An Exploration of the Nonlinear Optimization of Polynomial Functions Through the Use of Differential Calculus.

IB Mathematics

Word Count: 3035

May 2019

Contents

1 Introduction

With noticeable impact in areas such as economics, the petroleum industry, telecommunications networks, and shipping,[1] **mathematical programming**, which includes the study of linear and nonlinear optimization problems, has developed significantly throughout the twentieth century. Although past contributions to this field have been made by mathematicians such as Hitchcock,[2] Fourier,[3] and Kantorovich,[4] it was not until WWII and soon thereafter in 1947 that the basis of today's modern linear optimization algorithms were founded by George B. Dantzig and John von Neumann.[5] Dantzig in June 1947, working out of a Pentagon funded U.S. Air Force program "Scientific Computation of Optimal Programs," developed the Simplex Algorithm, an algorithm that allows for the optimization of linear programming problems.[6] Whereas former mathematicians had solved for specific optimization cases through techniques that did not involve the boundary of a geometric space, Dantzig's Simplex Algorithm followed the edges of the geometrically confined space (as defined by the original conditions of the problem) in order to reach the solution point.[7]

[1]Overton Michael L., "Linear Programming.," *Draft for Encyclopedia Americana*, 1997,

[2]Hitchcock Frank L., *The Distribution of a Product from Several Sources to Numerous Localities*, vol. 20, 1-4, 224–230, doi:10.1002/sapm1941201224, eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/sapm1941201224, https://onlinelibrary.wiley.com/doi/abs/10.1002/sapm1941201224.

[3]Fourier, Jean Baptiste Joseph. "Solution d'une question particulière du calcul des inégalités." Chapter. In Oeuvres de fourier: Publiées par les soins de gaston darboux, edited by Jean Gaston Darboux, 2:315–19. Cambridge Library Collection - Mathematics. Cambridge: Cambridge University Press, 2013. doi:10.1017/CBO9781139568159.016.

[4]Kantorovich, L.V "Mathematical Methods in the Organization and Planning of Production," Publication House of the Leningrad State University, 1939, 68 pp. Translated in Management Science, Vol. 6, 1960, pp. 366-422.

[5]Overton, "Linear Programming"

[6]Gass, Saul I., and Arjang Assad. An Annotated Timeline of Operations Research: An Informal History. New York: Kluwer Academic Publishers, 2005, 61.

[7]Dantzig, G.B. (1987c). "Origins of the Simplex Method," in S.G Nash (ed.), Proceedings of the ACM Conference on a History of Scientific Computing, ACM Press, Addison-Wesley Publishing Company, 1990, 141–151.

2 Linear and Nonlinear Optimization

Linear programming is a mathematical field comprising of **linear optimization**, a set of procedures for maximizing or minimizing a function, known as an **"objective function"** subject to a series of requirements on the given variables known as **"constraint equations"**, in which both the objective function and the constraint equations are of linear nature. [8] An example follows:

Example 1.

$$\text{Maximize } P(x_1, x_2) = x_1 + 2x_2$$
$$\text{Subject to: } 5x_1 + 8x_2 \ < \ 14$$
$$x_2 - x_1 \ < \ 5$$
$$x_1 < 15$$
$$x_1 \geqslant 0, \ x_2 \ \geqslant \ 0$$

Example 1. is of linear nature as it involves the maximization of solely linear functions. We note that the final constraint equation requires both variables to be nonnegative, a constraint which is often implied.

This example may be solved either graphically or through the use of the Simplex Method, a technique used for the optimization of linear functions.

Nonlinear optimization focuses on maximizing or minimizing objective functions where either the objective function or the constraint equation is of nonlinear nature.

An example follows:

Example 2.

$$\text{Maximize } P(x_1, x_2) = 200 \ \left(-(x_1 - 9)^2 \ - \ (x_2 - 3)^2 \right)$$
$$\text{Subject to: } 5x_1 + 8x_2 \ < \ 14$$
$$x_2 - x_1 \ < \ 5$$
$$x_1 < 15$$
$$x_1 \geqslant 0, \ x_2 \ \geqslant \ 0$$

Example 2. is of nonlinear nature as it involves the maximization of a polynomial function with degree 2.

This example may not be solved through the use of the Simplex Method, as it is not of linear nature. A

[8] Strayer, James. Linear Programming and Its Applications. Springer, 1989. p. 1

graphical analysis can help lead towards an optimal solution but is not sufficient to guarantee that the point is a maximum or a minimum.

3 Statement of Problem

I will be exploring and analyzing different algorithmic methods so as to find a solution to my research question:

How can I maximize or minimize nonlinear optimization problems over two variables, whose objective or constraint equations are polynomials with degree two or greater?

In this work I will begin by discussing the problem of finding a solution to a linear optimization problem via the Simplex method and graphical analyses and show why the Simplex method is inefficient when used on nonlinear optimization problems. I will then attempt to develop a technique used for the optimization of nonlinear functions using differential calculus and graphical analysis. The graphing of an optimization problem with n variables leads to the graphing of the corresponding objective function in dimension $n+1$. Since visualizing four or more dimensions poses a significant logistical problem, throughout this work I will be focusing on optimization problems with two variables accompanied by three dimensional graphical images. This exploration involves linear and nonlinear optimization techniques, differential calculus, as well as methods of graphical analysis.

4 The Simplex Method

The Simplex method provides a way to solve the objective function by iterating over and finding solutions to sets of equations.[9] This process introduces **slack variables** in order to convert the inequality into an equation, and the problem into a **canonical slack maximization** and **canonical slack minimization** linear programming problem.[10] The Simplex method then encodes these variables into a tableau, known as a **Tucker tableau** (Table 1.) and makes use of an iterative process of modifying the tableau's rows in order to achieve a function that maximizes or minimizes the objective function.

Slack variables are commonly represented by $t_1, t_2, t_3, \dots\ t_m$.[11] A canonical slack maximization problem is in the following form[12] :

[9] Wisniewski, Mik, and Jonathan Klein. Linear Programming. Palgrave MacMillan, 2001. p. 30
[10] Strayer, *Linear Programming*, 29
[11] Ibid.
[12] Ibid, 28

$$\text{Maximize } f(x_1, x_2, \ldots, x_n) = c_1x_1 + c_2x_2 + \ldots c_nx_n - d$$

$$\text{Subject to: } a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n - b_1 = -t_1$$

$$a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n - b_2 = -t_2$$

$$\ldots$$

$$a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n - b_m = -t_m$$

$$t_1, t_2, \ldots t_m \geqslant 0$$

$$x_1, x_2, \ldots x_n \geqslant 0$$

This canonical slack maximization problem corresponds with the following Tucker tableau:

Table 1: A Tucker tableau

x_1	x_2	...	x_n	-1	
a_{11}	a_{12}	...	a_{1n}	b_1	$= -t_1$
a_{21}	a_{22}	...	a_{2n}	b_2	$= -t_2$
$\vdots$	$\vdots$		$\vdots$	$\vdots$	
a_{m1}	a_{m2}	...	a_{mn}	b_m	$= -t_m$
c_1	c_2	...	c_n	d	$=f$

The first step of the Simplex Method involves choosing a numerical element of the Tucker tableau which will be used to rearrange a Tucker tableau in the subsequent step.[13]

Step 1.

(1) Make sure that at least one c value is positive. If no c value is positive, then the optimal solution is already shown

(2) Choose a positive c_j value

(3) Make sure at least one a value in the same column as c_j is positive. If no a value is positive, then the problem is not bounded

(4) List all values of b_i/a_{ij} such that $a_{ij} > 0$ as i ranges from 1 to m inclusive. Find the minimum of the listed values. This minimum result will be in the form b_p/a_{pj}.

Let the pivot point a_{pj} be known as p.

The second step of the Simplex Method involves the transformation of a Tucker tableau into another

[13]Strayer, *Linear Programming*, 29

Tucker tableau which is closer to the optimal solution. [14] The steps are outlined below:

Step 2.

(1) Switch the variable's shown on the same row and column as p but do not change the plus or minus signs

(2) Change p to $1/p$

(3) Divide every entry in the same row as p by p

(4) Divide every entry in the same column as p by $-p$

(5) If a number s is not in the same column or row as p, then find the number in the same column as s and in the same row as p and denote this number q.

(6) Find the number in the same row as s and in the same column as p and denote this number r

(7) Replace s with $(ps - qr)/p$

Shown below is a solution to Example 1 utilizing the Simplex Method. After its implementation, a valid solution to the maximization problem is found.

Table 2: The initial tableau representing Example 1

x_1	x_2	-1	
1	0	15	$= -t_a$
-1	1	5	$= -t_b$
-5	-8	-14	$= -t_c$
1	2	0	P

$$c_1 > 0, c_2 > 0$$

Choose

$$c_1 = 1$$
$$\min\{\frac{b_1}{a_1 1} = \frac{15}{1}\} = 15$$

[14]Ibid, 34

Table 3: After Step 1 of the Simplex Method, the pivot element is selected (∗).

x_1	x_2	-1	
1*	0	15	$= -t_a$
−1	1	5	$= -t_b$
−5	−8	−14	$= -t_c$
1	2	0	P

Sample Calculation:

$$1 \to 1/1 = 1,\ 0 \to 0/1 = 0,\ 15 \to 15/1 = 15$$

$$-1 \to -(-1/1) = 1,\ 1 \to (ps - qr)/p = (1\cdot 1 - 0\cdot(-1))/1 = 1,$$

$$5 \to (ps - qr)/p = (1\cdot 5 - 15\cdot(-1))/1 = 20$$

$$-5 \to -(-5/1) = 5, -8 \to (ps - qr)/p = (1 \cdot (-8) - 0 \cdot (-5))/1 = -8$$

$$-14 \to (ps - qr)/p = (1 \cdot (-14) - 15 \cdot (-5))/1 = 61$$

$$1 \to -1/1 = -1,\ 2 \to (ps - qr)/p = (1 \cdot 2 - 0 \cdot 1)/1 = 2,$$

$$0 \to (ps - qr)/p = (1\cdot 0 - 15\cdot 1) = -15$$

Table 4: Elements are rearranged according to Step 2 of the Simplex Method.

t_a	x_2	-1	
1	0	15	$= -x_1$
1	1	20	$= -t_b$
5	−8	61	$= -t_c$
−1	2	−15	P

Table 5: After Step 1 of the Simplex Method, the pivot element is selected (∗).

t_a	x_2	-1	
1	0	15	$= -x_1$
1	1	20	$= -t_b$
5	−8	61	$= -t_c$
−1	2	−15	P

Table 6: Elements are rearranged according to Step 2 of the Simplex Method.

t_a	x_2	-1	
1	0	15	$= -x_1$
1	1	20	$= -t_b$
5	-8	61	$= -t_c$
-1	2	-15	P

As the bottom row is all non-positive, an optimal solution has been reached. Setting t_a and t_b to 0 yields a value of 15 and 20 for x_1 and x_2 respectively. These choices result in a slack variable value for t_c of 221 and the maximum value of P of 55.

An equally valid solution may be reached graphically, as the fundamental theorem of linear programming states that values at vertices will be the maximum or minimum values. [15] To find all possible vertices, the constraint inequalities may be converted into equations and then successive pairs of equations may be solved.

$$5x_1 + 8x_2 < 14 \rightarrow x_2 = \frac{14 - 5x_1}{8} \quad (5)$$

$$x_2 - x_1 < 5 \rightarrow x_2 = 5 + x_1 \quad (6)$$

$$x_1 < 15 \rightarrow x_1 = 15 \quad (7)$$

Equating (5) & (6):

$$5 + x_1 = \frac{14 - 5x_1}{8} \rightarrow 40 + 5x_1 = 14 - 5x_1 \rightarrow 10x_1 = 54 \rightarrow x_1 = \frac{27}{5} \therefore x_2 = \frac{52}{5}$$

$$P(x_1, x_2) = P\left(\frac{27}{5}, \frac{52}{5}\right) = \frac{27}{5} + 2\left(\frac{52}{5}\right) = \frac{131}{5} = 26.2$$

Equating (5) & (7):

$$x_1 = 15 \therefore x_2 = \frac{14 - 5(15)}{8} = \frac{-61}{8}$$

$$P(x_1, x_2) = P\left(15, \frac{-61}{8}\right) = 15 + 2\left(\frac{-61}{8}\right) = -0.25$$

Equating (6) & (7):

$$x_1 = 15 \therefore x_2 = 5 + 15 = 20$$

$$P(x_1, x_2) = P(15, 20) = 15 + 2(20) = 55$$

[15] Tardella, F. The fundamental theorem of linear programming: extensions and applications, Optimization, 60:1-2, 283-301, DOI: 10.1080/02331934.2010.506535

Figure 1: Shows the objective functions and constraint equations for Example 1.

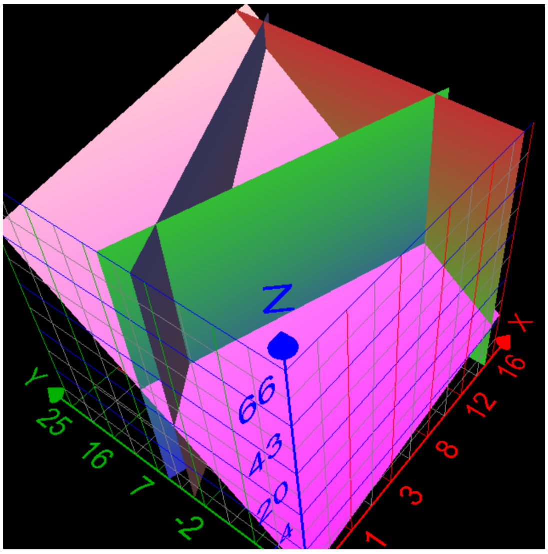
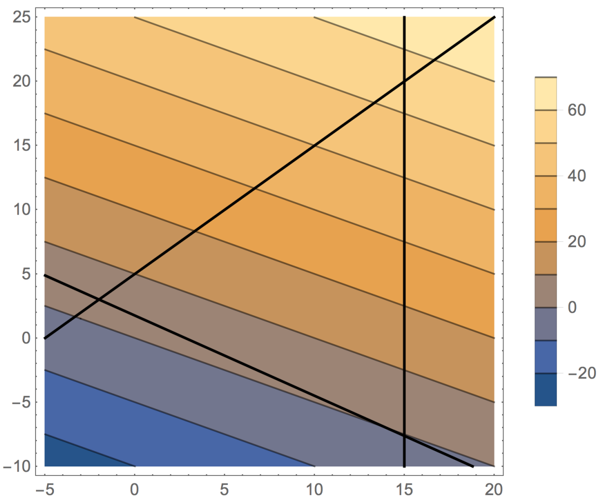


Figure 2: Shows the contour plot of the objective functions and constraint equations for Example 1. This allows us to reduce the 3D graph into 2 dimensions so that values of the objective function may be examined more easily.



Visually, Figure 2. confirms the solution found through the Simplex Method, as the maximal point appears to be on the vertex at (15, 20) with a value that approximates 60.

5 Nonlinear Optimization

When it comes to nonlinear optimization, the technique implemented in linear optimization is insufficient for the following reasons: The added component of an extra degree or nonlinear function renders the Simplex Method unusable; the optimal value may occur at an interior, boundary, or extreme point within the feasibility region, thus invalidating the Simplex Method which looks only for extreme points[16]; and the theorem of linear programming does not hold for nonlinear problems signifying that the optimal solution may occur at any value within the feasible region determined by the constraint equations.

Since our solution range has increased drastically, the number of points we must examine increases significantly as well. One way to reduce the feasibility range is to follow the **gradient** of maximal or minimal ascent. This gradient will lead us in the direction of a point in which the objective function will change to the greatest extent. By continually following the gradient, one will eventually reach the minimal or maximal point of the objective function. The gradient of a function over two variables is equal to the vector whose components are the derivatives of the function with respect to both x_1 and x_2.
Another successful method includes setting both components of the gradient to 0, simultaneously solving for x_1 and x_2 to obtain critical points of the function, and then testing these critical points by plugging them into the objective function in order to find a local maximum/minimum.

5.1 Nonlinear Optimization Example 2.

Looking at Example 2.'s objective function, we can find its gradient by using impartial differentiation; first we take the derivative with respect to x_1 and then with respect to x_2 to obtain:

$$\frac{\partial f(x_1, x_2)}{\partial x_1} = -400 \cdot (-9 + x_1)$$

$$\frac{\partial f(x_1, x_2)}{\partial x_2} = -400 \cdot (-3 + x_2)$$

$$\text{D} = \{-400 \cdot (-9 + x_1),\ -400 \cdot (-3 + x_2)\}$$

[16]Bradley, Stephen P., Arnoldo C. Hax, and Thomas L. Magnanti. Applied Mathematical Programming. Reading, MA: Addison-Wesley, 1992, p.413.

The first element of D is denoted D_{x1} and the second element of D is denoted D_{x2}. Next, we must normalize the vector D, or convert it to a form in which its magnitude is 1. We normalize the vector so that its magnitude may be determined through other means, either arbitrarily (Method 1.) or systematically (Method 2.). Mathematica's "Normalize" function was utilized in order to find the following normalized vector N[17]:

$$N = \left\{ -\frac{400\cdot(x_1-9)}{\sqrt{160000\cdot|x_1-9|^2+160000\cdot|x_2-3|^2}}, -\frac{400\cdot(x_2-3)}{\sqrt{160000\cdot|x_1-9|^2+160000\cdot|x_2-3|^2}} \right\}$$

Where the first element of N is denoted N_{x1} and the second element of N is denoted N_{x2}.

5.2 Method 1:

Proceeding, we choose a point at random within the constraint equations and follow the normal line until the maximum point is reached. Figure 3. And Table 7. jointly show how this process, in which the line starting at length l is multiplied by $\frac{2}{3}$ in magnitude upon each consecutive step, asymptotically leads to the optimal point as the number of steps $n \to \infty$.
The general form of this process is associated with the following algorithm:

Algorithm 1: Finding the optimal solution through approximation with arbitrary starting line length l and arbitrary ratio k.

(1) Choose a point $A = (c_{x1}, c_{x2})$ that meets all of the constraint equations, i.e. is within the feasibility range

(2) Set l to an arbitrary length less than the domain of the function

(3) Update c_{x1} to $c_{x1} + l \cdot N_{x1}$ and c_{x2} to $c_{x2} + l \cdot N_{x2}$

(4) Divide l by $k \in (1, \infty)$

(5) Repeat steps 3 to 5 "n" number of times

(6) The final value of $A = (c_{x1}, c_{x2})$ will approximate the optimal solution

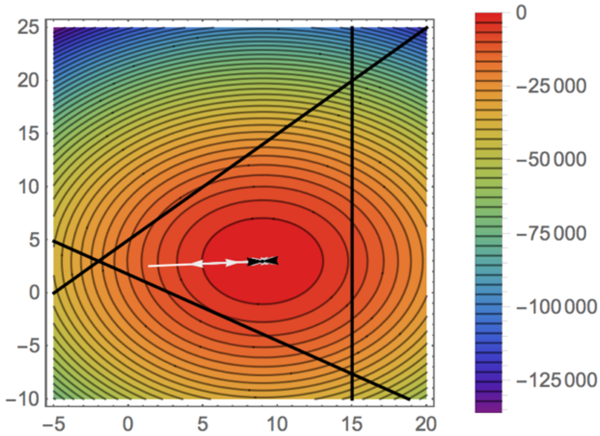
Table 7. Compares key elements of the current point at each step throughout Algorithm 1. with a total of n=16 steps.

Trial Number	Line length l:	x_1	x_2	$P(x_1, x_2)$
0	8	1.3219121	2.5301852	-11834.752

[17] See Appendix A

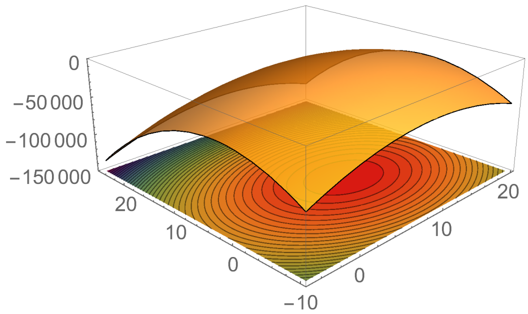
1	4	9.3069776	3.0187837	-18.917615
2	2	3.9836006	2.6930514	-5051.6961
3	1	7.5325186	2.9102062	-432.31292
4	2^{-1}	9.8984639	3.0549761	-162.05197
5	2^{-2}	8.3211670	2.9584629	-92.507903
6	2^{-3}	9.3726983	3.0228050	-27.88482
7	2^{-4}	8.6716775	2.9799103	-21.639857
8	2^{-5}	9.1390247	3.0085068	-3.8800462
9	2^{-6}	8.8274599	2.9894424	-5.9763114
10	2^{-7}	9.0351698	3.0021520	-0.2483085
11	2^{-8}	8.8966965	2.9936790	-2.1423137
12	2^{-9}	8.9890120	2.9993277	-0.0242376
13	2^{-10}	9.0505557	3.0030935	-0.5130890
14	2^{-11}	9.0095266	3.0005829	-0.0182190
15	2^{-12}	8.9821738	2.9989092	-0.0637925
16	2^{-13}	9.0004090	3.0000250	-0.0000335

Figure 3: The steps taken by following the normal line to reach the maximum value. Line gets darker as n increases.



Through the contour plot shown in Figure 3., the maximal value of the objective function becomes much more apparent as it is narrowed down to a smaller feasibility area with the changing colour gradient.

Figure 4: Shows the objective functions and corresponding contour plot in 3D. Visually the higher z values correspond to a redder tint in the corresponding contour plot.



5.3 Method 2:

Although Table 7. does approach the optimal value of the function, it is computationally inefficient as it requires a large number of steps n in order to increase its accuracy. To mitigate this deficiency, a value of l can more methodically be chosen so that it when added to the current point it reaches an optimal value at a much faster convergence rate and is more computationally efficient. The following algorithm describes this process.

Algorithm 2: Finding the optimal solution through approximation with a specific line length l at each step n where the variable l is not predetermined.

(1) Choose a point $A = (c_{x1}, c_{x2})$ that meets all of the constraint equations, i.e. is within the feasibility range

(2) Initiate a new variable l

(3) Update c_{x1} to $c_{x1} + l \cdot N_{x1}$ and c_{x2} to $c_{x2} + l \cdot N_{x2}$

(4) Plug in c_{x1} and c_{x2} into the objective function $f(x_1, x_2)$ to obtain $g(c_{x1}, c_{x2}) = f(c_{x1}, c_{x2})$[18]

(5) Solve for l by setting g' to equal 0

(6) Update c_{x1} to $c_{x1} + l \cdot N_{x1}$ and c_{x2} to $c_{x2} + l \cdot N_{x2}$, since the value of l has been set to a constant

(7) Repeat steps 2 to 6 "n" number of times

(8) The final value of $A = (c_{x1}, c_{x2})$ will approximate the optimal solution

[18]Note that the value of c_{xp} has been modified to $c_{xp} + l \cdot N_{xp}\ \forall p \in \{1, 2\}$ in the previous step.

Table 8. Compares key elements of the current point at each step throughout Algorithm 2 with a total of n=3 steps.

Trial Number	Line length l:	x_1	x_2	$P(x1, x2)$
0	7.69245	1.3219121	2.5301852	-11834.752
1	0	9.0000000	3.0000000	0.0000000
2	0	9.0000000	3.0000000	0.0000000
3	0	9.0000000	3.0000000	0.0000000

As shown in Table 8., Algorithm 2 converges on an optimal solution on the first trial, yielding much more optimal results in terms of computational efficiency than those found in Table 7.
The process by which this optimal solution was found is as follows:

(1) $A = (c_{x1}, c_{x2}) \to A = (1.3219121, 2.5301852)$

(2) $c_{x1} = 1.3219121 + 0.9981330 \cdot l, c_{x2} = 2.5301900 + 0.0610748 \cdot l$

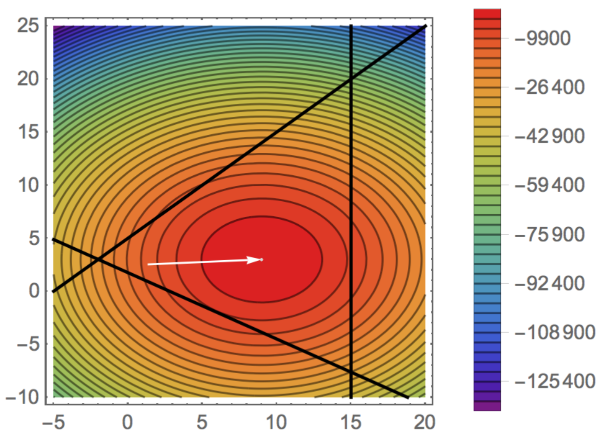
(3) $g(c_{x1}, c_{x2}) = f(c_{x1}, c_{x2}) = 200(-(-0.469815 + 0.0610748 \cdot l)^2 - (-7.67809 + 0.998133 \cdot l)^2)$

(4) $l = 7.69245$

(5) Update c_{x1} and c_{x2} to $c_{x1} + l \cdot N_{x1}$ and $c_{x2} + l \cdot Nx2$ respectively

(6) Step 5. results in the point $B = (9.0000000, 3.0000000)$

Figure 5: Shows the steps taken by following Algorithm 2



5.4 Method 3:

To find the maximum value of $f(x_1, x_2)$ using differential calculus, without regard for the constraint equation, one may set $\frac{\partial}{\partial x_1} f(x_1, x_2) = 0$ and $\frac{\partial}{\partial x_2} f(x_1, x_2) = 0$ yielding

$$-400(x_1 - 9) = 0 \rightarrow x_1 = 9$$
$$-400(x_2 - 3) = 0 \rightarrow x_2 = 3$$

Or the point $P = (9, 3)$. Where $P_0 = 9$ and $P_1 = 3$. To guarantee that this value is the maximum value, one may perform the second derivative test.

$$Let\ D = D(P_0,\ P_1) = f_{x1x1}(P_0,\ P_1)\ \cdot\ f_{x2x2}(P_0,\ P_1) - [f_{x1x2}(P_0,\ P_1)]^2$$
$$f_{x1x1} = \frac{\partial f(x_1, x_2)}{\partial x_1^2} = -400$$
$$f_{x2x2} = \frac{\partial f(x_1, x_2)}{\partial x_2^2} = -400$$
$$f_{x1x2} = \frac{\partial}{\partial x_1} \frac{\partial}{\partial x_2} f(x_1, x_2) = 0$$
$$D = -400 \cdot (-400) - (0)^2 = 160000$$

As $D > 0$ and $f_{x1x1}(P_0, P_1) < 0$, there exists a relative maximum at P.[19]
The point P lies within the feasibility range of Example 2., and satisfies all of the constraint equations, therefore it is an optimal solution to this optimization problem.

6 Conclusion

Throughout the various examples which were presented and analyzed throughout this paper, it is primordial that differential calculus plays an important role in the optimization of nonlinear functions. Formally defining two different methods of optimization, it was found that the method relying on a systematic approach to find the magnitude of the directional vector line which is added to the original point, performs far better than the method whose selection of magnitude is arbitrary. This research, however, has certain limitations. The behaviour of the two models was not tested near the constraint equations, nor was it tested on functions with nearly identical maximum or minimum values. If the models were tested near constraint equations, there is a possibility that the line towards the optimal point would lead towards a point outside of the constraint equations. Furthermore, if a function with

[19] "Calculus III - Relative Minimums and Maximums." Calculus II - Approximating Definite Integrals. Accessed October 02, 2018. http://tutorial.math.lamar.edu/Classes/CalcIII/RelativeExtrema.aspx.

extremely similar optimality points was tested, then a new model must be developed in order to account for this semblance, as Method 1 and Method 2 would likely lead to differing optimal solutions upon numerous iterations.

7 Future Work

Future work includes the analysis and evaluation of many more variations of nonlinear optimization problems including more complex functions such as trigonometric functions, functions with degree greater than 2, or functions with three or more variables. Although Algorithm 2 poses a systematic solving method, the creation of a less systematic method of narrowing down the optimal point using randomly generated points spread throughout the graph may yield successful results. As the behaviour of points near the constraint equations is difficult to ascertain, an analysis of Method 1 and Method 2 near the constraint equations would be beneficial. An additional method could rely on the use of linear regression in order to convert a nonlinear function into a set of linear equations which can be solved via the Simplex Method.

8 Software Used

Throughout this extended essay, the program “Wolfram Mathematica” has been used extensively in order to graphically display optimization problems and efficiently format tables. The application “Graphing Calculator 3D” has been used to graph three dimensional functions. Code relevant to this paper may be found in Appendix A. LaTeX originally developed by Leslie Lamport is used for the typesetting of this paper.

Appendix A.

Mathematica Code used to create Figure 2.

```
p1 = ContourPlot[x + 2 y, {x, -5, 20}, {y, -10, 25},
  PlotLegends -> Automatic]
f[x_] := x + 5
g[y_] := 15
h[x_] := (14 - 5 x)/8
p2 = Plot[{f[x], h[x]}, {x, -5, 20}, PlotRange -> {-10, 25}, PlotStyle -> Black]
p3 = ParametricPlot[{15, y}, {y, -10, 25}, PlotRange -> {{-5, 20}, {-10, 25}}, AspectRatio ->
0.5, PlotStyle -> Black]
img = Show[p1, p2, p3]
```

Mathematica Code used to create Figure 4.

```
graph2 = Module[{f}, f[x_, y_] := 200 (-(x - 9)^2 - (y - 3)^2);
  Show[Plot3D[f[x, y], {x, -5, 20}, {y, -10, 25},
    PlotStyle -> Opacity[0.8],
    PlotRange -> {Automatic, Automatic, {-150000, 500}},
    Mesh -> None],
   Graphics3D[
    ContourPlot[f[x, y], {x, -5, 20}, {y, -10, 25}, Axes -> False,
      ColorFunction -> "Rainbow", Contours -> 40][[
     1]] /. {x_Real, y_Real} -> {x, y, -150000}],
   ViewPoint -> {-1.8, -1.8, 1}, ViewAngle -> 0.7, ImageSize -> 300]]
```

Mathematica Code used in Method 2 and for the creation of Figure 5. And Table 8.

```
f[x_] := x + 5;
g[y_] := 15;
h[x_] := (14 - 5 x)/8;
p1 = Plot[{f[x], h[x]}, {x, -5, 20}, PlotRange -> {-10, 25},
   PlotStyle -> Black];
p2 = ParametricPlot[{15, y}, {y, -10, 25},
   PlotRange -> {{-5, 20}, {-10, 25}}, AspectRatio -> 0.5,
   PlotStyle -> Black];
p4 = ContourPlot[
   200 (-(x - 9)^2 - (y - 3)^2), {x, -5, 20}, {y, -10, 25},
   PlotLegends -> Automatic, Contours -> 40,
   ColorFunction -> "Rainbow"];
img = Show[p4, p1, p2];
f[x_, y_] := 200*(-(x - 9)^2 - (y - 3)^2)
gx[x_] := Derivative[1, 0][f][x, 0]
gy[y_] := Derivative[0, 1][f][0, y]
ox = 1.3219120704162712`;
oy = 2.5301852263773057`;
arrows = {};
mult = 8;
```

```
xpoints = {};
ypoints = {};
PP = {};
Do[norm = Normalize[{gx[ox], gy[oy]}];
 cx = ox + z*Part[norm, 1]; cy = oy + z*Part[norm, 2];
 optz = Solve[D[f[cx, cy], z] == 0]; mult = z /. optz[[1]];
 cx = ox + mult*Part[norm, 1]; cy = oy + mult*Part[norm, 2];
 ar = Graphics[{GrayLevel[1 - (n - 1)/25],
    Arrow[{{ox, oy}, {cx, cy}} ], Frame -> False,
    PlotRange -> {{-1.5, 1.5}, {-1.5, 1.5}},
    AspectRatio -> 1/GoldenRatio, Axes -> True}];
 arrows = Append[arrows, ar]; ox = cx; oy = cy;
 xpoints = Append[xpoints, StringForm["``", NumberForm[cx, 20] ]];
 ypoints = Append[ypoints, StringForm["``", NumberForm[cy, 20] ]];
 PP = Append[PP, StringForm["``", NumberForm[f[cx, cy], 20] ]], {n, 5}]
app = Show[img, arrows[[1 ;; 5]]]
```

Mathematica Code which outputs the normalized function N:

```
[1, Input] Normalize[{-400*(-9 + x1), -400*(-3 + x2)}, {x1, x2}]
[2, Output] {-((400 (-9 + x1))/{x1, x2}[{-400 (-9 + x1), -400 (-3 + x2)}]), -((
  400 (-3 + x2))/{x1, x2}[{-400 (-9 + x1), -400 (-3 + x2)}])}
```

Bibliography

Bradley, Stephen P., Arnoldo C. Hax, and Thomas L. Magnanti. Applied Mathematical Programming. Reading, MA: Addison-Wesley, 1992, p.413.

Calculus III - Relative Minimums and Maximums." Calculus II - Approximating Definite Integrals. http://tutorial.math.lamar.edu/Classes/CalcIII/RelativeExtrema.aspx.

Dantzig, G.B. (1987c). "Origins of the Simplex Method," in S.G Nash (ed.), Proceedings of the ACM Conference on a History of Scientific Computing, ACM Press, Addison-Wesley Publishing Company, 1990, 141–151.

Fourier, Jean Baptiste Joseph. "Solution D'une Question Particulière Du Calcul Des Inégalités." Chapter. In Oeuvres De Fourier: Publiées Par Les Soins De Gaston Darboux, edited by Jean Gaston Darboux, 2:315–19. Cambridge Library Collection - Mathematics. Cambridge: Cambridge University Press, 2013. doi:10.1017/CBO9781139568159.016.

Gass, Saul I., and Arjang Assad. An Annotated Timeline of Operations Research: An Informal History. New York: Kluwer Academic Publishers, 2005, 61.

Hitchcock, F.L., "The Distribution of a Product from Several-Sources to Numerous Localities," J. Math. Phys., Vol. 20, 1941, pp. 224-230.

Kantorovich, L.V., "Mathematical Methods in the Organization and Planning of Production," Publication House of the Leningrad State University, 1939, 68 pp. Translated in Management Science, Vol. 6, 1960, pp. 366-422.

Overton, Michael L. "Linear Programming." Linear Programming. December 20, 1997. https://cs.nyu.edu/overton/g22_lp/encyc/article_web.html.

Strayer, James. *Linear Programming and Its Applications.* Springer, 1989

Tardella F. The fundamental theorem of linear programming: extensions and applications, Optimization, 60:1-2, 283-301, 2011. DOI: 10.1080/02331934.2010.506535

Wisniewski, Mik, and Jonathan Klein. *Linear Programming.* Palgrave MacMillan, 2001.