL(p)$ is the fitness of the point p, and β is the gradient vector at that same point).

Previously, we saw the population moving in a somewhat similar manner; however, the height of the peak caused it to move much more rapidly than other points – so much more so that the peak started looking hook-shaped, which really doesn't make any sense biologically. The addition of the 'fourth-root' for δp lessens the disparity between the highest points and the points around them, while keeping to the general theme that higher points move faster. This creates the effect of the peak points 'dragging' the lower points along with them, which is exactly what we want. The hook-shaped peak is much less likely to occur.

4.3. **Death.** This model still has several problems. First, points with very small fitness values don't die off like they should. There is a limit to the speed of evolution

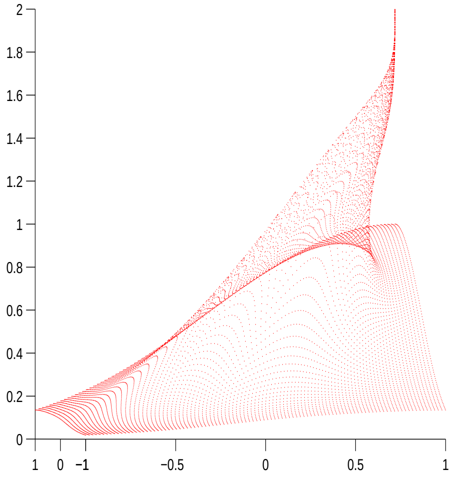

FIGURE 5. After 5 iterations. Individuals near the trait mean move the farthest up the landscape towards the fitness peak.

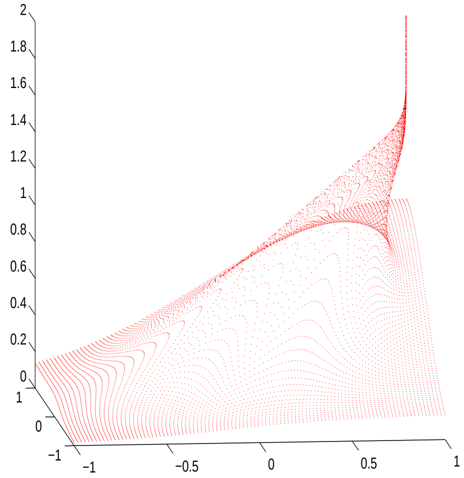

FIGURE 6. After 10 iterations. Some of them reach the peak.

(interesting side note: this is actually a defined concept, and the rate of evolution of a species is, naturally, measured in 'darwins'). Therefore if a point, no matter how population dense, is just too far from the fitness peak, it should die off.

Similarly, points with very small population densities, which would otherwise lie stagnant on the AL and wreak havoc with the system, should also die off. There's no provision for this in my model.

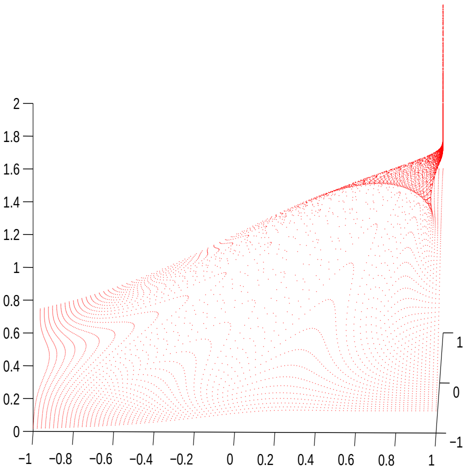

FIGURE 7. After 50 iterations. The population distribution begins to look very distorted.

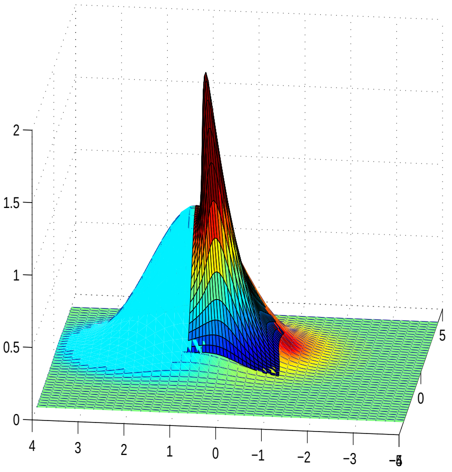

FIGURE 8. After 10 iterations, with a less potent natural selection force; in this graphic, the Landscape is visible 'underneath' the population.

4.4. **Collapse on to the peak.** With a little thought, it becomes obvious that after an absurdly high number of iterations every point ends up squarely atop the fitness peak. This is biological nonsense. A basic fact of evolution is that with every generation (read: iteration) comes a certain amount of variation without which natural selection would have no pool to select from. This variation would be represented in this model as a 'smearing out' of the population in phenotype space.

Highly dense population points should induce a population increase in the points around them with every successive iteration, thus giving rise to a well-distributed population at any point in the process rather than everything simply collapsing in on itself.

5. Modeling Variation

The model we've created so far moves a population towards the peak, but there's no spreading out from generation to generation. Eventually, the population will become homogeneous. We now implement a spreading algorithm that moves 'population points' away from one another. This is accomplished by finding the overall mean of the population, and then with each generation (iteration) move every point a bit away from that mean.

5.1. **A Numerical Example.**

First, we're going to find the overall trait mean of the population as a whole.

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i f(x)}{\sum_{i=1}^{n} x_i}$$

and

$$\bar{y} = \frac{\sum_{i=1}^{n} y_i f(y)}{\sum_{i=1}^{n} y_i},$$

the trait means for traits **x** and **y** respectively. Now, we create a vector

$$\delta p = (x - \bar{x}, y - \bar{y})$$

(in other words, the vector pointing from the overall mean of the population to the point (x, y)). So now we have a collection of vectors pointing from the overall trait mean to every point in the population. We can use this now to 'push' all the points in the population away from the mean, simulating spread in the population.

Previously, we just added in a multiple of the gradient vector with each iteration. This had the effect of 'pulling' each point towards the peak of the landscape. We're still going to do that, but we also add in δp (times a scaling constant), thus expanding the population.

5.2. **Results.**

As can be seen in Figure 9, the population has now moved over the hump, acting against the force of the gradient vector. Figure 10, however, shows that the spreading has taken over; the population has grown huge and dwarfs the landscape itself.

Apparently, then, we need to find a balance between accounting for the pull of selection and the spread of variation, so that one does not overwhelm the other.

6. Concluding remarks

We have come a fair ways towards creating a rudimentary model of the evolution of a population of organisms. It's time now to step back and consider how we got where we are today, where the model stands, and where everything is going.

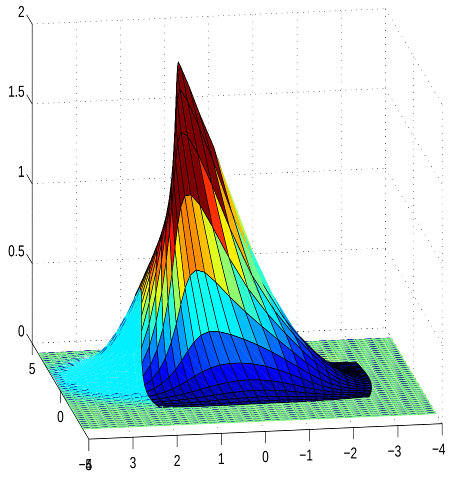

Figure 9. After 20 iterations, with an allowance for spread. The population has grown over the peak, something that was impossible before.

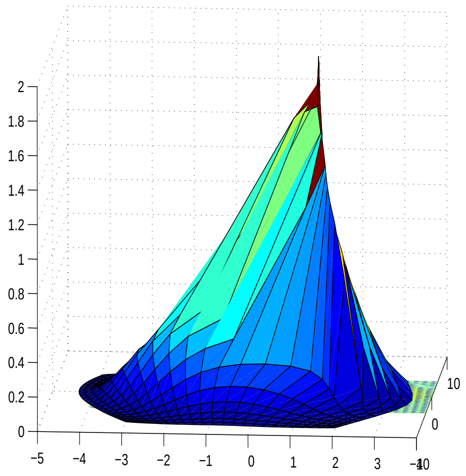

Figure 10. After 50 iterations. The population has grown over the visible landscape.

6.1. **Groundwork.** First, we went through some basic biology, and defined terms such as fitness and phenotype. From there, we talked about the concept of the Adaptive Landscape, and how we can picture natural selection and other forces of evolution moving that population about in 'phenotype-space.' The next step, naturally, was to translate this heuristic into something from which we could construct a viable computer simulation. Using an array of data points, each data point representing both a certain set of phenotypes and the number of organisms in the population with those phenotypes, we put together a population.

But this population was just floating in limbo. We constructed a simple AL using a Gaussian distribution, and overlaid the population on it (this was possible since both the landscape and the population array are based around co-ordinates in phenotype space). Next, to move beyond static to dynamic. We decided to use the gradient vector of the adaptive landscape as the direction of natural selection (which makes sense, since in this case the gradient points towards the direction of maximum fitness increase). So every generation, each point in the population array gets moved a certain distance along the AL's gradient vector.

However, after enough generations, it's clear that this leads to a complete collapse of the population - every point ends up at a fitness peak and can never leave. So we introduced a smearing factor, an expansion of the population in every direction in phenotype space with every generation. This represents genetic variability that is invariably and repeatedly introduced with every new generation, and it has the effect of spreading the population out no matter where it lives on the Adaptive Landscape.

We had a population, a landscape, and two evolutionary forces: selection and mutation (the underlying cause of the spread). It was time to run the model.

6.2. **Where we are now.** Individually, the components worked as desired. The population array idea lends itself well to a point-by-point parsing of the population as a whole, and it's not hard to graph the population on top of the landscape. Selection and drift also, individually, have pretty much the desired effect.

The problem lies in the last two components, and it is one of balance. For a small number of generations, the population moves as one expects: dense population points in the array move quickly towards the local fitness peaks, and outliers edge their way in the same direction. The population points spread out as they go, as well.

However, after any more than just a few generations, the population moves up towards a local peak and, once there, either a) explodes in every direction or b) collapses in on itself.

6.3. **Where to go.** The obvious first step is to alter the model so that, once a population is perched on a peak, the two forces of natural selection come to an equilibrium and the population hovers around the peak, maintaining a certain variability.

However, this is easier said than done. Better results may be obtained by 'tweaking' the constants of proportionality that govern selection and spread, but it seems cleaner to be able to derive the correct (or at least *some* correct combination) from first principles.

Other features - such as monitoring absolute population size, allowing multiple peaks, speciations, extinctions - can be added down the line, but for now, the balance issue needs to be corrected before going any further. More research is needed to figure out how to proceed.

6.4. **Applicability.** There is an overriding question that has loomed over everything we've done so far: is this even useful? Can we, with these vast oversimplifications, actually produce a computer model that gives us any sort of useful insights? It's a difficult question to answer. Certainly, the model described above, with only two different phenotypes measured and a very simplistic population model, would be only useful as a teaching tool or visualization aid. No population of real-life organisms, however completely controlled, would submit to such a bare-bones modeling.

However, by gathering an intimidatingly large amount of empirical data, it may be possible to create a many-dimensional phenotype space that actually does provide enough descriptive power to model, at least generally, a population. Even more empirical evidence would be needed to research the landscape itself - how does it change, at what timescales, is it predictable in any way? The landscape is a very elusive thing. If you model it based on how you see a species reacting, you're simply creating what you expect to see rather than actually predicting any outcome. The trick would be to model a landscape based on evidence, and then observe a change in the *real* environment and try to reflect those changes in your model. If the population reacts in a way consistent with the simulation, that would be very promising news.

For the time being, however, the Adaptive Landscape remains almost completely in the realm of the theoretical. Though not yet useful as a computational or predictive tool, it is nonetheless highly useful to gain a conceptual understanding of evolutionary change on a population level.

References

[1] Arnold, Pfrender and Jones. Genetica 112-113:9-32, 2001