2022

Sensitivity Analysis of Basins of Attraction for Gradient-Based Optimization Methods

Gillian King
Bowdoin College

Sensitivity Analysis of Basins of Attraction for Gradient-Based Optimization Methods

An Honors Paper for the Department of Mathematics
By Gillian King

Bowdoin College, 2022

Acknowledgements

I would like to thank Professor Adam Levy for his continued support in my goals as a researcher and student, and Sonia Shah for being my research companion. I would also like to thank my friends for their coding assistance and excitement about helping my research dreams come to life. Finally, I would like to thank my family for their encouragement, love and support.

Contents

1. Abstract

This project is an analysis of the effectiveness of five distinct optimization methods in their ability in producing clear images of the basins of attraction, which is the set of initial points that approach the same minimum for a given function. Basin images are similar to contour plots, except that they depict the distinct regions of points–in unique colors–that approach the same minimum. Though distinct in goal, contour plots are useful to basin research in that idealized basin images can be inferred from the steepness levels and location of extrema they depict. Effectiveness of the method changes slightly depending on the function, but is generally defined as how closely the basin image models contour information on where the true minima are located, and by the clarity of the resulting image in depicting well-defined regions. The methods are tested on four distinct functions which were chosen to assess how each method performs in the presence of various challenges. This project ranks the five methods for their overall effectiveness and consistency across the four functions, and also analyzes the sensitivity of the methods when small changes are made to the function. In general, less sensitive and consistently effective methods are more applicable and reliable in applied optimization research.

Keywords: Basins of attraction, sensitivity analysis, optimization, minimization, Line Search, Newton's Method

Introduction

2. Basins of Attraction

In the mathematical field of dynamical systems, attractors are a set of states toward which a system tends to evolve. Basins of attraction are visual representations of many simulations of individual points and their evolution toward equilibria. Figure 1 below shows a dynamical system of two equilibria, one at (0,1) and another at (.45, 1.2). The area of the basin shaded in red represents the points that evolve toward (.45, 1.2), and the blue band on the left of the image represents the points that evolve toward (0,1).

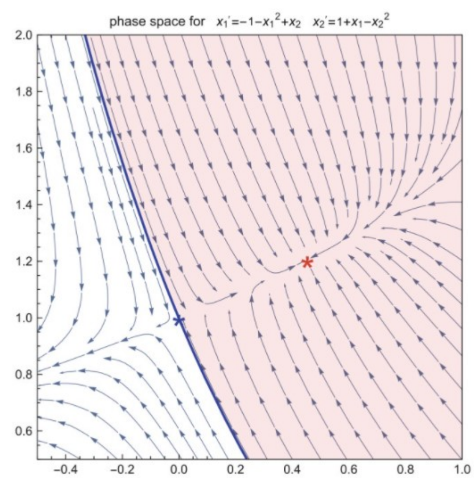


Figure 1. Basin of Attraction for a Differential Equation (Levy 2019, 25)

The application of basins of attraction to optimization research is relatively new and the predominant focus of my research. The basin images I produce in my research are rooted in optimization methods to find minima of functions. Thus, though distinct, basins of attraction in dynamical systems are a useful analogy to motivate discussion about basins of attraction in optimization research.

3. Overview of Optimization Methods: Effectiveness and Efficiency

Five optimization methods were chosen to form comparisons on how effectively each method finds the minima of different types of functions. Effectiveness in optimization research is highly subjective, yet there are several qualities of basin images that suggest that some methods perform "better" or "more effectively" than others.

The most direct metric for basin effectiveness in this project is the ability of each method to accurately color the areas that approach the same minimum in a clearly-defined manner.

Contour plots are useful in that the ideal basin images for each function can be inferred from the maxima, minima and steepness levels they depict.

For example, one function used to evaluate the methods is: $\sin(x) \cdot \cos(y) + \cos^2(y)$. Note the shape of the contour plot zoomed out and in the range $x \in [-4, 4]$ and $y \in [-4, 4]$:

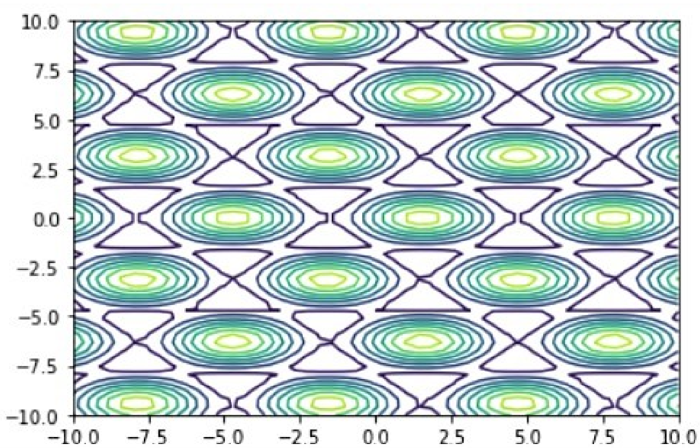


Figure 2. Contour Plot for $\sin(x) \cdot \cos(y) + \cos^2(y)$

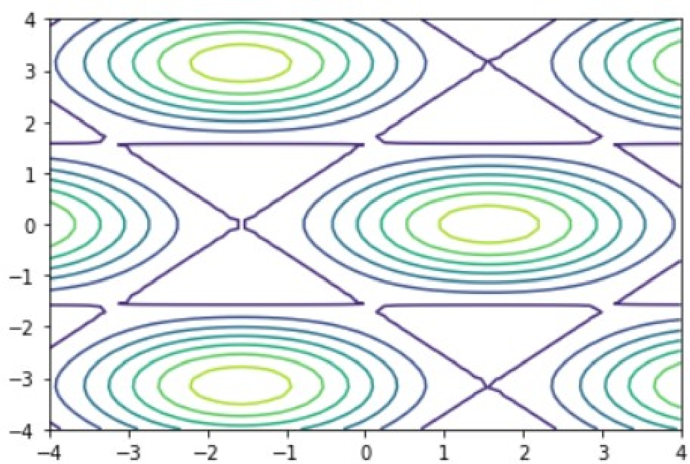
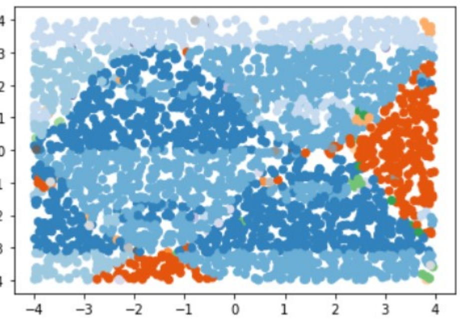
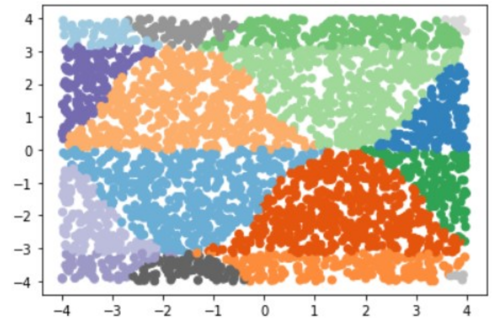


Figure 3. Zoomed-in Contour Plot for $\sin(x) \cdot \cos(y) + \cos^2(y)$

The function has infinite minima and maxima, and the distinct minima are found inside each purple triangle. The below two figures are side-by-side comparisons of the basins of attraction for the Steepest Descent Method (SD) and Gradient Descent with Momentum Method (GDM), which will be referenced and explored in more detail throughout this paper. The color of each point in the basin image corresponds to the minimizer that method approached using the given method, a process which will be explored later in the paper. Note the difference in clarity (mixed colors or undefined boundaries) of the distinct groupings of colors around the minima shown in the contour plot:



(A) Basin for Steepest Descent



(B) Basin for Gradient Descent with Momentum

In image A, points in similar regions seem to correspond to the same minima in a way that mirrors the minima (purple triangles) in the contour plot. As previously mentioned, the basin images will not directly resemble the contour plots because they are intended to represent different patterns. However, in an ideal situation, the basin images would show distinct, clearly-defined groupings of points that are near the same minimizers. Intuition on where these areas are is provided by the contour plots, but in general, the better the basin images follow the intuition given through the contour plots, the higher the accuracy of the method.

Image B illustrates how the Gradient Descent with Momentum Method accurately finds the distinct central minima and colors them uniquely, as each individual region near a minimum should be colored differently. My research would thus classify GDM as a more effective method for this function than SD.

A key aspect of my research is how the methods are affected by slight changes in the functions. In general, if small changes to the function (that do not drastically affect where the minima are) alter the basin image dramatically, the method would be classified as highly sensitive. Further discussion will follow about the intersection of sensitivity and effectiveness, but in general, highly sensitive methods are less effective at producing accurate basins.

Within individual runs of the optimization method, efficiency can be classified by the number of steps taken to find the minimizer, and the degree of oscillation between steps. In general, if a run of one method took many small steps, and a run of another took a few larger steps to find the same minimizer, the second run would be deemed ‘more efficient’ than the first. For the sake of this project, however, efficiency only decreases the integrity of the method if the run-time is significantly affected.

4. Overview of Optimization Methods: Steepest Descent with the Golden-Section Search

The first method chosen is the Steepest Descent Method with the Golden-Section Search (SD). This method is an iterative procedure to generate a point from another using a step of a certain length, α, in the direction of greatest descent.

Procedure: Define x_{k+1} as the point we are generating, and $\nabla f(x_k)$ as the direction we choose next, the negative gradient.

1) Start with an initial guess.
2) Update guess with the formula $x_{k+1} = x_k + \alpha \nabla f(x_k)$
3) Repeat while absolute value of the negative gradient is larger than some tolerance value, while replacing the initial guess.

The method gets its name from the step length, α, in the direction of greatest descent. Note the following procedure which finds the optimal alpha using Golden-Section search.

Procedure to find the most optimal α:
First, consider the following image and description:

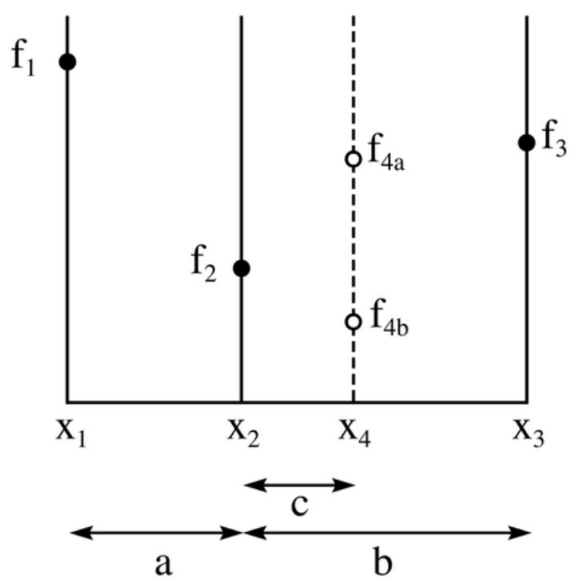


Figure 4. Golden Search Visual Representation (Wikipedia 2022)

The image above represents one run of the Golden-Section search, where the function values at x_1, x_2 and x_3 have already been evaluated. Because f_2 is a smaller value than f_1 and f_3, a minimum in this region necessarily falls between x_1 and x_3. The next step in the procedure is to consider a new point, f_4, which should lie in the biggest interval in consideration (here, because the interval $[x_2, x_3] > [x_1, x_2]$, we place f_4 in between x_2 and x_3).

If f_4 is larger than f_2, as shown by f_{4a}, then the minimum lies between x_1 and x_4. If f_4 has a smaller value than f_2, as shown by f_{4b}, then the minimum lies between x_2 and x_3. This process is repeated to produce guesses that are closer and closer to the true minimum.

Thus, the new search interval is either $[x_1, x_4]$ with a length of $a + c$, or $[x_2, x_3]$ with a length of b. The algorithm requires that these intervals be equal, so b must necessarily equal $a + c$. With manipulation, $b = (a + c)$ becomes $(x_3 - x_2) = (x_2 - x_1) + (x_4 - x_2)$ which is the same as $(x_3 - x_2) = x_4 - x_1$ or $x_4 = x_3 - x_2 + x_1$.

Note that the original spacing of x_2 between x_1 and x_3 matches the spacing of the triples x_1, x_2, x_4 or x_2, x_4, x_3. By maintaining this same proportion, the point in consideration is guaranteed to not be too close to either point on the boundary, and that the interval width shrinks by the same proportion every time. With the mathematics fully worked out, the ratio $\frac{b}{a} = \frac{1+\sqrt{5}}{2}$, which is the golden ratio (Wikipedia 2022).

In the contour plot below, the blue lines represent the individual steps the method took to find the minimizer, starting from a random point (the ‘x’ at about (-1.7, 0.25)). In addition, note that when the method finds the lowest point in the direction of the negative gradient, it turns 90 degrees to continue the process in a new direction. Here, the method approached a true minimizer near the point (-1.6, 1.0). Using the above algorithm, the method makes a turn when the step length is optimized in that direction.

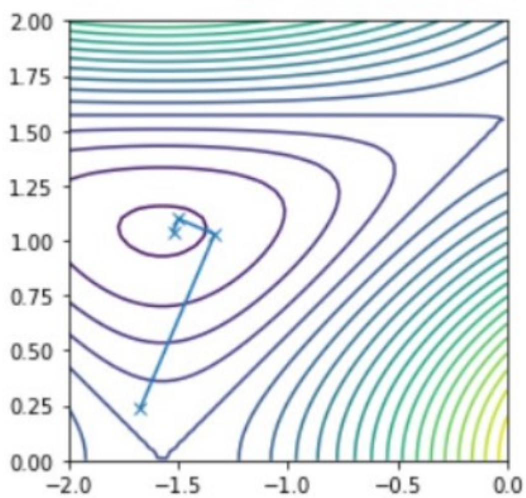


Figure 5. One Run of Steepest Descent Method

5. Backtracking with the Armijo-Goldstein Condition

The Backtracking Methods are iterative procedures to determine the amount to move in a particular direction. Like the SD method, these methods use the negative gradient, but they differ from SD in the amount they choose to move in the direction of the negative gradient.

Procedure:
While $f(x_k) - (f(x_k - \alpha\nabla f(x_k)))$ - $\frac{\alpha}{2}(\|\mathbf{v}\|)^2 < 0$ is true, repeatedly multiply α by β, effectively halving the distance along the direction in consideration. Thus, the full procedure becomes:
1) Fix parameters $\alpha = 1$ and $\beta = 1/2$ where α is the step rate, and β halves α after each iteration.
2) While the condition $f(x_k) - (f(x_k - \alpha\nabla f(x_k)))$ - $\frac{\alpha}{2}(\|\mathbf{v}\|)^2 < 0$ is true, repeatedly multiply α by β, effectively halving the distance along the direction in consideration.
a) If the condition becomes false, restart process in new direction.
3) Continue by repeatedly finding the new step length in each direction.
4) Stop the procedure when the gradients become small.

Note the frequency of small steps in the plot below:

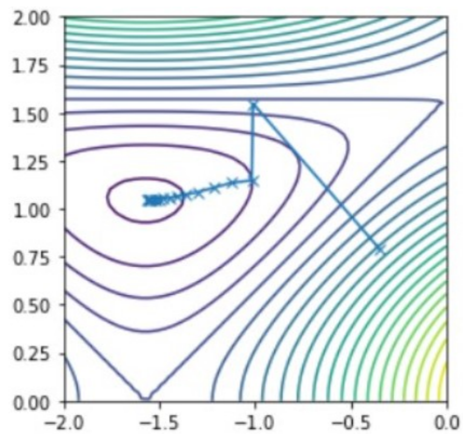


Figure 6. One Run of Backtracking Method with Armijo Condition

With regard to the function being minimized here, the run-time of the Backtracking with the Armijo-Golstein condition is not significantly affected by the decrease in efficiency of the steps. As mentioned in the section on effectiveness, parameters for effectiveness on individual runs of the method include the amount of steps it takes to approach the minimizer, and the degree of oscillation of the steps. As compared to one run of Steepest Descent as show in Figure 4, my analysis would classify this run of Backtracking as 'less effective', as more steps are taken that oscillate at a higher rate.

6. Backtracking without the Armijo-Goldstein Condition

Procedure:

1) Fix parameters $\alpha = 1$ and $\beta = 1/2$ where α is the step rate, and β halves α after each iteration.
2) While the condition $f(x_k) - (f(x_k - \alpha\nabla f(x_k))) < 0$ is true, repeatedly multiply α by β, effectively halving the distance along the direction in consideration.
a) If the condition becomes false, restart process in new direction.
3) Continue by repeatedly finding the most optimal step length in each direction.
4) Stop the procedure when the gradients become small.

Note that the following plot shows how the method does not use the turn decision based on the negative gradient:

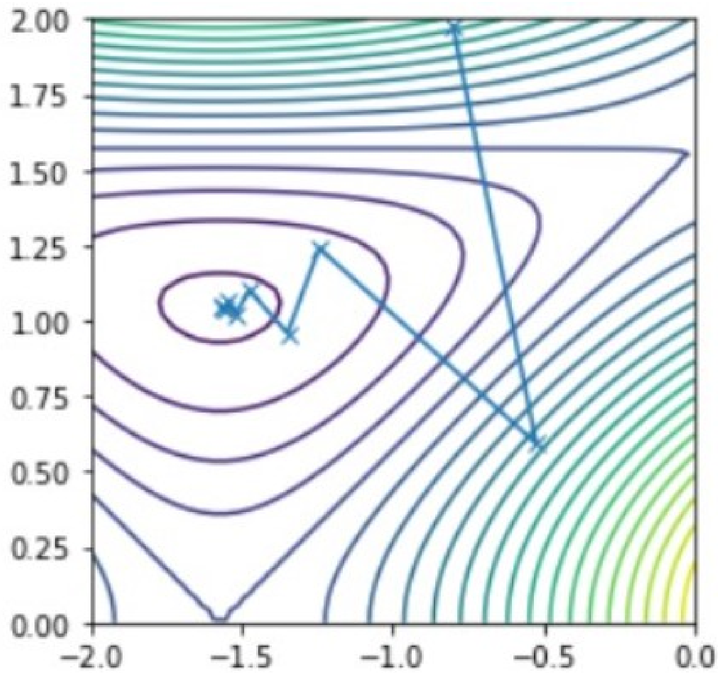


Figure 7. One Run of Backtracking Method

In Figure A below, the run takes fewer steps, but the steps oscillate more dramatically than in Method B. The consistency of the steps in Method B, despite the frequency, may make Method B more effective depending on the optimization problem in question and which parameters are more or less important in different applied contexts.

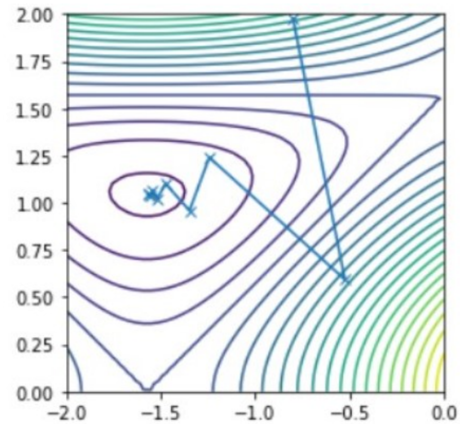


Figure 8. Image A: One Run of the Backtracking Method

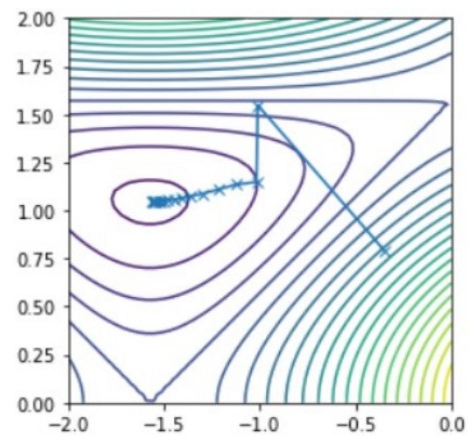


Figure 9. Image B: One Run of Backtracking Method with Armijo-Goldstein Condition

7. Gradient Descent with Momentum Condition

Gradient Descent with Momentum is similar to SD, except that it uses a parameter called the learning rate to determine the step size after each iteration. In addition, the momentum value attempts to use information from the last guess in order to propel the method forward (in general, it can help the method bypass plateaus or shallow local minima). While the code has not exceeded the set maximum iterations, and the difference in function values between two successive iterations is not below a set threshold, the method continues to find the next guess.

Procedure:

1) Start with an initial guess.
2) Update guess with the formula $x_{k+1} = x_k\gamma - \epsilon\nabla f(x_k)$, where γ is the momentum value and ϵ is the learning rate
3) Repeat while absolute value of the negative gradient is larger than some tolerance value, while replacing the initial guess.

Note the drastically high frequency of small steps in the following run, even compared to Backtracking with the Armijo-Goldstein Condition:

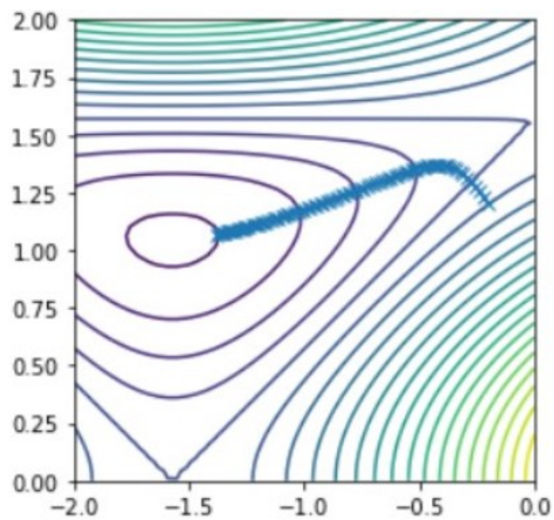


Figure 10. One Run of Gradient Descent with Momentum

As compared to the Backtracking Method with the Armijo-Goldstein condition, this method takes a large amount of steps due to the fixed, and very small, learning rate parameter. When the learning rate of a method is too small, however, the efficiency of the method can be compromised.
The optimal learning rate and momentum values for each function-method pair are different as shown in the sensitivity analysis portion of this paper.

8. Newton's Method

The general procedure for this method is similar to the other four, but has some significant differences. The procedure is:

1) Start with an initial guess.
2) Update guess with the formula $x_{k+1} = x_k - \nabla^2 f(x_k)^{-1} \nabla f(x_k)$. Note here that the inverse Hessian, a square matrix of second-order partial derivatives, replaces the step length in the SD Method.
3) Repeat while absolute value of the negative gradient is larger than some tolerance value, while replacing the initial guess.

For this particular function, this method dramatically fails to converge to the proper minima. There are a few possible reasons for this, which will be referenced throughout the remaining sections:
1) This method does not favor minimizers over maximizers. Thus, with a function with many maximizers as well as minimizers, the method will prioritize locating both.
2) The method cannot progress when the second derivative of $f(x)$ is zero, or when the determinant of the Hessian Matrix is zero.
3) The method does not work well when the function has many oscillations.
4) The random guess was not close to the minima.

9. General Research Procedure: Finding Minima in Python

Finding the minimum that an individual point approaches, and simulating this process many times, is generalized through the following steps. A **run** is defined as the process starting from the original, randomized point and ending at a minimum.
1) Start with a randomized guess in the (x,y) coordinate plane.
2) For the first run, record the minimum the point approached. Assign the random point a unique color.
3) On the next run, if the new random point approached the same minimum as another point within a set tolerance value*, color it the same. If not, create a new color to represent the new minimum.
4) Simulate this process many times, creating a basin image that records all the randomized points and the minima they approached, as indicated by the color of the points.

***A Note on Tolerance:**

A run of the method is controlled by two tolerance values: the tolerance for the magnitude of the gradient at the final point, and the tolerance for the similarity required of points to be considered ‘the same minimum.’ If the gradient changes very little between the steps of a run, it is likely, with exception, that the run is approaching a minimum. If the tolerance is too constrictive, it may fail to recognize all points that should be considered the same minimum. If the tolerance is not constrictive enough, however, the method may incorrectly include inaccurate points as minima.

The tolerance values were slightly adjusted throughout the analyses of the methods to optimize the quality of the basin image. If a method did not respond more positively to a slight tolerance change, a sub-optimal value was used to further illustrate the sensitivity of the method.

Note the basin plot images for 10, 100, and 1000 simulations for the function $\sin(x) \cdot \cos(y) + \cos^2(y)$ using the Backtracking Method with the Armijo-Goldstein Condition:

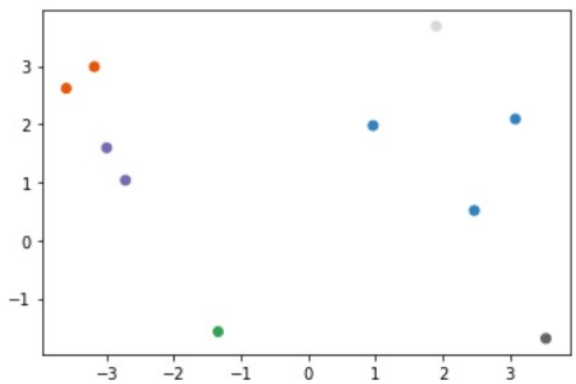


FIGURE 11. 10 runs of Backtracking with Armijo-Goldstein Condition

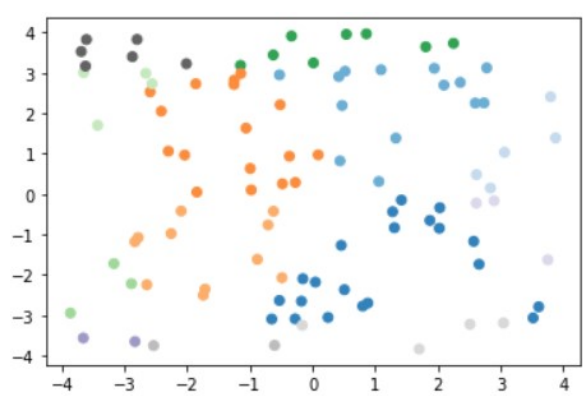


FIGURE 12. 100 runs of Backtracking with Armijo-Goldstein Condition

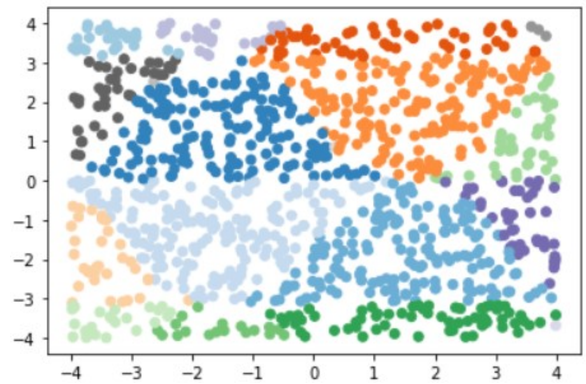


FIGURE 13. 1000 runs of Backtracking with Armijo-Goldstein Condition

Note that points of the same color approached the same minimum. The evolution of the basin image from 10 points, 100 points and 1000 points shows the process of the method finding a new minimum, coloring it a new color, and using that color for points that evolve to the same minimum.

10. A Note on Basin Colors

This project uses colormaps to produce aesthetic color schemes that illustrate the locations of the minima. Colormaps are accessed through the library Matplotlib, and are available in several forms. This project relies on qualitative colormaps, which are mostly used in cases where unique colors or color gradients are not important in mapping trends. For example, a project that uses a colormap to illustrate the amount of air pollution in specific neighborhoods of a city would not use a qualitative colormap, because the colors would specifically be assigned to different levels of pollution. For this project, the colors themselves do not hold any specific meaning. Rather, it is the distinct boundaries that the colors form that are important in establishing the locations of the minima.

'tab20c' is the predominant colormap used in this project. Its unique range of colors are shown below:

FIGURE 14. tab20c colormap

There are twenty distinct colors available through the 'tab20c' colormap. To produce a new color, the algorithm uses on an arbitrary increment value. The increment value that is set does not matter because qualitative colormaps automatically normalize the color values in the range [0,1] depending on the amount of colors used. In the case where a method uses less than the maximum amount of colors offered by a qualitative colormap, this normalization process will ensure that each minimum has a unique color. In the case where a method requires more than 20 colors to illustrate where the minima are, colors will be repeated even though they correspond to different minima. Even though this is the case, there is a low probability that adjacent colors are the same. Even with this drawback of qualitative colormaps, they are highly useful for their capacity to create images with highly distinct and vibrant colors.

11. Functions Used to Evaluate the Methods

The following four functions were used to evaluate the methods in this research project. These functions were chosen for a variety of reasons, predominately to show the strengths and weaknesses of the different methods:

1) $\sin(x) \cdot \cos(y) + \cos^2(y)$
2) $\frac{1}{4}(1 + 4(x^2 + (-1 + y)^2))(x^2 + (1 + y)^2)$
3) $(x - 4 - y)^2$
4) $x \sin(x) + y \sin(y)$

1. $\sin(x) \cdot \cos(y) + \cos^2(y)$ has a unique structure in that its minima and maxima form a pattern that repeats indefinitely–every maxima is directly surrounded by 6 minima. For reference, a limited range of its contour plot and a plot of the function are included below (note again that the indivdiual purple triangles represent the minima, and the green ovals represent the maxima):

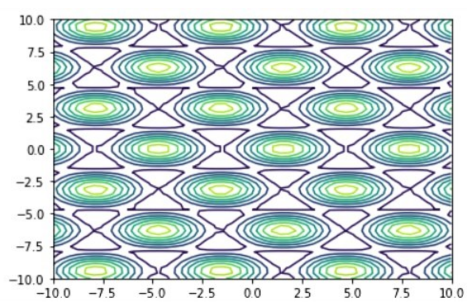


Figure 15. Contour plot of $\sin(x) \cdot \cos(y) + \cos^2(y)$

This function is useful to test how well a method works with a function of infinite minima spread consistently around infinite maxima.

2) The function $\frac{1}{4}(1+4(x^2+(-1+y)^2))(x^2+(1+y)^2)$ has two minima close together, one local and one global. The true, global minimum is found at (0, -1), and the local minimum is found near (0, .85).
Below are the functions' contour plot and plot:

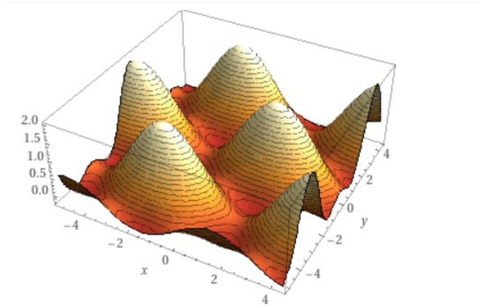


FIGURE 16. Plot of $\sin(x) \cdot \cos(y) + \cos^2(y)$

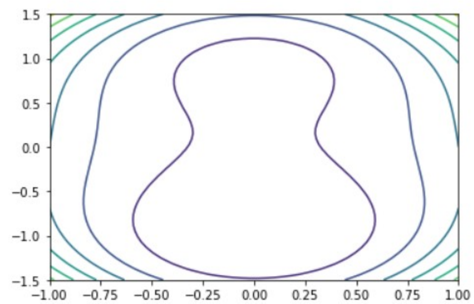


FIGURE 17. Contour plot of $\frac{1}{4}(1 + 4(x^2 + (-1 + y)^2))(x^2 + (1 + y)^2)$

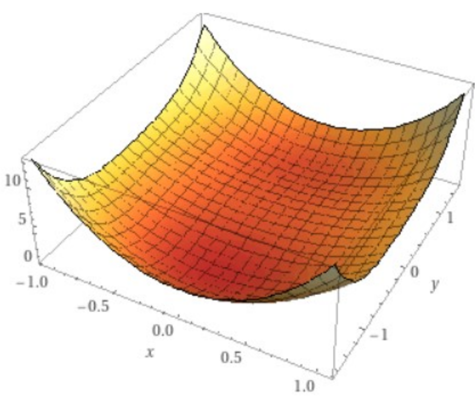


FIGURE 18. Plot of $\frac{1}{4}(1 + 4(x^2 + (-1 + y)^2))(x^2 + (1 + y)^2)$

This function is useful to test how the methods respond to the presence of a local minimum in close proximity to a global one. In this case, the methods are evaluated on how well they pass through the local minimum to find the global minimum. Functions of this form (with local and global minima in close proximity) are useful measures of effectiveness for the methods. Ideally, a function's random initial points would bypass the local minimum to find the global one.

3) The function $(x-4-y)^2$, like $\sin(x)\cdot\cos(y)+\cos^2(y)$, has infinite minima, which are found at the bottom of a symmetric, half-pipe structure. Below is its contour plot and plot:

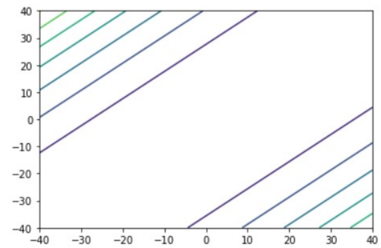


FIGURE 19. Contour plot of $(x-4-y)^2$

Note that the whitespace in the middle of the contour (between the purple lines) represents the infinite minima that lie in the middle of the function. For reference, a plot of the function is included:

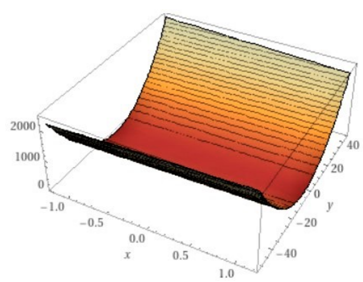


FIGURE 20. Plot of $(x-4-y)^2$

This function is useful in evaluating how well the methods identify the infinite minima of such a half-pipe structure. In the case of this function, when the 'x' is held constant, all the points on each line approach the same minimum.

4) The function $x\sin(x)+y\sin(y)$ has no global minima, but has a condense grouping of local minima of varying depths around the origin. Note the structure of its contour plot and plot:

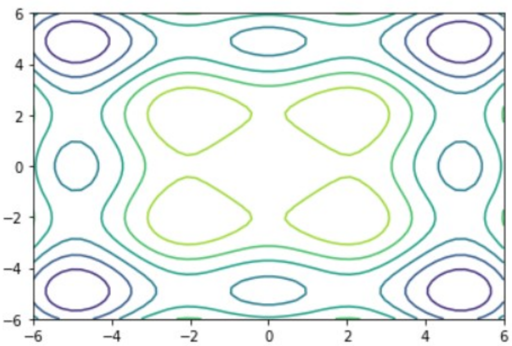


FIGURE 21. Contour plot of $x\sin(x) + y\sin(y)$

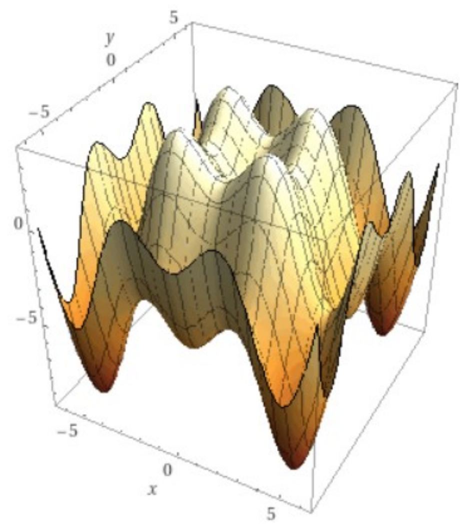


FIGURE 22. Contour plot of $x\sin(x) + y\sin(y)$

The contour plot reveals the 9 minima of varying depths in the range $x \in [-6, 6]$ and $y \in [-6, 6]$. This function is useful as a means to explore how the methods respond to many minimizers in a limited range of differing depths. The methods are evaluated based on their ability to distinguish the many minima in the small range.

12. Effectiveness of Methods on Function 1, $\sin(x) \cdot \cos(y) + \cos^2(y)$, and a Discussion on the Order that the Methods are Run

To evaluate the effectiveness of the methods on Function 1, the following basin images for the five methods are considered. However, first note the following important caveats of the procedure for locating and coloring the minima. First, note that Steepest Descent on its own is not as effective as Backtracking with and without the Armijo-Goldstein Condition for this function. Because SD was coded before any other method, it stores information on the minima that affects the rest of the methods if they run in the same kernel. Examine the following basin image from 3,000 simulations from the SD method:

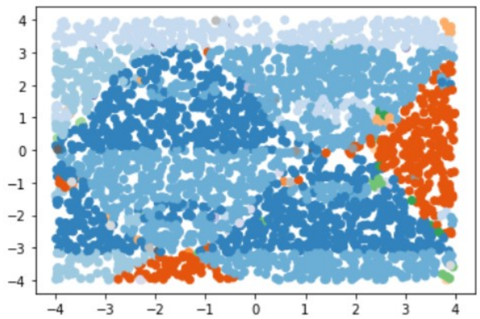


FIGURE 23. Basin Image for $x\sin(x) + y\sin(y)$

The function has four main minima in the region $x \in [-4, 4]$ and $y \in [-4, 4]$, minima 1-4, though some basins show nearby minima on the periphery. Note the following contour plots for reference, the first of which was coded to match the rectangular aspect ratio of the basin. The numbers correspond to distinct minima in the region.

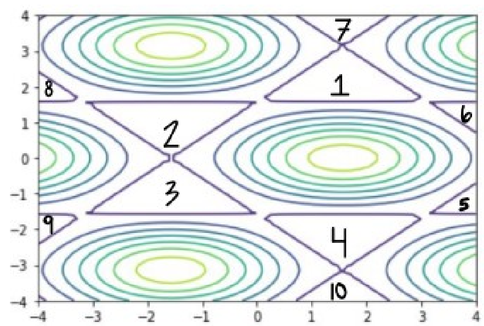


FIGURE 24. Numbered Contour Plot for $x\sin(x) + y\sin(y)$

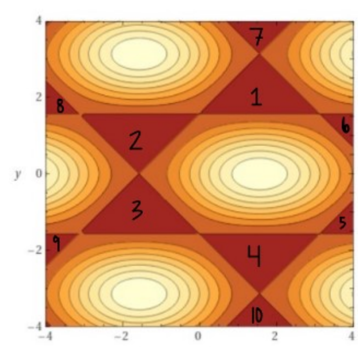


FIGURE 25. Numbered Contour Plot for $x\sin(x) + y\sin(y)$

Note the following about SD's effectiveness in finding the minima:

1) The method finds only one minimzer, not three, in regions 1, 3 and 10, though these regions contain distinct minima. The method also finds only one minimizer in regions 2 and 4, though these regions also have distinct minima.

2) The regions are not all homogeneous in color, though the boundaries are relatively well-defined.

3) Region 6 incorrectly shares a color with a different minimizer off-screen.

4) The top portion of the basin is homogeneous in color, though there are additional minima off-screen near that region.

12.1. **A Note on Order of the Methods as They Run in the Kernel.**

Examine the following basin image from 3,000 simulations of the backtracking method, with and without the Armijo condition:

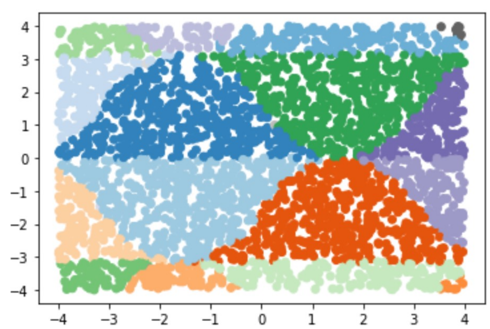


FIGURE 26. Basin Image for Armijo-Backtracking for Function 1

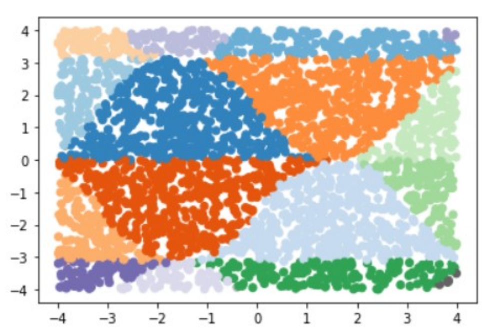


FIGURE 27. Basin Image for Backtracking without the Armijo Condition for Function 1

Because these are independent simulations, meaning that the kernel was cleared before running the method, the color scheme will be different due to the randomized sampling, but the areas in which minima are held should be the same. These methods are more clearly defined than SD, and locate the minima more effectively. Based on the parameters for effectiveness as laid out in the earlier sections of the project, the Backtracking Methods would be deemed more effective than SD for Function 1, assuming all methods are running independently.

What information does SD store that causes the other methods to be less effective when they run in the same kernel for this function?

The other methods are less effective when SD runs first mostly due to the tolerance values set to classify the endpoints of runs as preexisting minima. When the SD Method approaches an endpoint, the method parses a preexisting list of endpoints, and matches points within a set tolerance value. Specifically, the method will identify an endpoint as the same as another if its x-coordinate is within some number of the x-coordinate of a preexisting endpoint, and if its y-coordinate is within the same number of the y-coordinate that same preexisting endpoint. If not, it will declare the endpoint as a new minimum, and assign a new color to it.

When the Backtracking Methods run, the endpoints each run produces are classified based on their proximity to the endpoints that SD produced, which may not be an accurate measure of where the minima actually are. This is problematic not only because the Backtracking Methods potentially use incorrect information to locate the minima, but also because even small tolerance values can prevent the Backtracking methods from finding additional minima.

12.2. **Dependent Basins.** Note the pictures above for the independent simulations of the Backtracking Methods. Now note the qualities of the basins when SD runs first, as well as the SD basin, as shown below:

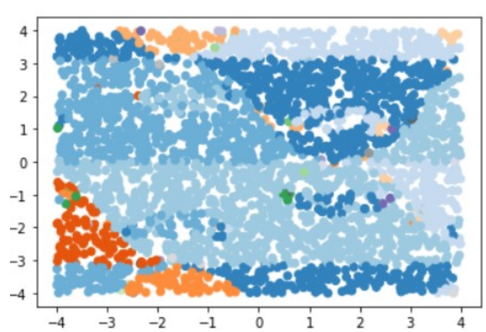


FIGURE 28. Basin Image for SD for Function 1

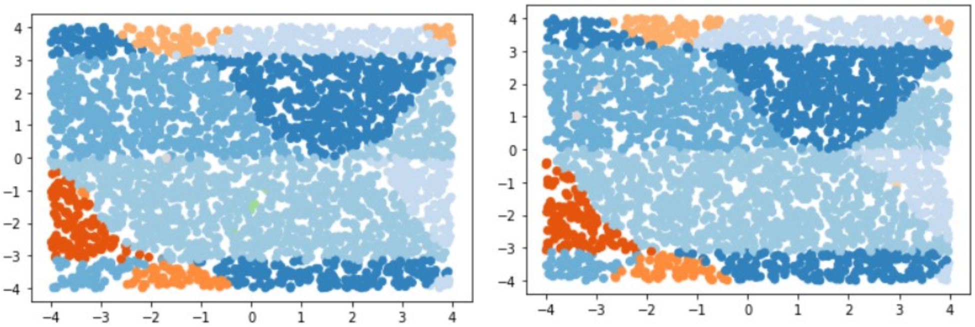


FIGURE 29. Left: Dependent Basin for Backtracking with the Armijo-Golstein condition; Right: Dependent Basin for Backtracking without the Armijo-Goldstein condition

When the Backtracking Methods run first, they do not have a dramatic effect on the SD basin:

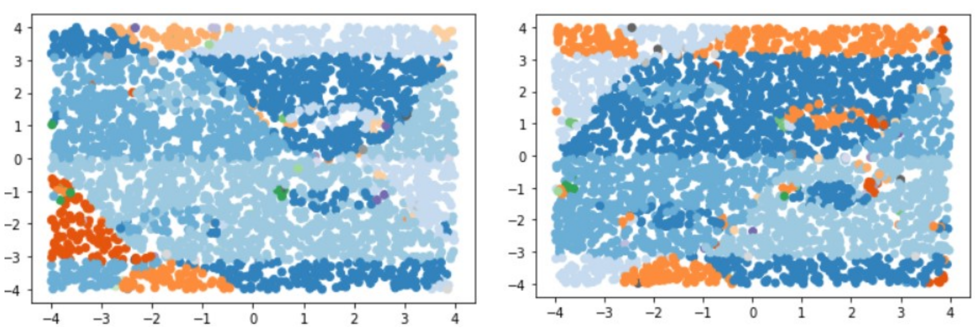


FIGURE 30. Left: SD Basin when run alone; Right: SD Dependent Basin

The images are similarly undefined and amorphous, and show that if a method is independently less effective than another, the order that the method runs is often negligible. Furthermore, the Newton Method is also largely inaccurate for this function, so the order that the methods are run has minimal effects on it. More discussion on order is to follow for the remaining functions, as different methods are more effective for different types of functions.

Below are the five basin images for Function 1 (all run independently because the first method, SD, was not as effective as other methods):

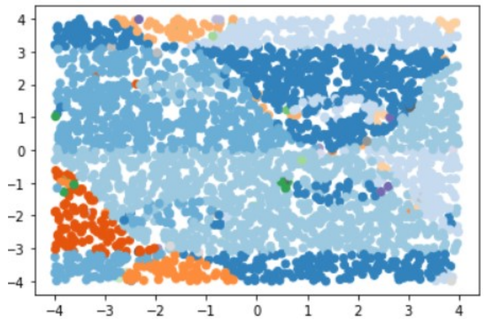


FIGURE 31. Basin Image for SD for Function 1

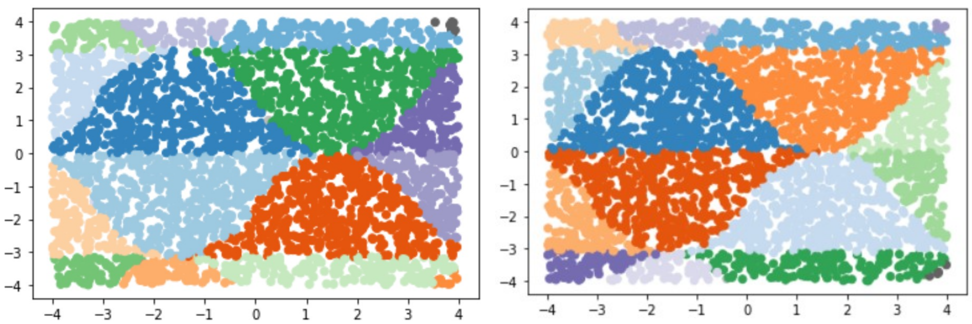


FIGURE 32. Left: Independent Basin for Backtracking with the AG Condition; Right: Independent Basin for Backtracking without the AG condition

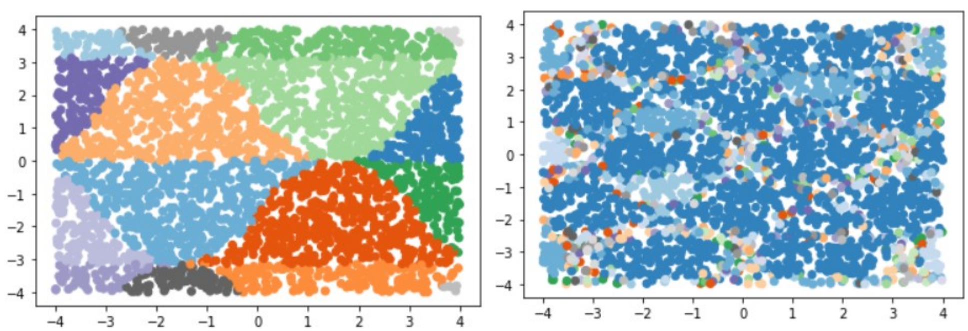


FIGURE 33. Left: Independent Basin for Gradient Descent with Momentum; Right: Independent Basin for Newton's Method

Note the following about the methods:

1) As previously mentioned, Newton's Method performs ineffectively when there are many maximizers as well as minimizers, the second derivative at a random point is zero, and the function has many oscillations. Function 1 exhibits these qualities, as its second derivative is zero at an infinite amount of locations (in the transitions between low-spots and high-spots) and the function has an infinite amount of oscillations that are very close together. Note also that the ring structures seem to correlate with the areas where the second derivative is zero.

2) The Backtracking Methods and the GDM method are the most effective here. The Backtracking Methods are highly effective because they decide the step length based on the most optimized outcome. In addition, GDM performs well with this function because of the low learning rate values chosen, which means that the steps between methods are small and steady in the direction of the negative gradient. On the other hand, SD chooses based on a predetermined ratio which can decrease precision, and Newton's Method fails for a variety of reasons due to properties of the function.

13. Effectiveness of the Methods on Function 2, $\frac{1}{4}(1+4(x^2+(-1+y)^2))(x^2+(1+y)^2)$

To evaluate the effectiveness of the methods on Function 2, the five basin images for each method are considered. First, recall the function's contour and plot, and note the location of the two minimizers, minimizer 1 at (0, -1) and minimizer 2 at (0, 0.85):

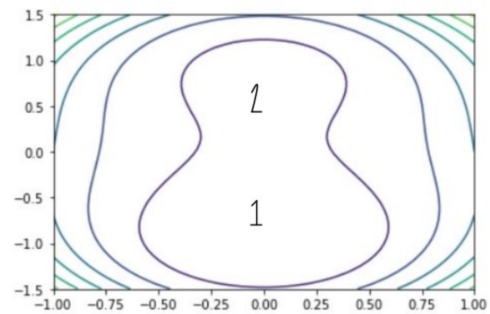


Figure 34. Contour Plot with Numbers for Reference

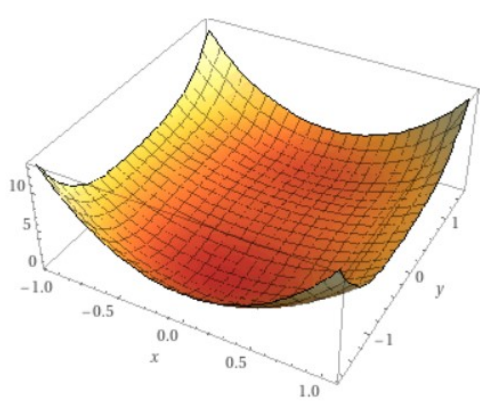


Figure 35. Function 2 Plot

An important consideration when evaluating Function 1 is the order that the methods are run, as SD does not locate as many minima as some of the other methods. In the case of Function 2, many of the methods find the same two minima, but differ in accuracy. In running the code, the methods need only find the **same** minima for the order to not significantly impact the performance of the method. For this function, dependent coloring is preferable only to keep the coloring consistent (i.e., blue and gray will represent the same minima when running different methods).

Consider the following basin images for Function 2. Note that the GDM Method is run with the parameters that optimize the method's effectiveness.

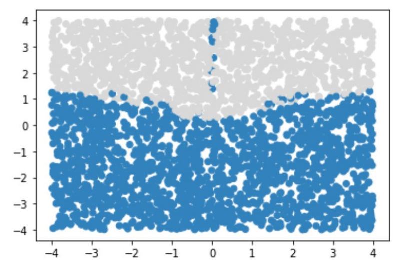


FIGURE 36. SD Basin Image for Function 2

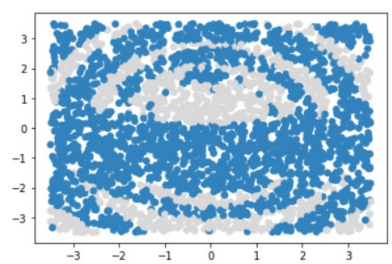


FIGURE 37. Backtracking with the AG Condition Basin Image for Function 2

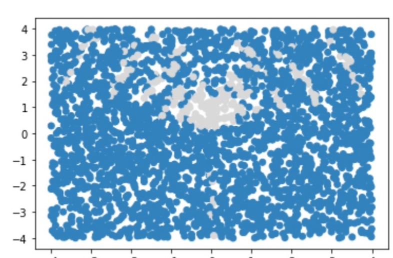


FIGURE 38. Backtracking Basin Image for Function 2

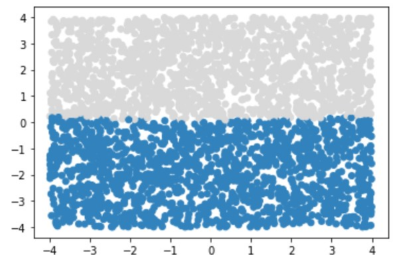


FIGURE 39. GDM Basin Image for Function 2

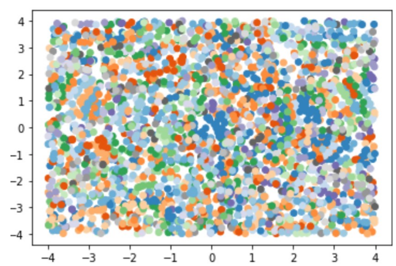


FIGURE 40. Newton Basin Image for Function 2

When evaluating the basins, is important to consider that this function was primarily chosen to evaluate the methods' ability to bypass a local minimizer that is in close proximity to a global one. In this image, the blue dots represent all the points that approached (0, -1), and the gray dots represent all the points that approached (0, 0.85).

Right away, it is clear that the Backtracking Methods are the most successful methods at bypassing the local minimizer, as the images are far more blue than the other basins. Both SD and GDM seem to locate where the minima are, but the points closer the local minimum for both methods have a more difficult time bypassing it. The SD basin, furthermore, shows a small blue region in the midst of the gray portion of the basin, which corresponds to the area where the negative gradient gets steeper as the points approach the local minimum. In areas where the function is steeper, the negative gradient gets longer, and the function may overshoot the closest minimum to some far away point. Here, the method is overshooting to find the global minimum, but in general such overshooting causes the methods to be less accurate in narrowing in on a minimum in a close region.

Furthermore, Newton's Method again largely fails in its effectiveness in locating the minima, which results primarily from two of the previously-mentioned reasons that Newton's Method fails. First, it was noted that the method does not work well when the function has many oscillations, especially when they are close together. In the case of this function, the local and global minima are in close proximity to one another, and thus it is likely that the method would fail when trying to locate them. Because the minima are located close together near the center of the basin, it is likely that the points off-center are affected by the distance from the minima.

In Summary:
1) Backtracking without the AG Condition performs the best for this function, as it is minimally impacted by the presence of the local minimum. Backtracking without the AG Condition performs slightly better than Backtracking with the AG Condition, as its basin contains less gray.
2) The GDM and SD methods perform reasonably well here, but are more affected by the prescence of the local minimum than the Backtracking Methods.
3) Newton's Method largely fails here due to the the close proximity and high oscillations of the mimima, as well as the distance of some of the guesses from the minima.

14. Effectiveness of the Methods on Function 3, $(x - 4 - y)^2$

The following basin images below illustrate the effectiveness of the various methods on this function. First, however, recall the contour and plot of this function.

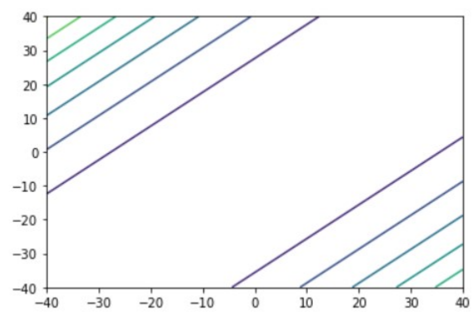


Figure 41. Contour Plot for Function 3

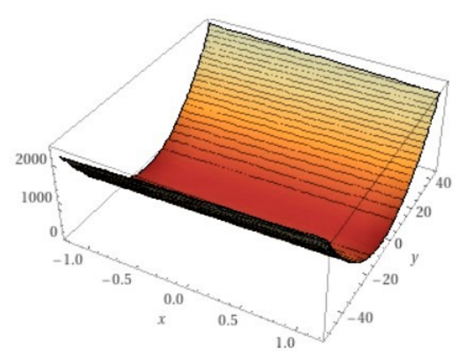


Figure 42. Function 3 Plot

There are infinite minima for this function, which lie in between the two raised ends of the function. Because of the limitations in the amount of colors offered by qualitative colormaps in Python, some colors are repeated in the range shown in the basin images below. However, because all three methods perform simiarly well, dependent running is preferred to keep the color scheme consistent. When the range is smaller (also shown below), the method produces less than 20 minima, and thus no colors are repeated.

The following basins illustrate how all the points along a line with the x-value held constant lead to the same minima in the center of the half-pipe structure. Note that the lines on the contour plot represent the varying heights of the plot. This means that the lines on the basins themselves (that document all the points on the line that lead to the same minima) will be shifted 90 degrees from the contour lines.

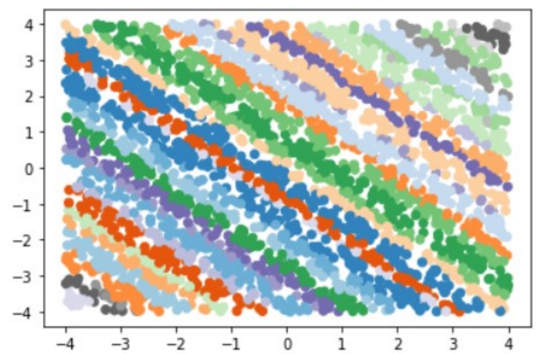


FIGURE 43. SD Basin for Function 3

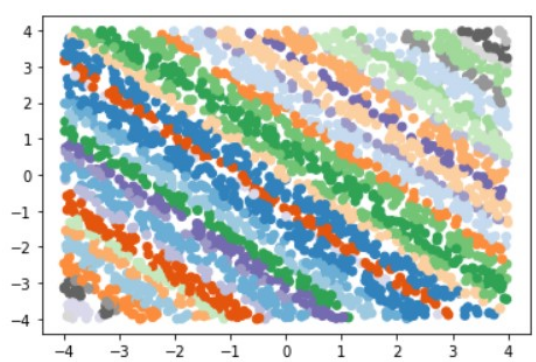


FIGURE 44. Backtracking Basin with AG Condition for Function 3

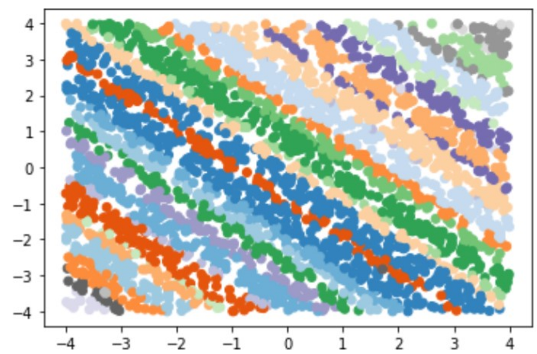


Figure 45. GDM Basin for Function 3

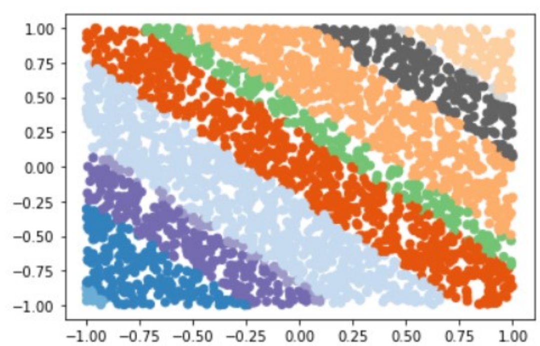


Figure 46. Zoomed-in Basin Plot for SD

The three basins shown above are consistent and accurate representations of the minima locations, and that no one method drastically outperforms any of the others. Thus, SD, Backtracking with the Armijo-Goldstein condition, and GDM perform well and relatively equally for this function. The final basin, labeled 'Zoomed-in Basin Plot for SD,' shows how the colormap reverts to its normal coloring scheme when there are less than 20 minima. In other words, colors are only repeated when necessary–in the case where the method identifies more than 20 minima. The varying widths of the color strips are a consequence of the tolerance values and order in which the minima are found.

Backtracking without the Armijo-Goldstein Condition and Newton's Method fail to produce basin images for this function. In the case of Backtracking without the Armijo-Goldstein Condition, future discussion will explore how the steepness of the function impacts the ability of this method to converge.

Newton's Method also fails to converge here, due to the fact that Newton's Method uses the inverse of the Hessian matrix, a square matrix of second-order partial derivatives, to compute the next step. The determinant for the Hessian matrix for this function is zero, which means that it does not have an inverse. Thus, the method cannot compute a new guess from an existing one.

In Summary:
1) SD, Backtracking with the Armijo-Goldstein Condition and the GDM perform similarly well for this function.
2) Newton's Method fails here, as the determinant of the Hessian matrix is zero.
3) Backtracking without the Armijo-Goldstein Condition fails here due to steepness issues.

15. Effectiveness of the Methods on Function 4, $x\sin(x) + y\sin(y)$

In order to discuss the effectiveness of the function $x\sin(x)+y\sin(y)$, first note its contour plot and plot in the range x ∈ [-4,4] and y ∈ [-4,4]:

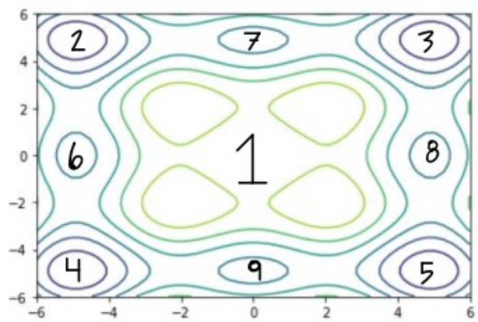


Figure 47. Numbered Contour Plot for Function 4

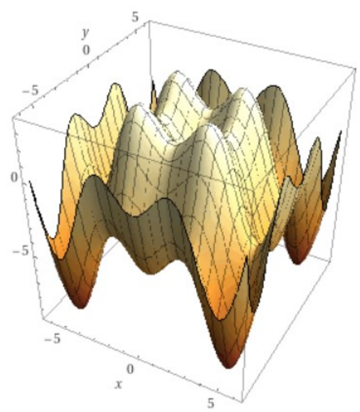


Figure 48. Function 4 Plot

The following basin images, in their varying levels of effectiveness, are shown below. Similarly to Function 1, the order the functions are run matters. Specifically, when a lower-performing method such as SD or Newton runs first, it decreases the effectiveness of the other methods. The basins shown below are run independently, with the exception of Backtracking with the Armijo-Goldstein condition and GDM, which are high-performing methods for this function. The consistent coloring scheme helps show the viewer that the methods are not only effective in finding the location of the minima, but also identify the same ones. Note the varying effectiveness of the basins:

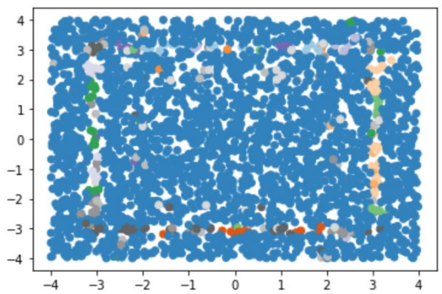


FIGURE 49. SD Basin for Function 4

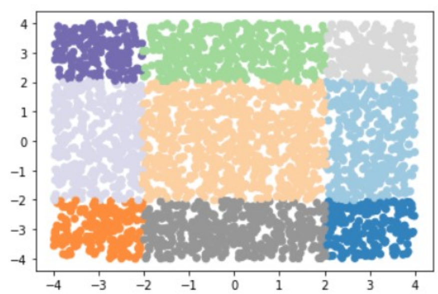


FIGURE 50. Backtracking with AG Condition for Function 4

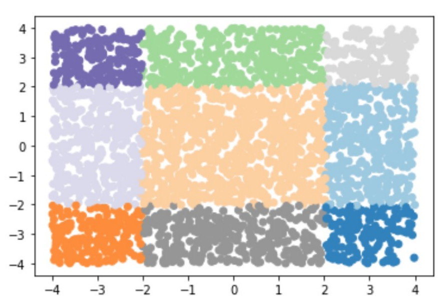


FIGURE 51. GDM Basin for Function 4

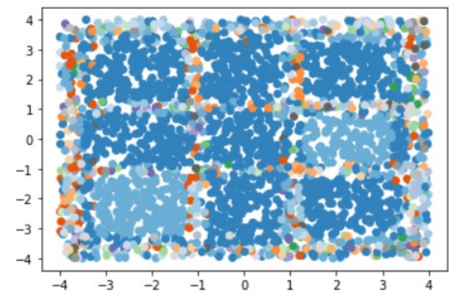


FIGURE 52. Newton's Method Basin for Function 4

The numbered contour plot above shows nine distinct minina in the range x ∈ [-6,6] and y ∈ [-6,6]. Though the basin images are plotted in the range x ∈ [-4,4] and y ∈ [-4,4], the slightly zoomed-out contour plot show the peripheral minima that Backtracking with Armijo-Goldstein and GDM identify with high effectiveness.

Though Newton's Method does not effectively locate the distinct minima, it is more effective than SD in that the basin is divided into nine regions, which means that the method identifies unique behavior in these regions, but does not identify these points as minima. Recall in Function 1 when the Newton Method failed when the function changed concavity:

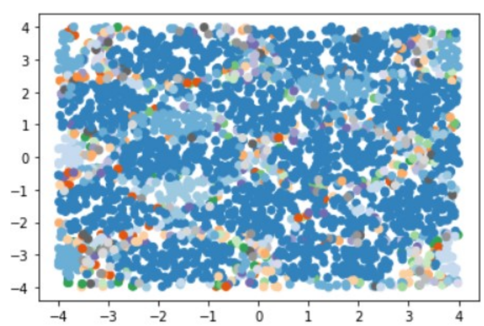


FIGURE 53. Newton's Method Function 1

The rings between the minima and the maxima are areas where Newton's Method fails, as the second derivative is zero in locations where the function changes concavity. In a similar way, Newton's Method fails around the nine regions in Function 4 where the function changes concavity. The method is not effective at locating the distinct minima, but the locations the method fails for both functions mirror the contour plot in interesting ways.

As mentioned in earlier sections, SD oftentimes fails to produce a clear basin when the function is steep in certain locations, because the negative gradient gets larger and produces less accurate guesses in these scenarios. The same is true for this function. In addition, the Backtracking Method without the Armijo-Goldstein Condition fails completely for this method, largely due to the large amount of minima in a limited range. This will be discussed in more detail in the sensitivity analysis portion of the paper, but, in general, Backtracking without the Armijo-Goldstein Condition is sensitive to large amounts of minima in confined spaces.

In Summary:
1) Backtracking with the Armijo-Goldstein Condition and the GDM Method perform similarly and with high effectiveness for this function.
2) SD largely fails here, mostly because of the steepness of the function in this region.
3) Though Newton's method also largely fails for this function, the locations where the second derivative is zero reveal the locations of the minima, even though the method does not recognize them as such.
4) Backtracking without the Armijo-Goldstein Condition fails due to the prescence of many minima in a confined region.

16. Sensitivity Analysis

The next part of this project will discuss the effectiveness of the methods in a new way, one that analyzes the consistency of the basins when a variety of changes are made to the function. Preserving the consistency of the methods with such function changes is important for different reasons, most notably due to the modeling process. When these methods are used in an applied modeling setting, building a function to represent real-life scenarios requires trial and error. It is advantageous for methods to not be sensitive to changes so that the modeler can produce meaningful results without unreasonable precision. In addition, methods with low sensitivity decrease the time costs of the estimation process. The following sections will discuss the sensitivity of each method with the functions explored in this project. Each section will detail the changes made to the function, the basin images to illustrate the changes, and a discussion about the methods' sensitivity.

17. Sensitivity of Function 1, $\sin(x)\cos(y) + \cos^2(y)$

The following tables illustrate how individual alterations to the function impact the performance of the Steepest Descent Method with the Golden-Section Search. Note that changing the coefficients or increasing the exponent on the last term to another even value does not impact the location of the minima, but rather the steepness of the function in the areas surrounding the minima. When the last term is changed to an odd number, however, it reduces the amount of minima as well as the steepness of the function in the areas surrounding the minima. Included are examples to illustrate how these changes do not impact the location of the minima, but rather the amount of minima or the amplitude of the function directly surrounding them. Lastly, if an altered function drastically changes the location of the minima, then it is not included in the sensitivity analysis. Note that if the number of minima is changed without altering the rest of the contour, then it is not considered a drastic change.

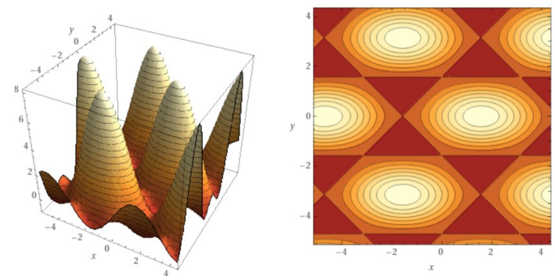


FIGURE 54. **Coefficient Changes**. Left: $8\sin(x)\cos(y) + 8\cos^2(y)$ Plot; Right: $8\sin(x)\cos(y) + 8\cos^2(y)$ Contour Plot

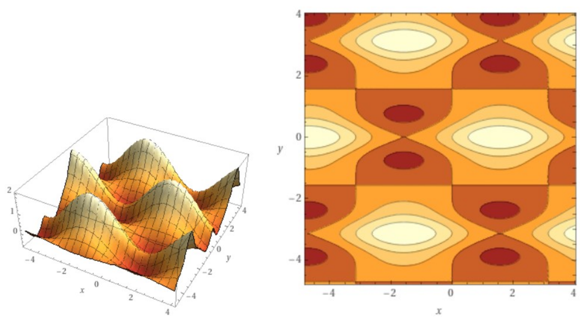


FIGURE 55. **Exponent Changes**. Left: $\sin(x)\cos(y) + \cos(y)^8$ Plot; Right: $\sin(x)\cos(y) + \cos(y)^8$ Contour Plot

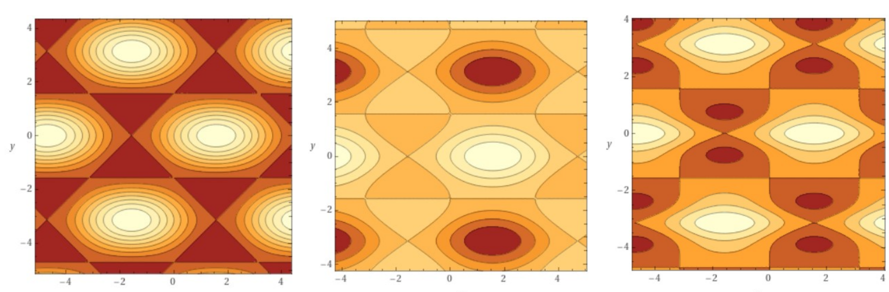
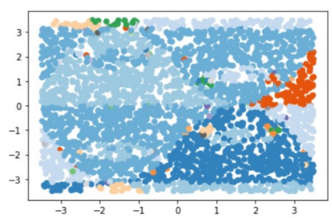
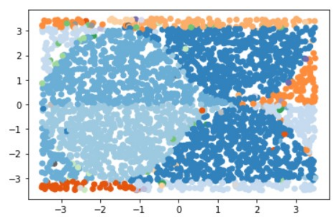
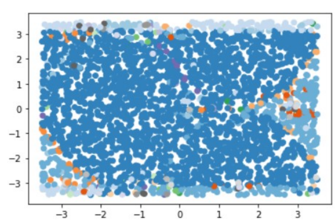
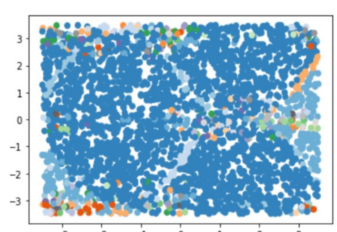
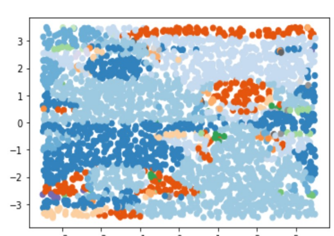
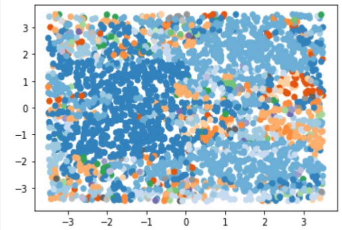
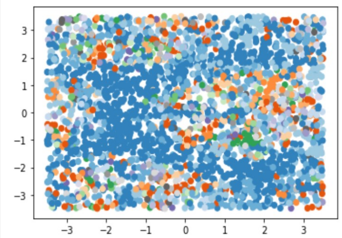
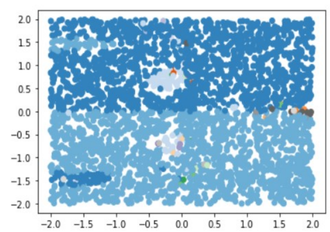
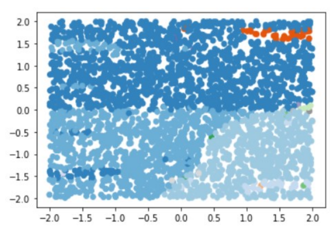
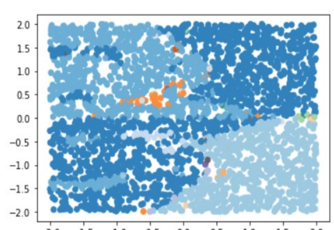
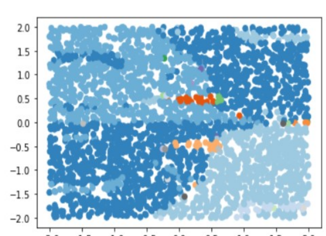


FIGURE 56. Left: Original Contour Plot; Center: Contour with Odd Exponent; Right: Contour Plot with Even Exponent

Function 1 Sensitivity Analysis: Steepest Descent with Golden-Section Search

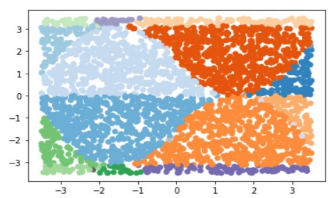
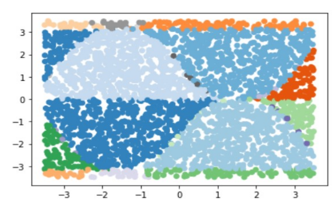
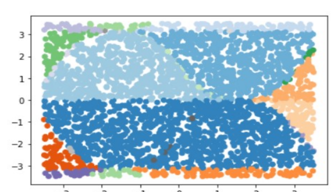
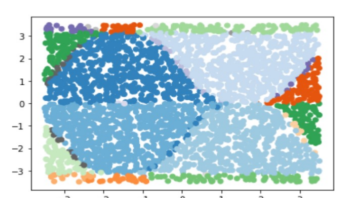
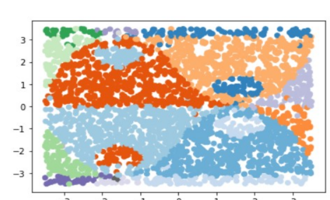
For the sensitivity analysis portion of this project, each function and its respective changes are run in different kernels so that they are not affected by the performance of any other. Based on previous discussion, this means each basin will have unique coloring schemes. In general, and especially for this method, the matching of specific colors is not as important as the basins showing similarly-defined regions with non-repeated colors.

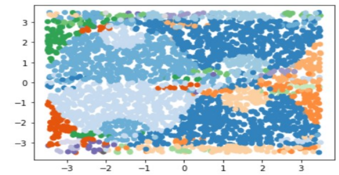
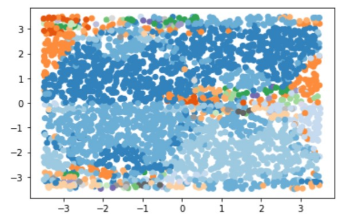
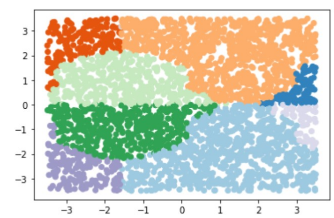
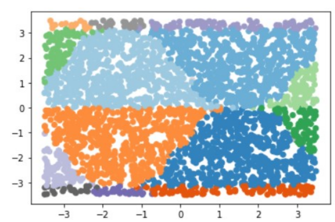
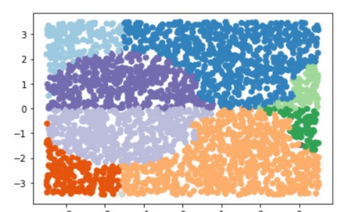
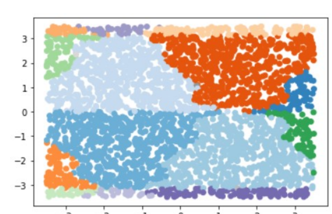
Function	Basin
Original Function	
$\frac{1}{2}\sin(x)\cos(y) + \frac{1}{2}\cos^2(y)$	
$\frac{1}{4}\sin(x)\cos(y) + \frac{1}{4}\cos^2(y)$	
$\frac{1}{8}\sin(x)\cos(y) + \frac{1}{8}\cos^2(y)$	
$2\sin(x)\cos(y) + 2\cos^2(y)$	

Function	Basin
$4\sin(x)\cos(y) + 4\cos^2(y)$	
$8\sin(x)\cos(y) + 8\cos^2(y)$	
$\sin(x)\cos(y) + \cos^3(y)$	
$\sin(x)\cos(y) + \cos^4(y)$	
$\sin(x)\cos(y) + \cos^5(y)$	
$\sin(x)\cos(y) + \cos^6(y)$	

These images show how the effectiveness of the SD Method is fairly sensitive to increases or decreases in the coefficients, as well as increases in the exponent in the final term of the function. Though the SD Method is not an effective method in and of itself, it is fairly sensitive to even small changes to the function, which further decreases the effectiveness of the method.

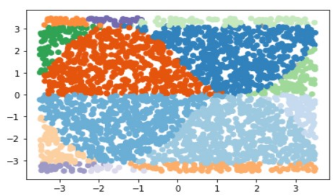
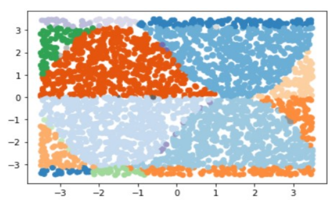
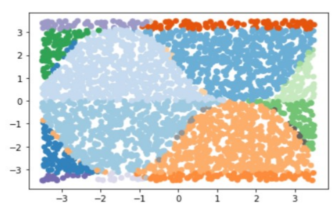
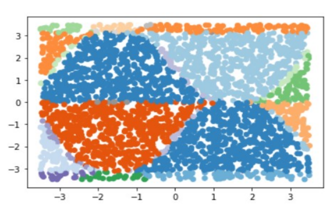
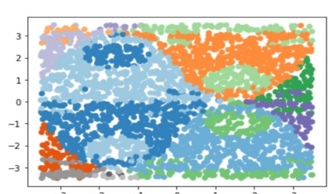
Function 1 Sensitivity Analysis: Backtracking with the Armijo-Goldstein Condition

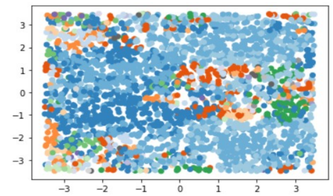
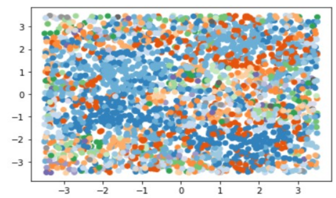
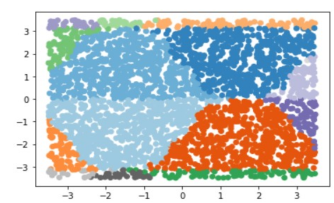
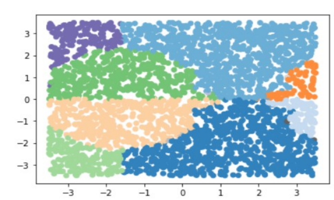
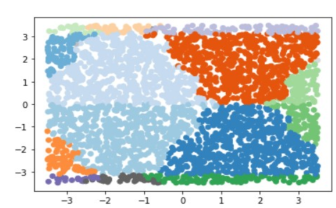
Function	Basin
Original Function	
$\frac{1}{2}\sin(x)\cos(y) + \frac{1}{2}\cos^2(y)$	
$\frac{1}{4}\sin(x)\cos(y) + \frac{1}{4}\cos^2(y)$	
$\frac{1}{8}\sin(x)\cos(y) + \frac{1}{8}\cos^2(y)$	
$2\sin(x)\cos(y) + 2\cos^2(y)$	

Function	Basin
$4\sin(x)\cos(y) + 4\cos^2(y)$	
$8\sin(x)\cos(y) + 8\cos^2(y)$	
$\sin(x)\cos(y) + \cos^3(y)$	
$\sin(x)\cos(y) + \cos^4(y)$	
$\sin(x)\cos(y) + \cos^5(y)$	
$\sin(x)\cos(y) + \cos^6(y)$	

As compared to the SD Method, Backtracking with the Armijo-Goldstein condition is not nearly as sensitive. Though its effectiveness decreases slightly with the changes to the coefficients above, the method remains remarkably consistent in finding the locations of minima as compared to the SD Method. The method stays the most consistent with small changes in both odd and even exponents. This method is ultimately a more consistent and less sensitive method than SD with the Golden-Section Search for Function 1.

Function 1 Sensitivity Analysis: Backtracking without the Armijo-Goldstein Condition

Function	Basin
Original Function	
$\frac{1}{2}\sin(x)\cos(y)+\frac{1}{2}\cos^2(y)$	
$\frac{1}{4}\sin(x)\cos(y)+\frac{1}{4}\cos^2(y)$	
$\frac{1}{8}\sin(x)\cos(y)+\frac{1}{8}\cos^2(y)$	
$2\sin(x)\cos(y)+2\cos^2(y)$	

Function	Basin
$4\sin(x)\cos(y) + 4\cos^2(y)$	
$8\sin(x)\cos(y) + 8\cos^2(y)$	
$\sin(x)\cos(y) + \cos^3(y)$	The method failed to converge
$\sin(x)\cos(y) + \cos^4(y)$	
$\sin(x)\cos(y) + \cos^5(y)$	
$\sin(x)\cos(y) + \cos^6(y)$	

As compared to Backtracking with the Armijo-Goldstein condition, this method performs marginally less effectively. The method is not noticeably sensitive to decreasing the coefficients until the coefficients are decreased to $\frac{1}{8}$. However, when the coefficients are increased, the basin images become cloudier than those of the Backtracking with the Armijo-Goldstein condition. The method also fails to converge with the function $\sin(x)\cos(y) + \cos^3(y)$, but not $\sin(x)\cos(y) + \cos^2(y)$ or $\sin(x)\cos(y) + \cos^4(y)$, which is a consequence of the method's sensitivity to changes in steepness. For the function $\sin(x)\cos(y) + \cos^3(y)$, the area between the minima and maxima is more plateaued than for the other methods, which suggests that the method may stall if a random point was too close to the plateaus.

Backtracking without the AG Condition is ultimately a slightly less effective method than Backtracking with the AG condition, though both perform drastically better than the SD Method.

Function 1 Sensitivity Analysis: Gradient Descent with Momentum Method

The following basin images not only involve small changes to the function, but also tweaks in the learning rate (LR) and momentum value (MV) parameters discussed in the introduction to the GDM Method. For each function, the LR and MV parameters were both chosen from three possible values: 0.01, 0.34, and .99. The first five images are of the original function with learning rate-momentum value (LR-MV) combinations of (.34-.34), (.34-.99), (.34-.01), (0.01-.34) and (.99-.34).

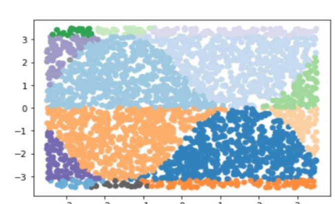
Note the following about the basin images shown below:

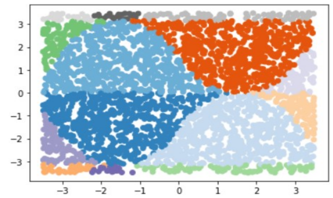
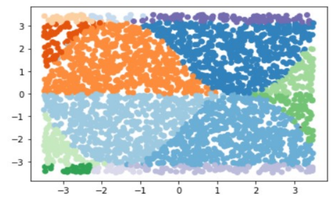
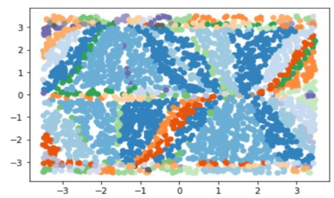
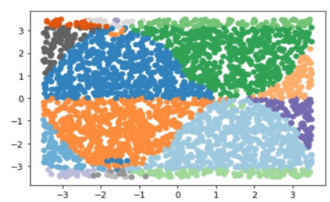
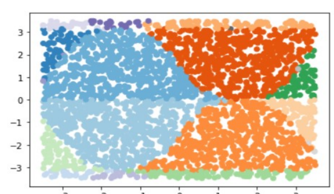
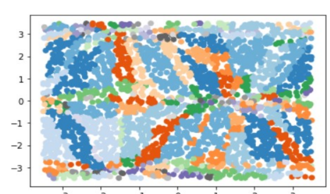
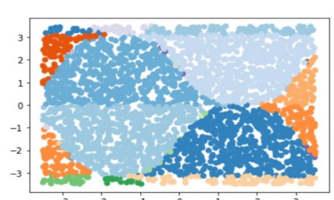
1) The basins from the tweaked functions shown below have default LR-MV parameters of (.34-.34) unless otherwise noted.

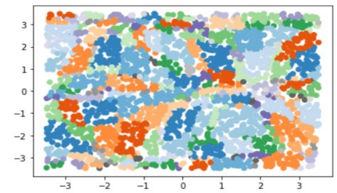
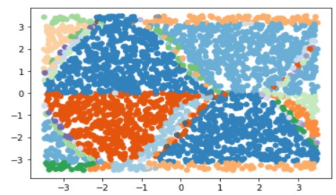
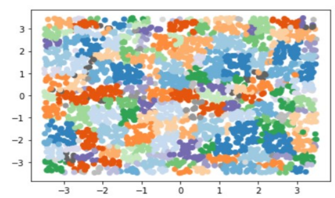
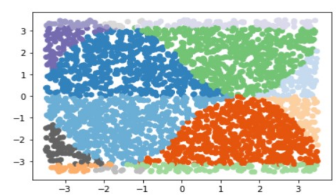
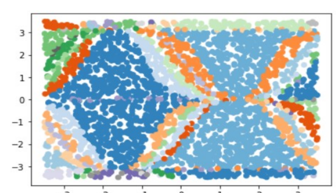
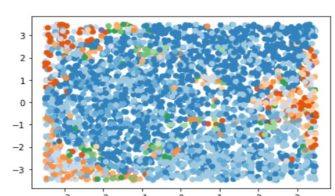
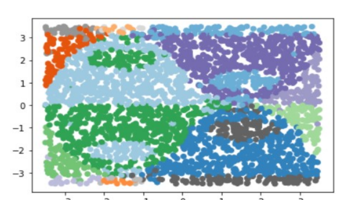
2) Any other basin from a non-original function shown below is pictured because its basin deviates from that of the original function with the exact same parameters, or because its pattern is drastically unexpected from functions with the same alterations but different LR-MV values. For example, if an altered function with a LR of .10 and an MV of .34 produces a basin that deviates drastically from the behavior of the basin of the original function with an LR of .10 and an MV of .34, the basin of the tweaked function will be shown below.

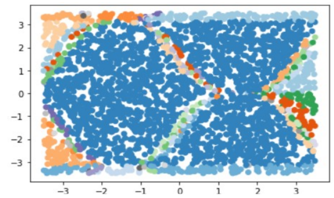
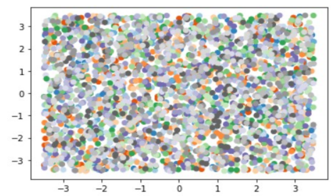
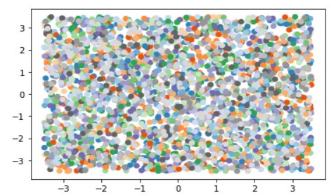
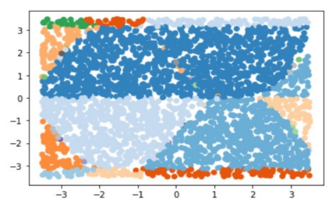
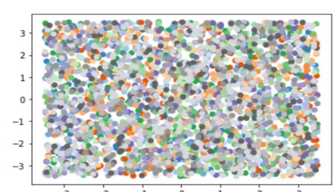
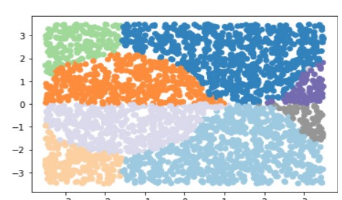
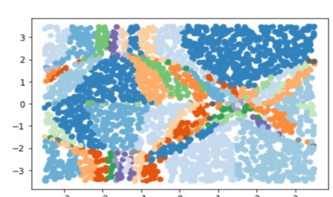
3) If just the LR modification is made, assume that differing MV values did not drastically change the image.

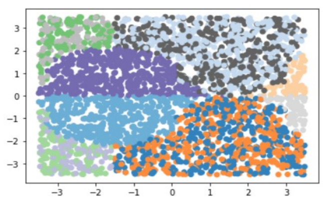
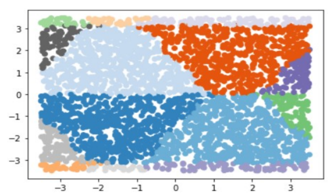
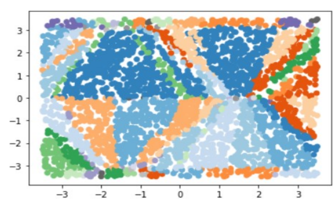
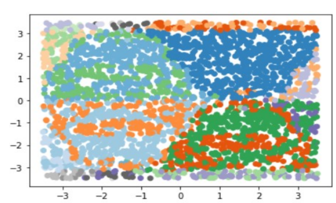
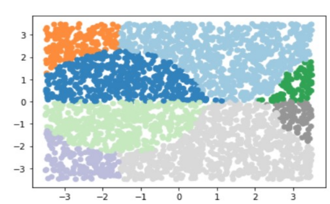
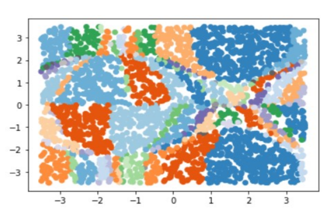
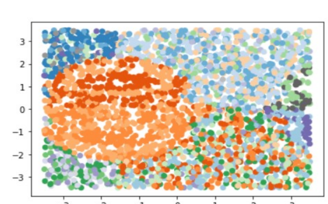
Reminder: If a basin with a particular LR-MV combination is excluded, it is because the basin image resembles the behavior of the original function with the same parameters, or the behavior of other images with the same function modifications, but different LR-MV combinations. Ultimately, this section intends to illustrate the patterns and unexpected tendencies of this process when LR-MV combinations are introduced.

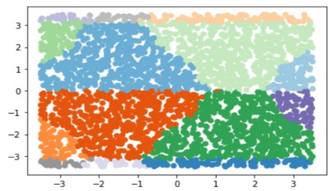
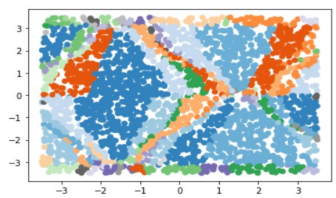
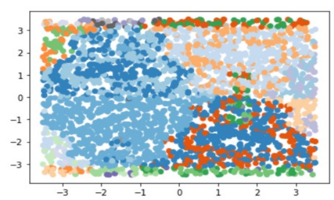
Function	Basin
Original Function, LR = .34, MV = .34	

Function	Basin
Original Function, LR = .34, MV = .99	
Original Function, LR = .34, MV = .01	
Original Function, LR = .01, MV = .34	
Original Function, LR = .99, MV = .34	
$\frac{1}{2}\sin(x)\cos(y) + \frac{1}{2}\cos^2(y)$	
$\frac{1}{2}\sin(x)\cos(y) + \frac{1}{2}\cos^2(y)$, LR = 0.01	
$\frac{1}{4}\sin(x)\cos(y) + \frac{1}{4}\cos^2(y)$	

Function	Basin
$\frac{1}{4}\sin(x)\cos(y) + \frac{1}{4}\cos^2(y)$, LR = 0.01	
$\frac{1}{8}\sin(x)\cos(y) + \frac{1}{8}\cos^2(y)$	
$\frac{1}{8}\sin(x)\cos(y) + \frac{1}{8}\cos^2(y)$, LR = 0.01	
$2\sin(x)\cos(y) + 2\cos^2(y)$	
$2\sin(x)\cos(y) + 2\cos^2(y)$, LR = 0.01	
$2\sin(x)\cos(y) + 2\cos^2(y)$, LR = 0.99	
$4\sin(x)\cos(y) + 4\cos^2(y)$	

Function	Basin
$4\sin(x)\cos(y) + 4\cos^2(y)$, LR $= 0.01$	
$4\sin(x)\cos(y) + 4\cos^2(y)$, LR $= 0.99$	
$8\sin(x)\cos(y) + 8\cos^2(y)$	
$8\sin(x)\cos(y) + 8\cos^2(y)$, LR $= 0.01$	
$8\sin(x)\cos(y) + 8\cos^2(y)$, LR $= 0.99$	
$\sin(x)\cos(y) + \cos^3(y)$	
$\sin(x)\cos(y)+\cos^3(y)$, LR $=$ 0.01	

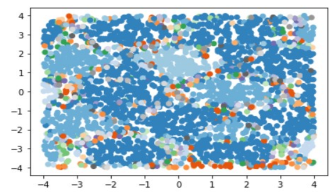
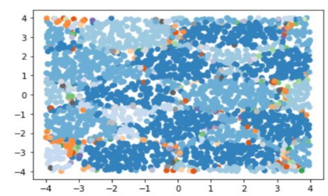
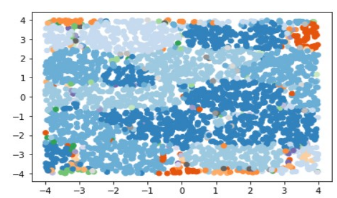
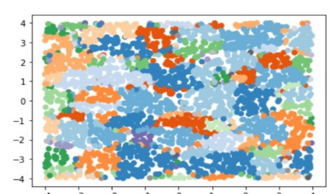
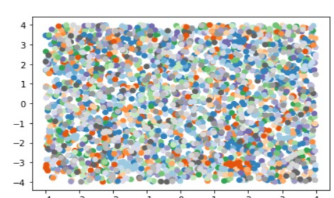
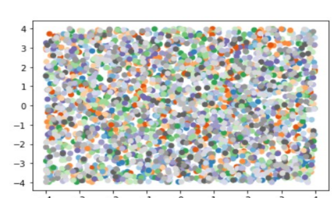
Function	Basin
$\sin(x)\cos(y)+\cos^3(y)$, LR = 0.99	
$\sin(x)\cos(y)+\cos^4(y)$	
$\sin(x)\cos(y)+\cos^4(y)$, LR = 0.01	
$\sin(x)\cos(y)+\cos^4(y)$, LR = .99	
$\sin(x)\cos(y)+\cos^5(y)$	
$\sin(x)\cos(y)+\cos^5(y)$, LR = 0.01	
$\sin(x)\cos(y)+\cos^5(y)$, LR = 0.99	

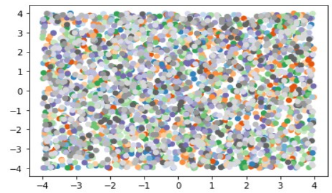
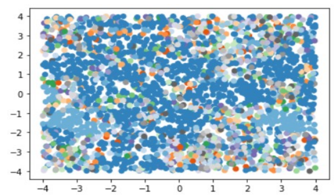
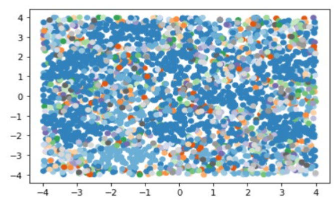
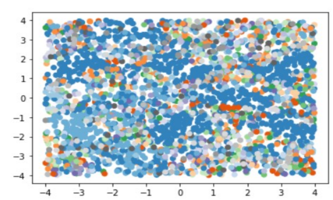
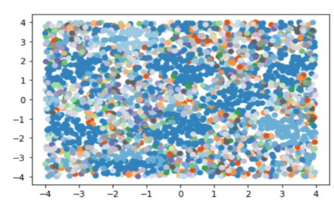
Function	Basin
$\sin(x)\cos(y) + \cos^6(y)$	
$\sin(x)\cos(y) + \cos^6(y)$, LR $=$ 0.01	
$\sin(x)\cos(y) + \cos^6(y)$, LR $=$ 0.99	

Note the following about the performance of the functions and parameters:
1) As the coefficients are tweaked decreasingly and increasingly away from the original function, the method became less effective for all LR-MV values.
2) Most of the time, functions with high LRs perform effectively
3) The functions with exponent changes all performed similarly; LR values of .34 outperformed LR values of .01 and .99.
4) The MV has insignificant effects on the basins as compared to the LR.

The GDM Method ultimately performs relatively consistently for LR and MV parameters of .34, and similarly in effectiveness to the Bactracking Methods. An advantage of the GDM Method over the Backtracking Methods is that the LR and MV values can be scaled to optimize the performance of the method for any given function. Thus, whereas the Backtracking Methods work the same for all functions, the LR and MV parameters can be tailored to optimize the performance of the method for different functions.

Function 1 Sensitivity Analysis: Newton's Method

Function	Basin
Original Function	
$\frac{1}{2}\sin(x)\cos(y)+\frac{1}{2}\cos^2(y)$	
$\frac{1}{4}\sin(x)\cos(y)+\frac{1}{4}\cos^2(y)$	
$\frac{1}{8}\sin(x)\cos(y)+\frac{1}{8}\cos^2(y)$	
$2\sin(x)\cos(y)+2\cos^2(y)$	
$4\sin(x)\cos(y)+4\cos^2(y)$	

Function	Basin
$8\sin(x)\cos(y) + 8\cos^2(y)$	
$\sin(x)\cos(y) + \cos^3(y)$	
$\sin(x)\cos(y) + \cos^4(y)$	
$\sin(x)\cos(y) + \cos^5(y)$	
$\sin(x)\cos(y) + \cos^6(y)$	

As previously mentioned, Newton's Method largely fails for this function, primarily because the function has many maximizers and minimizers, the second derivative is zero at a high volume of random points, and the function has many oscillations. Though all of the basin images are somewhat cloudy and largely fail to reveal the defined areas where the minima are, the function performed worst with an increase in coefficients. As compared with the rest of the other methods, Newton's Method is ultimately the least effective for Function 1.

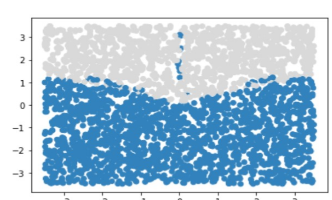
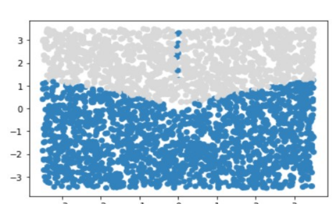
Sensitivity Analysis Ranking: Function 1

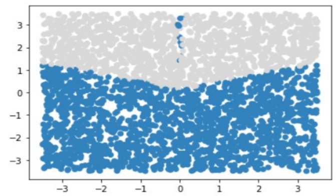
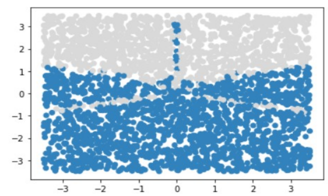
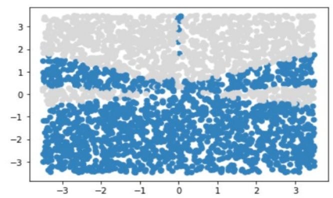
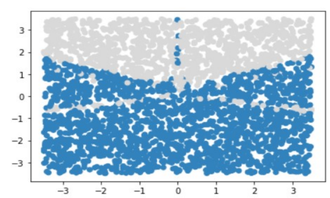
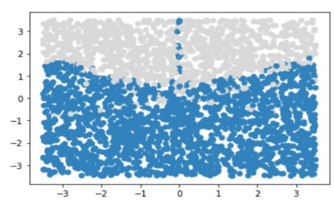
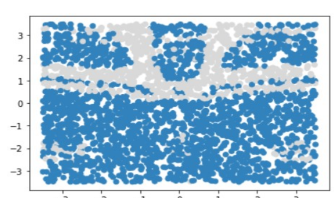
Note the following effectiveness rankings of the methods on Function 1:
1) Gradient Descent with Momentum with learning rate and momentum values at .34
2) Backtracking Method with the Armijo-Goldstein Condition
3) Backtracking Method without the Armijo-Goldstein Condition
4) Steepest Descent with the Golden-Section Search
5) Newton's Method

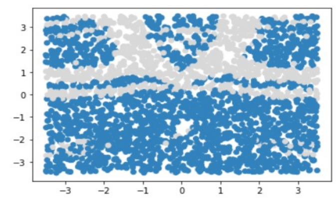
18. Sensitivity Analysis of Function 2, $\frac{1}{4}(1+4(x^2+(-1+y)^2))(x^2+(1+y)^2)$

As with the first function, the changes explored to Function 2 do not change the location of the minima, but rather the behavior of the function surrounding the minima. In general for Function 2, decreasing and increasing the front coefficient beyond $\frac{1}{4}$ alters the steepness of the function around the minima. When the exponent on the first x^2 term is changed to x^4, it slightly widens the region containing the minima. When the final x^2 term is changed to x^4, it decreases the steepness of the areas immediately surrounding the minima. In general, however, none of the changes impact the locations of the minima themselves.

Steepest Descent with Golden-Section Search

Function	Basin
Original function	
$\frac{1}{8}(1+4(x^2+(-1+y)^2))(x^2+(1+y)^2)$	

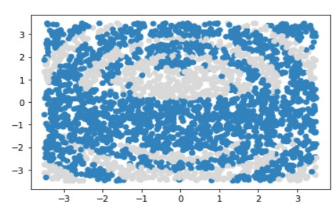
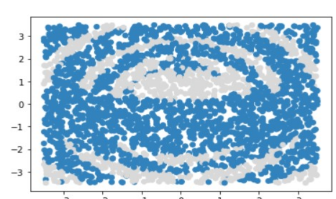
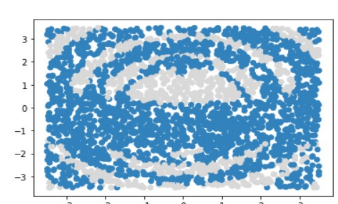
Function	Basin
$\frac{1}{10}(1+4(x^2+(-1+y)^2))(x^2+(1+y)^2)$	
$\frac{1}{2}(1+4(x^2+(-1+y)^2))(x^2+(1+y)^2)$	
$2(1+4(x^2+(-1+y)^2))(x^2+(1+y)^2)$	
$4(1+4(x^2+(-1+y)^2))(x^2+(1+y)^2)$	
$8(1+4(x^2+(-1+y)^2))(x^2+(1+y)^2)$	
$\frac{1}{4}(1+4(x^4+(-1+y)^2))(x^2+(1+y)^2)$	

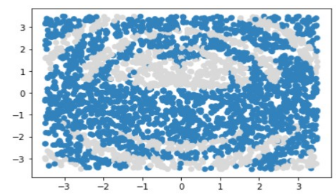
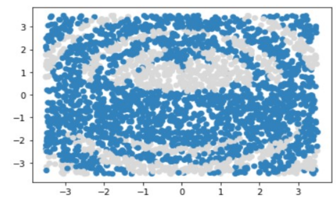
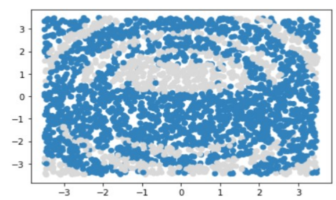
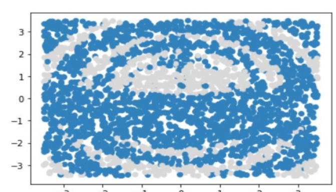
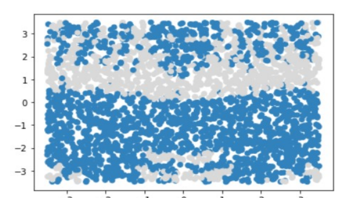
Function	Basin
$\frac{1}{4}(1+4(x^2+(-1+y)^2))(x^4+(1+y)^2)$	

Note the consistency of the basins when the coefficients in the front of the function are altered despite the emergence of the bands of gray when the coefficients are increased. Also note the consistency of the strip of blue points in the top region of the image in all the functions where the coefficients are altered.

When the exponents on the inside terms of the function are increased, the areas the minima inhabit become marginally smaller. As a result, less points approach the gray minima and instead approach the global minimizer. Though the majority of points approach the global minimum and succeed at bypassing the local one, a large portion of the basin is still comprised of points that fail to approach the true minimum.

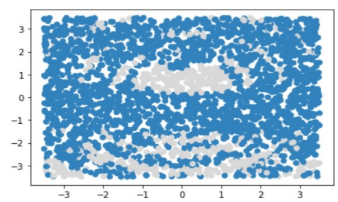
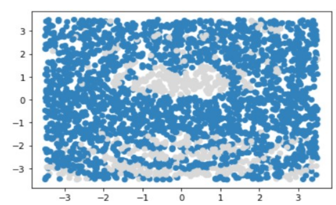
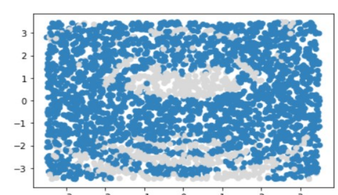
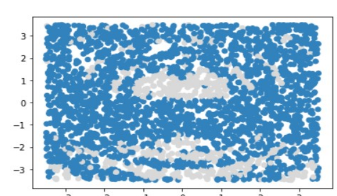
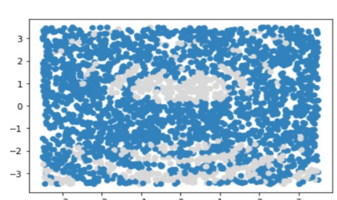
Backtracking with the Armijo-Goldstein Condition

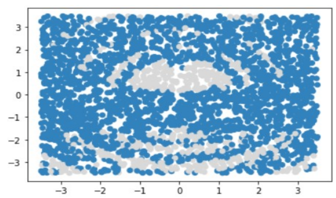
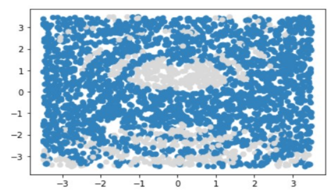
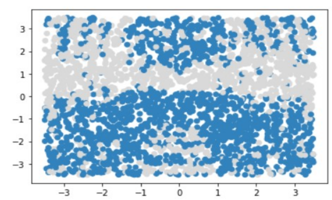
Function	Basin
Original Function	
$\frac{1}{8}(1+4(x^2+(-1+y)^2))(x^2+(1+y)^2)$	
$\frac{1}{10}(1+4(x^2+(-1+y)^2))(x^2+(1+y)^2)$	

Function	Basin
$\frac{1}{2}(1+4(x^2+(-1+y)^2))(x^2+(1+y)^2)$	
$2(1+4(x^2+(-1+y)^2))(x^2+(1+y)^2)$	
$4(1+4(x^2+(-1+y)^2))(x^2+(1+y)^2)$	
$8(1+4(x^2+(-1+y)^2))(x^2+(1+y)^2)$	
$\frac{1}{4}(1+4(x^4+(-1+y)^2))(x^2+(1+y)^2)$	
$\frac{1}{4}(1+4(x^2+(-1+y)^2))(x^4+(1+y)^2)$	The method fails to converge

As compared with the SD Method, these basins have less-defined regions separating the local and global minima. In general, the more blue points appear on the basin, the better the method performs at bypassing the local minimum to locate the global one. In the top portion of the basin, where the local minimum is located, this method is more able to bypass the local minimum than the SD Method, as shown by the greater number of blue points. In the bottom portion of the basin, close to the location of the global minimum, the method has more mixed results than with the SD Method for many of the basins. In general, the Backtracking Method with the Armijo-Goldstein Condition performed with similar effectiveness to the SD Method, though the basin images have different characteristics.

Backtracking without the Armijo-Goldstein Condition

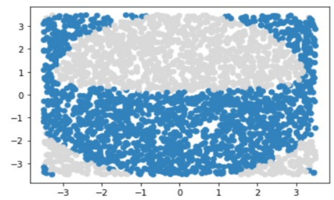
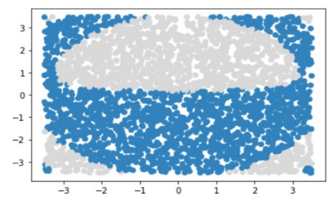
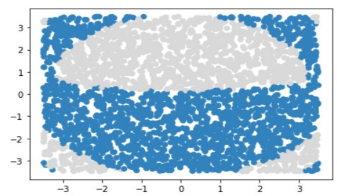
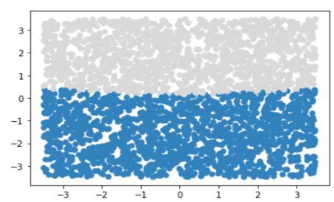
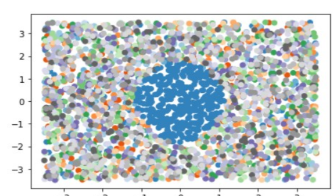
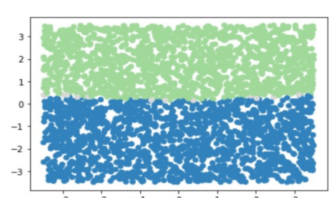
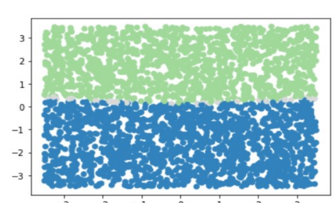
Function	Basin
Original Function	
$\frac{1}{8}(1+4(x^2+(-1+y)^2))(x^2+(1+y)^2)$	
$\frac{1}{10}(1+4(x^2+(-1+y)^2))(x^2+(1+y)^2)$	
$\frac{1}{2}(1+4(x^2+(-1+y)^2))(x^2+(1+y)^2)$	
$2(1+4(x^2+(-1+y)^2))(x^2+(1+y)^2)$	

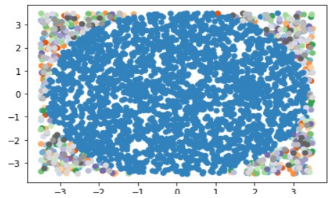
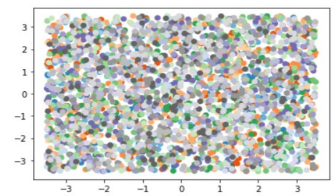
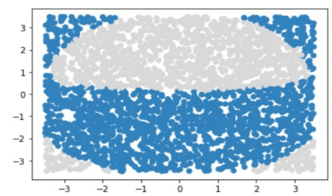
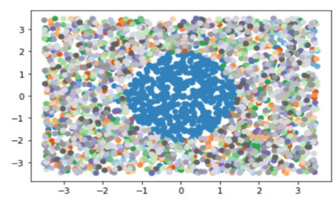
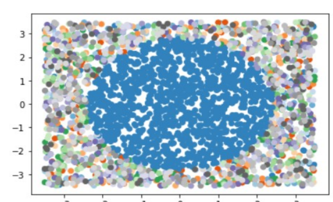
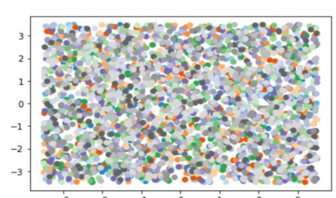
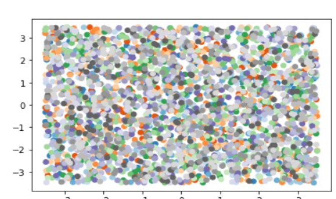
Function	Basin
$4(1+4(x^2+(-1+y)^2))(x^2+(1+y)^2)$	
$8(1+4(x^2+(-1+y)^2))(x^2+(1+y)^2)$	
$\frac{1}{4}(1+4(x^4+(-1+y)^2))(x^2+(1+y)^2)$	The method fails to converge
$\frac{1}{4}(1+4(x^2+(-1+y)^2))(x^4+(1+y)^2)$	

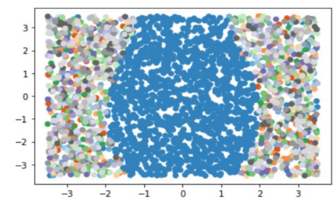
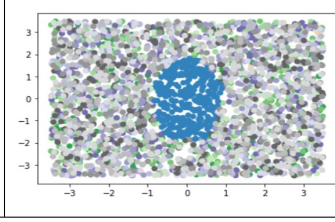
The performance of this method is very similar to the Backtracking Method with the Armijo-Goldstein Method, especially in the appearance of the gray rings surrounding the minima. However, because these basins have a much lower proportion of gray as in the previous method, this method is ultimately more effective. As in the case of the Backtracking without the Armijo-Goldstein Method, this method greatly outperforms the SD Method.

Sensitivity Analysis for Function 2: Gradient Descent with Momentum

For this function, the LR that produces the most optimal basin images is 0.02. Learning rates and momentum values of 0.009, 0.02, and .2 were also used to explore their effects on this method. The basins following those of the original function will depict functions with the typical alterations with a default LR of 0.02, and any other LR-MV combination that yields a result different than that of the original function with the same LR-MV, or a drastically different basin with similar LR-MV values. When a basin contains a color other than gray or blue, the method locates additional minima elsewhere other than the two true minima, since the colormap's default colors for basins of two minima are blue and gray.

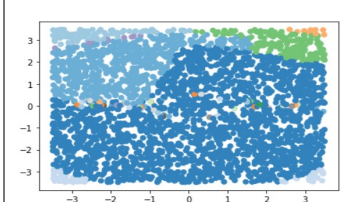
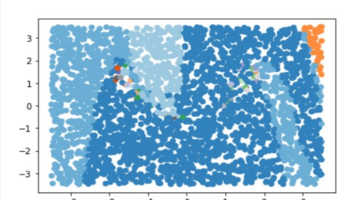
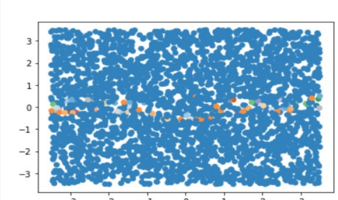
Function	Basin
Original Function, LR = 0.02, MV = 0.02	
Original Function, LR = 0.02, MV = 0.2	
Original Function, LR = 0.02, MV = 0.009	
Original Function, LR = 0.009, MV = 0.02	
Original Function, LR = 0.2, MV = 0.02	
$\frac{1}{8}(1+4(x^2+(-1+y)^2))(x^2+(1+y)^2)$	
$\frac{1}{10}(1+4(x^2+(-1+y)^2))(x^2+(1+y)^2)$	

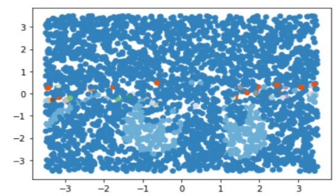
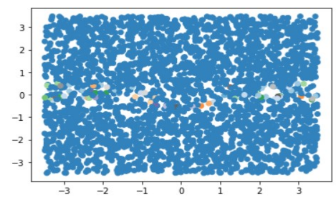
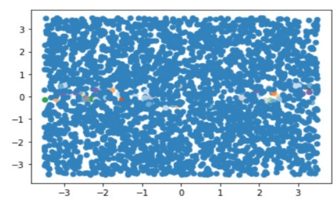
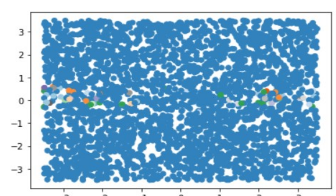
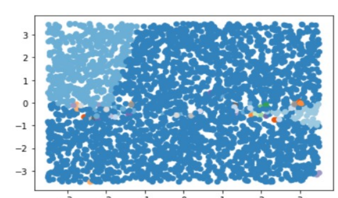
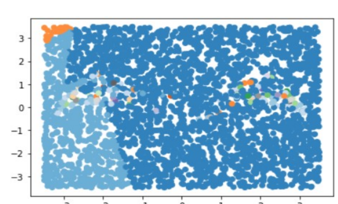
Function	Basin
$\frac{1}{2}(1+4(x^2+(-1+y)^2))(x^2+(1+y)^2)$	
$\frac{1}{2}(1+4(x^2+(-1+y)^2))(x^2+(1+y)^2)$, LR = 0.2, MV = 0.2	
$\frac{1}{2}(1+4(x^2+(-1+y)^2))(x^2+(1+y)^2)$, LR = 0.009, MV = 0.02	
$2(1+4(x^2+(-1+y)^2))(x^2+(1+y)^2)$	
$2(1+4(x^2+(-1+y)^2))(x^2+(1+y)^2)$, LR = 0.009, MV = 0.02	
$4(1+4(x^2+(-1+y)^2))(x^2+(1+y)^2)$	
$8(1+4(x^2+(-1+y)^2))(x^2+(1+y)^2)$	

Function	Basin
$\frac{1}{4}(1+4(x^4+(-1+y)^2))(x^2+(1+y)^2)$	
$\frac{1}{4}(1+4(x^2+(-1+y)^2))(x^4+(1+y)^2)$	

As compared with the previous methods, the GDM Method is highly sensitive to small function changes. The basins of the original function modifications are comparable to, but not as accurate as, the original basins of some of the other methods seen thus far. The method quickly loses effectiveness when the coefficients increase and decrease, and the exponents change value as they do in the final two functions. So far, the GDM Method is the least effective method for Function 2.

Newton's Method

Function	Basin
Original Function	
$\frac{1}{8}(1+4(x^2+(-1+y)^2))(x^2+(1+y)^2)$	
$\frac{1}{10}(1+4(x^2+(-1+y)^2))(x^2+(1+y)^2)$	

Function	Basin
$\frac{1}{2}(1+4(x^2+(-1+y)^2))(x^2+(1+y)^2)$	
$2(1+4(x^2+(-1+y)^2))(x^2+(1+y)^2)$	
$4(1+4(x^2+(-1+y)^2))(x^2+(1+y)^2)$	
$8(1+4(x^2+(-1+y)^2))(x^2+(1+y)^2)$	
$\frac{1}{4}(1+4(x^4+(-1+y)^2))(x^2+(1+y)^2)$	
$\frac{1}{4}(1+4(x^4+(-1+y)^2))(x^4+(1+y)^2)$	

This method appears to be the least effective for this function, as it fails to find the minima in any clearly-defined way. The most likely reason for this is the proximity of the minima to one another, which creates close oscillations that cause difficulties for the minima to properly distinguish between the minima.

Sensitivity Analysis Ranking: Function 2

Note the following effectiveness rankings of the methods on Function 2:
1) Backtracking without the Armijo-Goldstein Condition
2) Backtracking with the Armijo-Goldstein Condition
3) Steepest Descent with the Golden-Section Search
4) Gradient Descent with Momentum
5) Newton's Method

The Backtracking Methods continue to perform efficiently, and that the Newton's Method and the Steepest Descent with the Golden-Section Search continue to perform poorly. As compared to the previous function, the Gradient Descent with Momentum Method has the greatest (negative) change in performance.

Various functions require different LR-MV combinations to optimize the performance of the method. In the case of Function 1, an LR of .34 optimizes GDM's performance. For function 2, an LR of .02 optimizes the performance of the method. Because of the proximity of the oscillations in Function 2, a smaller learning rate is required to prevent overshooting the minima. In addition, the method is far more sensitive to small changes in the LR because of the close proximity of the minima. Thus, it appears that the GDM Method works best with functions whose minima are more spread.

19. Sensitivity Analysis of Function 3, $(x-4-y)^2$

For the following methods, a tolerance value was chosen so that the basins show many colors, with none repeated. Note again that the each function and its modifications are run in different kernels, so the ordering of colors is different in each basin. The most important point of comparison for this function is the matching of the size of each color-region, and the clarity of the basin image.

When the middle term of the function, -4, is increased and decreased, it slightly shifts the location of the the minima along the x and y axes. Below is a contour plot of $(x-4-y)^2$, $(x-8-y)^2$ and $(x+5+y)^2$ to illustrate these changes.

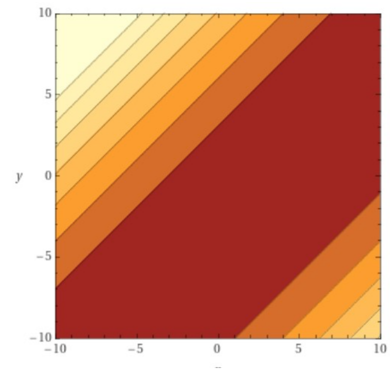


FIGURE 57. Contour Plot $(x - 4 - y)^2$

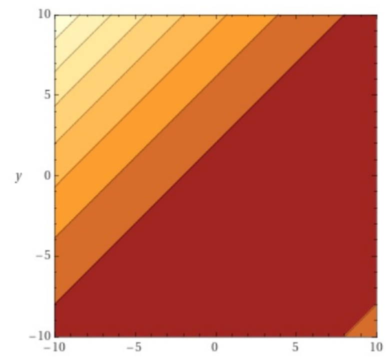


FIGURE 58. Contour Plot $(x - 8 - y)^2$

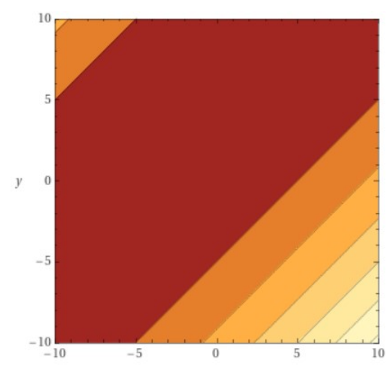
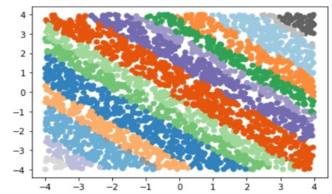
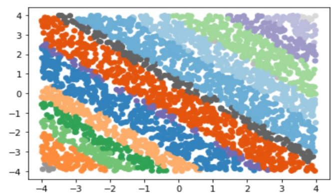
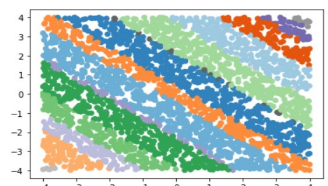
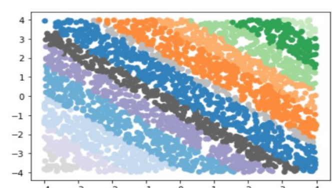
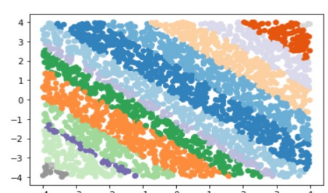
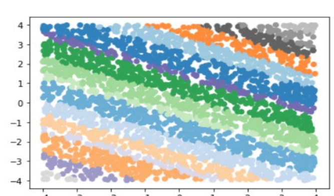
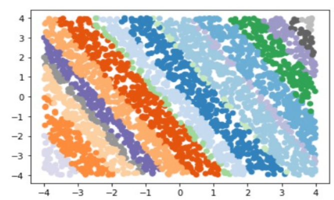


FIGURE 59. Contour Plot $(x + 5 - y)^2$

Additionally, when the final exponent is increased to another even number, the steepness of the contour drastically increases around the minima. Finally, when the coefficient in front of the x or y in the function is increased or decreased, the angle of the strip of minima to the axes changes. Note how the following function changes impact the performance of the method.

Steepest Descent with Golden-Section Search

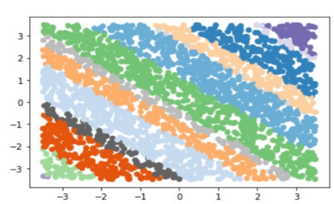
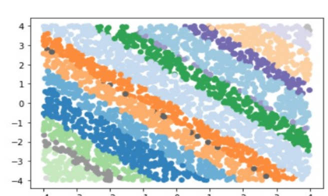
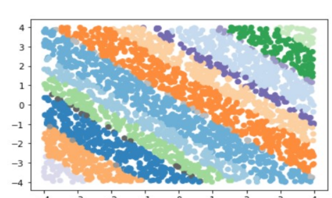
Function	Basin
Original Function	
$(x-6-y)^2$	
$(x-8-y)^2$	
$(x+2-y)^2$	
$(x+4-y)^2$	
$(x-4-y)^4$	The method fails to converge
$(x-4-y)^6$	The method fails to converge
$(2x-4-y)^2$	

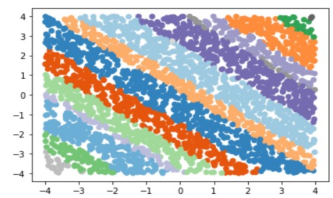
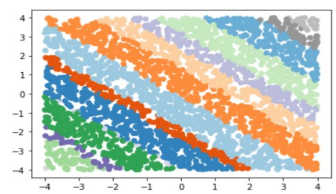
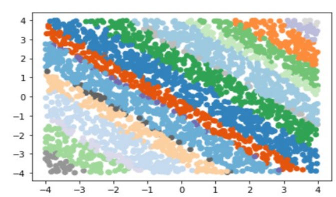
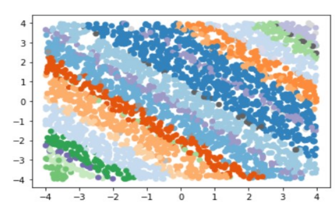
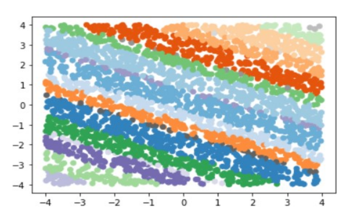
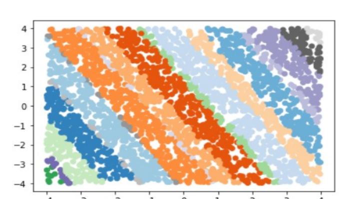
Function	Basin
$(x - 4 - 2y)^2$	

As previously stated, the thickness of each color depends on a set tolerance value. When the tolerance value is small, the width of each color decreases, and vice versa. Again, a moderate tolerance value is used to show many minima, but not enough to repeat colors.

This method is very consistent, and only fails when the steepness around the minima is increased drastically.

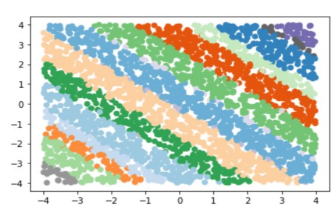
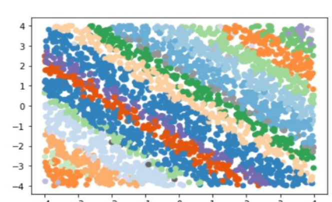
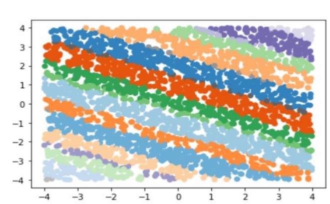
Backtracking with the Armijo-Goldstein Condition

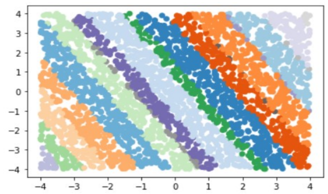
Function	Basin
Original Function	
$(x - 6 - y)^2$	
$(x - 8 - y)^2$	

Function	Basin
$(x+2-y)^2$	
$(x+4-y)^2$	
$(x-4-y)^4$	
$(x-4-y)^6$	
$(2x-4-y)^2$	
$(x-4-2y)^2$	

Most of the basins here have a similar effectiveness as those of the SD Method, where the method performs similarly well with all function changes. Unlike the SD Method, however, this method is still able to produce effective basins even with raised coefficients that increase the steepness around the minima. Whereas the SD Method fails to converge with the functions $(x-4-y)^4$ and $(x-4-y)^6$, the Backtracking Method is able to still produce effective basins. Though the $(x-4-y)^6$ basin repeats colors, a simple change in tolerance values will produce basins of all unique colors. Altogether, the Backtracking Method with the Armijo-Goldstein Condition is a more effective method for Function 2 than the SD Method.

Backtracking Method without the Armijo-Goldstein Condition

Function	Basin
Original Function	The method fails to converge
$(x-6-y)^2$	The method fails to converge
$(x-8-y)^2$	The method fails to converge
$(x+2-y)^2$	The method fails to converge
$(x+4-y)^2$	The method fails to converge
$(x-4-y)^4$	
$(x-4-y)^6$	
$(2x-4-y)^2$	

Function	Basin
$(x-4-2y)^2$	

This sensitivity analysis reveals why Backtracking without the Armijo-Goldstein Condition can fail in certain circumstances, but can begin working when modifications made to the function increase the steepness around the minima. Note the steepness of functions $(x-4-y)^2$, $(x-4-y)^6$, and $(x-4-2y)^2$.

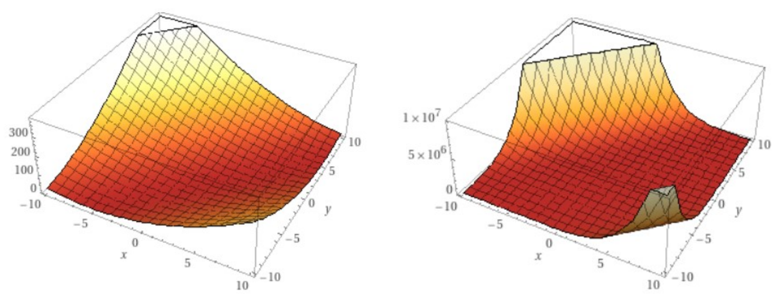


FIGURE 60. Left: $(x-4-y)^2$; Right: $(x-4-y)^6$

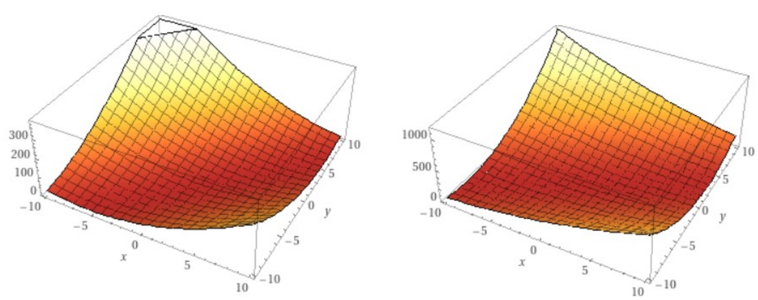


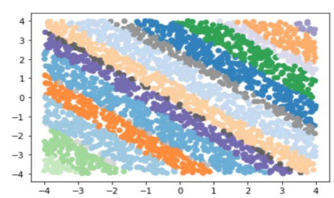
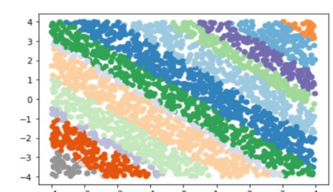
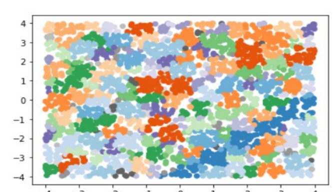
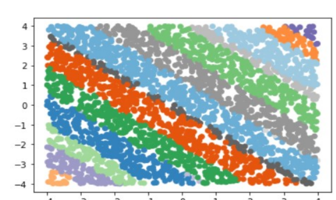
FIGURE 61. Left: $(x-4-y)^2$; Right: $(x-4-2y)^2$

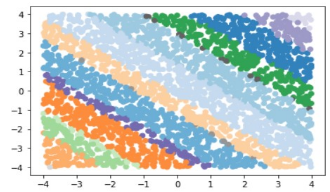
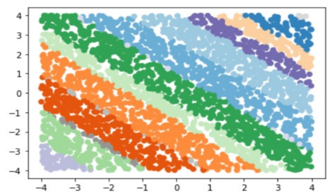
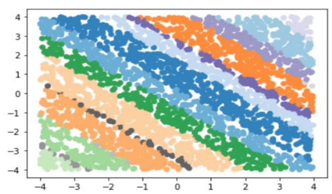
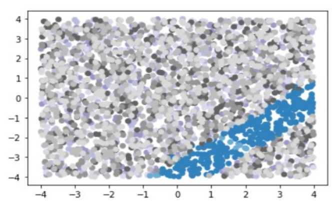
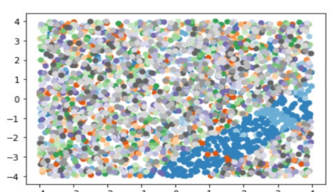
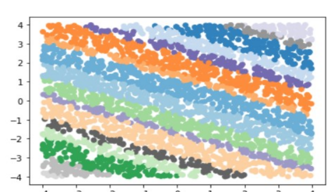
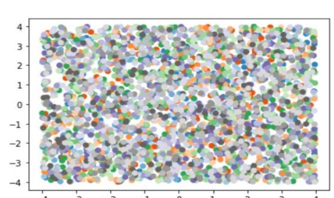
As shown in the plots above, an increase in the exponent or coefficient increases the steepness of the function. Thus, it appears that the steeper the function, the easier it is for the method to find the next guess with a low probability of cycling between guesses or failing to converge. As shown in the conclusion to this section, though the Backtracking with the AG Condition and the SD Method locate minima in similar ways as Backtracking without the AG Condition, certain characteristics make them better able to find minima. This method's limitations with steepness will be explored further in later sections.

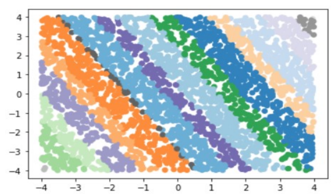
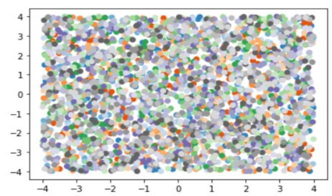
Gradient Descent with Momentum

Like GDM's performance with the other methods, this function has an optimal learning rate value with which the function performs best, and an MV value with largely insignificant effects. For this section, an LR of .2 is the default value. The effects of varying LR-MV pairs are shown for the original function. Basins with varying LR-MV pairs are also shown for other functions any time the change creates an unpredictable basin image.

For the original function, learning rates of .07, .2, and .5 are used.

Function	Basin
$(x-4-y)^2$, LR = .2	
$(x-4-y)^2$, LR = .07	
$(x-4-y)^2$, LR = .5	
$(x-6-y)^2$	

Function	Basin
$(x-8-y)^2$	
$(x+2-y)^2$	
$(x+4-y)^2$	
$(x-4-y)^4$, LR = .2, (& .07 & .5)	
$(x-4-y)^6$, LR = .2, (& .07 & .5)	
$(2x-4-y)^2$	
$(2x-4-y)^2$, LR = .5	

Function	Basin
$(x-4-2y)^2$	
$(x-4-2y)^2$, LR = .5	

The GDM Method is a very effective method when the learning rate is set to 0.02. When this parameter is increased to .5, the method loses much of its effectiveness. In addition, the previous method only produces basins when the function is very steep around the minima. However, this method loses its effectiveness completely in extremely steep cases, as seen with the basins for $(x-4-y)^4$ and $(x-4-y)^6$.

Relatively small changes to the momentum parameter continue to have largely insignificant effects on the basin images. The learning rate, however, seems to largely drive the effectiveness of the method.

Newton's Method

As noted in previous sections about this method, Newton's Method uses the inverse of the Hessian matrix to compute the next guess. If the determinant of the Hessian matrix is zero, it does not have an inverse, and the method therefore cannot progress. The original function and its 8 modifications as shown in the above method sections produce Hessian matrices whose determinants are zero. Thus, these functions stall the method.

This is a general limitation of Newton's Method, and another reason it is less advantageous than the other methods.

Sensitivity Analysis Ranking: Function 3

Note the following effectiveness rankings of the methods on Function 3:
1) Backtracking with the Armijo-Goldstein Condition
2) Steepest Descent with the Golden-Section Search
3) Backtracking without the Armijo-Goldstein Condition
4) Gradient Descent with Momentum (with a learning rate of .2)
5) Newton's Method

This ranking presents interesting nuances in the competition for the most effective method. Thus far, the Backtracking Methods have outperformed the SD Method due to the ways the Backtracking Methods produce the next guess. These two methods analyze the function values along a line, and select a 'better' value than the current guess as the new guess. The SD Method, however, uses the golden ratio to compute shrink the search interval to find the optimal step length in a given direction.

Thus far, the Backtracking Methods have both performed more effectively than the SD Method. However, the unique characteristics of Function 2 cause limitations for the Backtracking Method without the AG Condition. Because any point on the function necessarily has a line that will lead to the center minima, the SD Method gains effectiveness. The Backtracking Method with the Armijo-Goldstein condition works effectively for all function modifications simply because its constrained condition for step size allows the method to work even when Backtracking without the AG Condition fails, and because it works even when the function is very steep around the minima (unlike the SD Method).

The GDM Method does not perform consistently for this function, and loses its effectiveness when the area around the minima become steep. As previously mentioned, Newton's Method does not work for this function, as all of its modifications produce Hessian Matrices with determinants of zero.

20. Sensitivity Analysis of Function 4, $x \sin x + y \sin y$

In the region $x \in [-6, 6]$ and $y \in [-6, 6]$, there are nine main minima of varying depths, but no single minima deeper than the rest. The most effective basins below find all nine minima in clearly defined regions.

The sensitivity analysis of the function $x \sin x + y \sin y$ is limited in that only small coefficient changes seem to preserve the location of the minima. All other changes, even slight ones, drastically change the function's contour. Many changes are then not useful to the analysis, because the analysis aims at comparing the consistency of methods across function changes that have minimal effects on the locations of the minima. The effects of coefficient changes, which increase or decrease the amplitude of the function, are shown in the analysis below.

Recall Function 4's contour plot for reference:

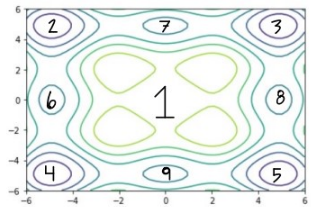
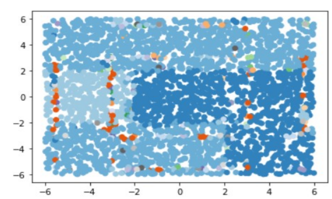
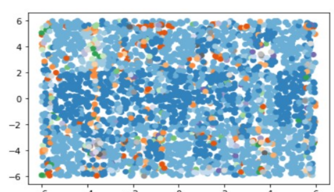
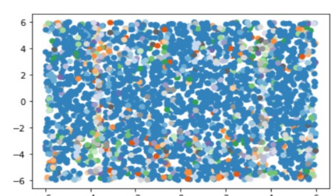
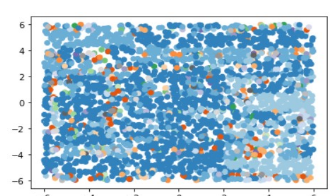
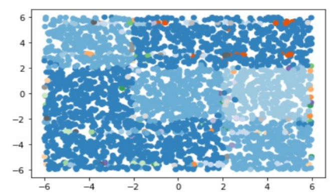
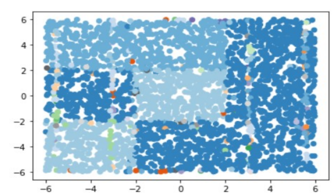
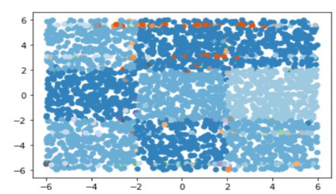
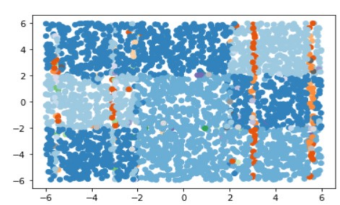


FIGURE 62. Function 4 Contour Plot

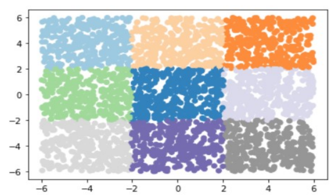
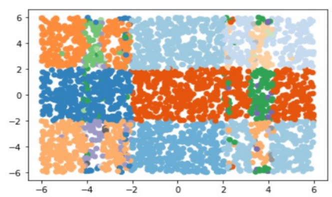
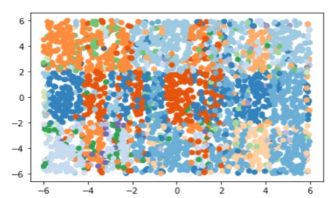
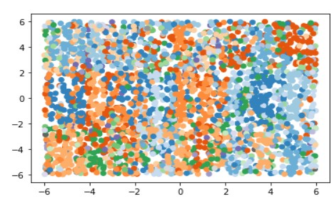
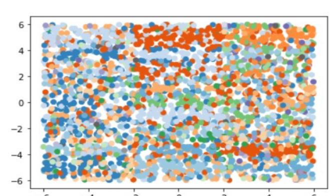
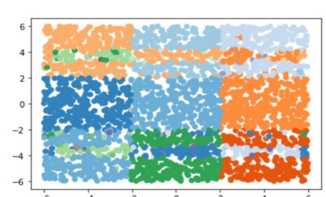
Steepest Descent Method with the Golden-Section Search

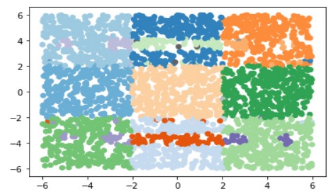
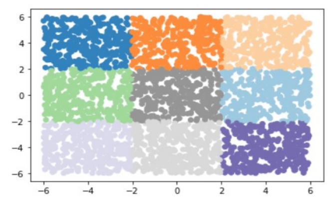
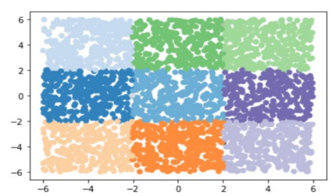
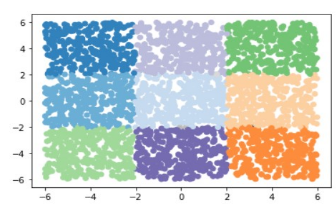
Function	Basin
Original Function	
$2x \sin x + y \sin y$	
$4x \sin x + y \sin y$	
$8x \sin x + y \sin y$	The method fails to converge
$x \sin x + 8y \sin y$	The method fails to converge
$x \sin x + 2y \sin y$	

Function	Basin
$\frac{1}{2}x\sin x + y\cos y$	
$x\sin x + \frac{1}{2}y\cos y$	
$\frac{1}{4}x\sin x + y\cos y$	
$x\sin x + \frac{1}{4}y\cos y$	

This method appropriately distinguishes nine regions, but fails to recognize distinct minima, as each basin contains largely the same colors throughout. It seems, however, that a general decrease in coefficients has a slight positive effect on the performance of the method, as the nine regions are more clearly distinguishable. Note that the decrease in coefficients decreases the steepness around the minima. Compare this trend to the Backtracking Method without the AG Condition for Function 3, which performed best with highly steep areas around the minima.

Backtracking with the Armijo-Goldstein Condition

Function	Basin
$x \sin x + y \sin y$	
$2x \sin x + y \sin y$	
$4x \sin x + y \sin y$	
$8x \sin x + y \sin y$	
$x \sin x + 8y \sin y$	
$x \sin x + 2y \sin y$	

Function	Basin
$\frac{1}{2}x \sin x + y \sin y$	
$x \sin x + \frac{1}{2}y \sin y$	
$\frac{1}{4}x \sin x + y \sin y$	
$x \sin x + \frac{1}{4}y \sin y$	

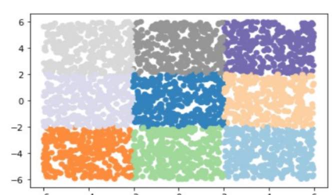
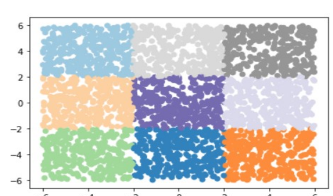
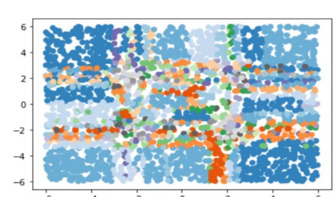
As with the SD Method, this method becomes more effective when the coefficients are decreased. This is because the functions with smaller coefficients tend to correspond with more well-defined pockets of minima. When the coefficients are increased, the minima are located in the same places, but the regions around them are less steep, and form poorly-defined pockets. The steeper the area around the individual minima, the better this method performs for functions that contain many minima close together.

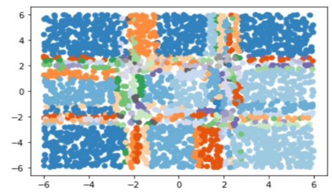
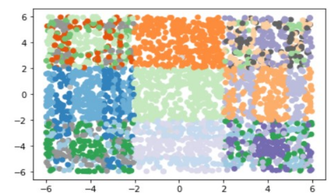
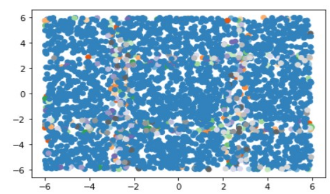
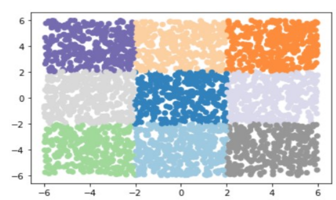
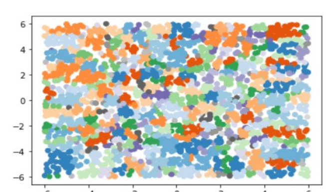
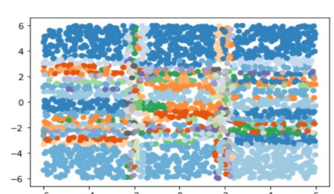
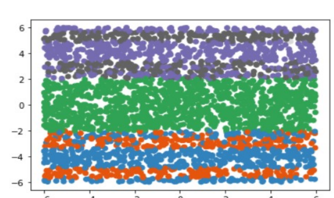
Backtracking without the Armijo-Goldstein Condition

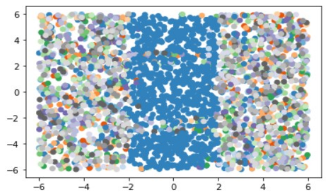
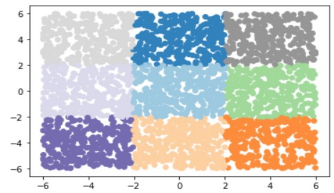
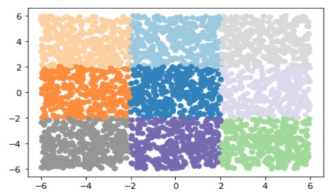
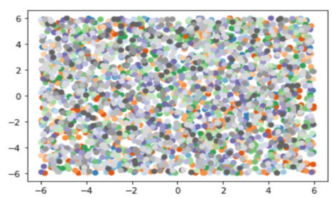
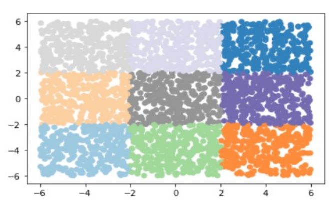
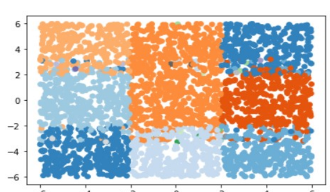
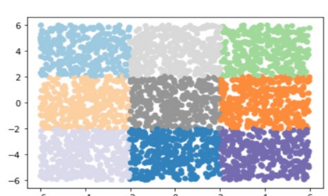
The method did not converge for the original function or any of its modifications. To review, the method works by finding a lower value along the negative gradient, and changing direction when a new guess is more optimal than a previous guess. This function presents particularly difficult challenges to the method, as it has many minima that are fairly close together. The method may begin successfully by finding better guesses after each iteration, but may eventually become stuck in an infinite cycle of bouncing back and forth from one minimum to another. Whereas the Backtracking with the Armijo-Goldstein Method has more control over finding minima because of a stricter step-size condition, this method has less accuracy, and increases the chances of non-convergence, especially in the cases of functions with many minima close together.

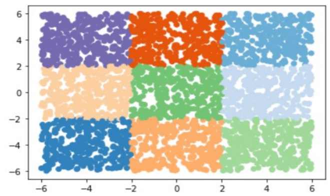
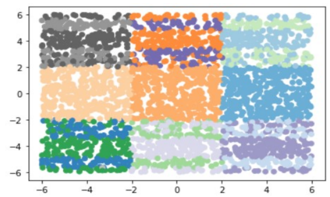
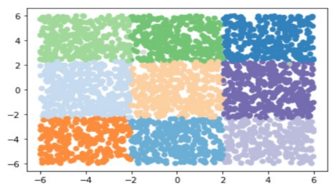
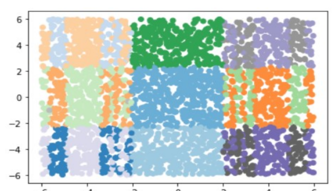
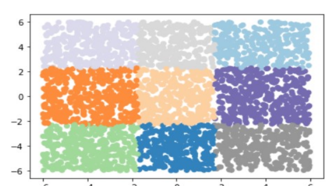
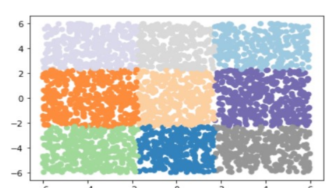
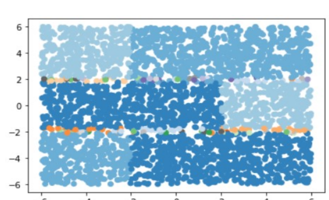
Gradient Descent with Momentum

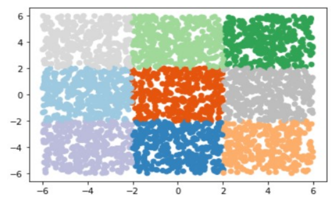
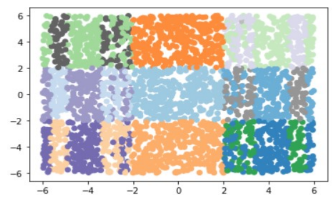
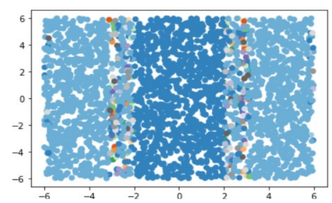
This analysis is similar to the other GDM Method analyses, where specific LR-MV values are used to explored the sensitivity of the function. Here, LR values of 0.0005, 0.05 and .5 and MV values of .2 and .9 are used.

Function	Basin
Original Function, LR = 0.05, MV = .2	
Original Function, LR = 0.05, MV = .9	
Original Function, LR = 0.0005, MV = .2	

Function	Basin
Original Function, LR = 0.0005, MV = .9	
Original Function, LR = 0.5, MV = 0.2	
Original Function, LR = 0.5, MV = 0.9	
$2x \sin x + y \sin y$, LR = 0.05, MV = .2 (and .9)	
$2x \sin x + y \sin y$, LR = 0.0005, MV = .2	
$2x \sin x + y \sin y$, LR = 0.0005, MV = .9	
$2x \sin x + y \sin y$, LR = 0.5, MV = .2	

Function	Basin
$2x\sin x + y\sin y$, LR $= 0.5$, MV $= .9$	
$4x\sin x + y\sin y$, LR $= 0.05$, MV $= 0.2$ (and .9)	
$8x\sin x + y\sin y$, LR $= 0.05$, MV $= 0.2$ (and .9)	
$8x\sin x + y\sin y$, LR $= 0.5$, MV $= 0.2$ (and .9)	
$x\sin x + 8y\sin y$, LR $= 0.05$, MV $= 0.2$	
$x\sin x + 8y\sin y$, LR $= 0.05$, MV $= 0.9$	
$x\sin x + 2y\sin y$, LR $= 0.05$, MV $= 0.2$ (and 0.9)	

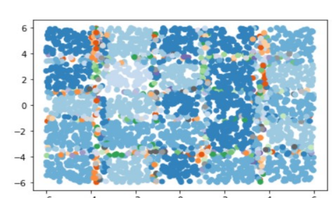
Function	Basin
$\frac{1}{2}x\sin x + y\sin y$, LR = 0.05, MV = 0.2 (and 0.9)	
$\frac{1}{2}x\sin x + y\sin y$, LR = 0.5, MV = 0.2	
$x\sin x + \frac{1}{2}y\sin y$, LR = 0.05, MV = 0.2 (and .9)	
$x\sin x + \frac{1}{2}y\sin y$, LR = 0.5, MV = 0.2	
$\frac{1}{4}x\sin x + y\sin y$, LR = 0.05, MV = 0.2 (and .9)	
$\frac{1}{4}x\sin x + y\sin y$, LR = 0.5, MV = 0.2	
$x\sin x + \frac{1}{4}y\sin y$, LR = 0.05, MV = 0.2	

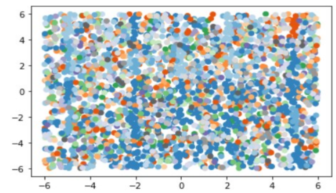
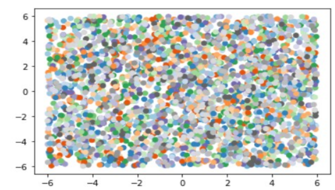
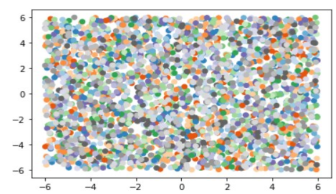
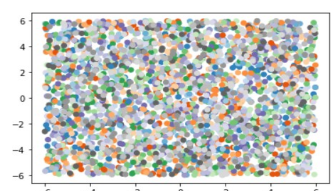
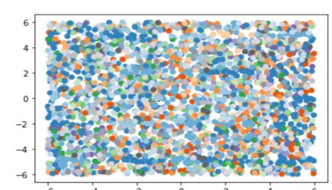
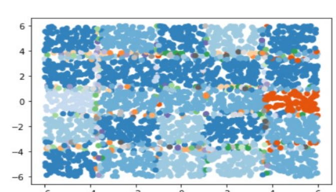
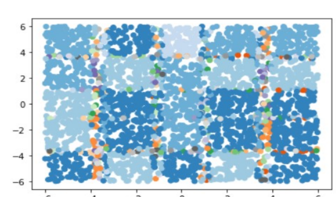
Function	Basin
$x \sin x + \frac{1}{4} y \sin y$, LR $= 0.05$, MV $= 0.9$	
$x \sin x + \frac{1}{4} y \sin y$, LR $= 0.5$, MV $= 0.2$	
$x \sin x + \frac{1}{4} y \sin y$, LR $= 0.5$, MV $= 0.9$	

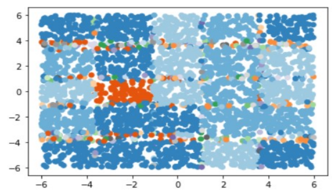
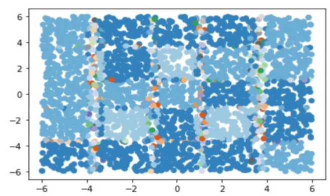
The GDM Method is very consistently accurate for this function with a learning rate of .05. For some of the functions with different LRs, the momentum value has more significant effects than with previous analyses of this method. In most cases, an MV value of .2 produces more accurate basins than functions with MV values of .9.

Note again that momentum is useful in propelling the guesses forward so as to avoid plateaus or local minima. Because the function lacks plateaus that would cause the method to stall, a large momentum value would increase the likelihood that the method overshoots the minima. Thus, a small momentum value is preferable to confine the guesses to a contained region where a minimum is held.

Newton's Method

Function	Basin
Original Function	

Function	Basin
$2x \sin x + y \sin y$	
$4x \sin x + y \sin y$	
$8x \sin x + y \sin y$	
$x \sin x + 8y \sin y$	
$x \sin x + 2y \sin y$	
$\frac{1}{2}x \sin x + y \sin y$	
$x \sin x + \frac{1}{2}y \sin y$	

Function	Basin
$\frac{1}{4}x\sin x + y\sin y$	
$x\sin x + \frac{1}{4}y\sin y$	

Compared to the other methods, the Newton Method does not perform effectively for this function. This function exhibits many of the behaviors that may cause Newton's Method to fail; for example, Newton's Method does not favor minima over maxima, which presents problems for functions with a structure like that of Function 4. In addition, there are many places where the function changes concavity, and the function has many close oscillations.

Sensitivity Analysis Ranking: Function 4

Note the following effectiveness rankings of the methods on Function 4:

1) Gradient Descent with Momentum (with LR of 0.05 and momentum of .2)
2) Backtracking with the Armijo-Goldstein Condition
3) Steepest Descent Method with the Golden-Section Search
4) Newton's Method
5) Backtracking without the Armijo-Goldstein Condition

Concluding thoughts and discussion will follow about the trends in effectiveness across the different methods, but note the following general conclusions for Function 4.

The Gradient Descent Method performs consistently well for only a specific learning rate, and even its less-optimal LR-MV modifications produce more accurate basins than many of the basins for the SD Method and Newton's Method. Furthermore, Backtracking with the Armijo-Goldstein Condition continues to perform effectively across all functions, while Backtracking without the AG Method fails due to the high amount of minima in a small region.

21. Conclusions

Note the following ranking for the methods across all four functions, where 1 represents the highest effectiveness and 5 represents the lowest effectiveness across all the methods. The most effective method for a particular function gains one point, and the least effective method gains five points. The method with the least amount of points is the most effective, and the method with the most amount of points is the least effective.

1) Backtracking with the Armijo-Goldstein Method
2) Gradient Descent with Momentum
3a) Backtracking without the Armijo-Goldstein Condition
3b) Steepest Descent with the Golden-Section Search
4) Newton's Method

Unsurprisingly, Newton's Method performs the least effectively across all functions, as there are many reasons why the method fails to produce an effective basin, or even converge. Functions 1 and 4 illustrate how many concavity changes in the function worsen the effectiveness of the method. Furthermore, Function 2 illustrates how proximity of minima affects the method, and Function 3 shows how a determinant of zero causes the method to fail to converge. Because of the many reasons that Newton's Method may fail, it is largely inapplicable in many research settings.

The tie for third between Backtracking without the AG Condition and the SD Method is an interesting display of how different trends and shortcomings of the methods can average out over time. In general, the SD Method seems to produce mediocre results that are slightly improved with lower-amplitude functions. Backtracking without the AG Condition, on the other hand, performs very efficiently in many cases, but can completely fail to converge with lower-amplitude functions or functions with many minima close together.

The GDM Method outperforms the other methods for Functions 1 and 4, but is not as effective for Functions 2 and 3. Functions 1 and 4 are similar in that they contain many minima and maxima close together in symmetric patterns. The Method does not seem to perform effectively with local and global minima very close together (as in the case of Function 2), or when a function is very steep around the minima (as in the case of Function 3). In general, low learning rates seem to optimize the performance of the method, and the momentum value is largely inconsequential except in Function 4, which helps prevent the method from overshooting the minima and converging to a different minimum elsewhere.

Overall, the Backtracking Method with the Armijo-Goldstein Condition performs the most consistently throughout the functions. Though it produces effective results for all the functions, it is especially effective for Functions 2 and 3, and less effective for Functions 1 and 4, which is the opposite trend of the GDM Method. This suggests that the effectiveness of the method is limited with functions with many minima and maxima in a close range. However, the method can handle a local minimum in the presence of a global one, as well as functions with high steepness around the minima.

This type of analysis allows the methods to be challenged in a variety of circumstances that may impact its reliability in applied contexts. For example, a method may produce effective results for a function, but lose its integrity when small changes are made to the objective function. In addition, this project's focus on visual analysis communicates the effectiveness of the methods, the shortcomings and successes of the methods in clear ways. By testing the method with a wide variety of function behaviors, this analysis also attempts to further prove the integrity of successful methods and further illustrate the shortcomings of less effective methods.

Bibliography

Bhat, R. (2020, October 22). Gradient descent with momentum. Medium. Towards Data Science. https://towardsdatascience.com/gradient-descent-with-momentum-59420f626c8f. Accessed 13 December 2021

Choosing colormaps in matplotlib. Choosing Colormaps in Matplotlib - Matplotlib 3.5.0 documentation. https://matplotlib.org/stable/tutorials/colors/colormaps.html. Accessed 13 December 2021

Difference between gradient descent method and steepest descent. (1964, March 1). Mathematics Stack Exchange. https://math.stackexchange.com/questions/1659452/difference-between-gradient-descent-method-and-steepest-descent. Accessed 13 December 2021

"Golden-Section Search." Wikipedia. Wikimedia Foundation, April 3, 2022. https://en.wikipedia.org/wiki/Golden-section-search.

Implementing backtracking line search algorithm for unconstrained optimization problem. (1966, October 1). Stack Overflow.
https://stackoverflow.com/questions/52204231/implementing-backtracking-line-search-algorithm-for-unconstrained-optimization-p. Accessed 13 December 2021

Klockner, A. Steepest Descent. Steepest descent. https://andreask.cs.illinois.edu/cs357-s15/public/demos/12-optimization/Steepest

Levy, Adam B. Attraction in Numerical Minimization. Cham, Switzerland: Springer, 2019.

Levy, A. B. (2017). Basics of practical optimization. Orient Blackswan.

Line search in gradient and Newton Directions. Line search in gradient and Newton directions - Computational Statistics and Statistical Computing 1.0 documentation. https://people.duke.edu/ ccc14/sta-663-2018/notebooks/S09E-Optimization-Line-Search.html. Accessed 13 December 2021

Newton's Method. Newton's method. http://www.ltcconline.net/greenl/courses/105/applications/NEWT.HTM. Accessed 13 December 2021

Saeed, M. (2020, November 4). Gradient descent in Python: Implementation and theory. Stack Abuse. Stack Abuse. https://stackabuse.com/gradient-descent-in-python-implementation-and-theory/. Accessed 13 December 2021