MAE 5930 - Optimization

Fall 2019

Homework 1

Jared Hansen

Due: 7:30 AM, Thursday September 12, 2019

A-number: `A01439768`

e-mail: `jrdhansen@gmail.com`

Purpose: the problems assigned help develop your ability to

- recognize and formulate optimization problems
- convert formulations into code using MATLAB's `linprog`, `quadprog`, and `fmincon`.
- find the minimum and infimum values of sets and functions.
- find the minimum values and locations of differentiable, unconstrained functions.

NOTE: you are welcome to use the equivalent functions in Python.

1. Consider the following table of nutritional data:

Foods	Price ($) per serving	Calories per serving	Fat (g) per serving	Protein (g) per serving	Carbs (g) per serving
Carrots	0.14	23	0.1	0.6	6
Potatoes	0.12	171	0.2	3.7	30
Bread	0.20	65	0	2.2	13
Cheese	0.75	112	9.3	7	0
Peanut Butter	0.15	188	16	7.7	2

You must consume at least 2000 calories, 50g of fat, 100g of protein, and 250g of carbs. The goal is to meet these requirements and minimize the total cost of buying your food.

(a) Formulate this into an optimization problem by defining the variables, objective, and constraints.

- **Variables:**
  - r = # of servings of carrots
  - p = # of servings of potatoes
  - b = # of servings of bread
  - h = # of servings of cheese
  - e = # of servings of peanut butter $\longrightarrow$ vector $x = [r, p, b, h, e]^T$

- **Objective Function:**
  $f(x) = (.14)(r) + (.12)(p) + (.2)(b) + (.75)(h) + (.15)(e)$ where $x = [r, p, b, h, e]^T$

- **Constraints:** I'll go into depth on how I framed the inequality constraint for fat in the diet as a model for how I arrived at the three additional constraints (calories, protein, carbs). The prompt specifies that we must consume at least 50g of fat in our diet. We can reach this minimum by adjusting $x = [r, p, b, h, e]^T$; in other words, by adjusting how many servings of each food we eat. Mathematically this constraint is:
  $\Big[[fatPerServingColumn]^T * [x] \leq 50\Big] = \Big[[0.1, 0.2, 0, 9.3, 16] * [r, p, b, h, e]^T \leq 50\Big]$
  $= \Big[(0.1)(r) + (0.2)(p) + (0)(b) + (9.3)(h) + (16)(e) \leq 50\Big]$
  To enter this constraint into MATLAB we must have it in the form $\big[someValue\big] \leq 0$, so we re-arrange it to get: $\Big(50 - \Big[(0.1)(r) + (0.2)(p) + (0)(b) + (9.3)(h) + (16)(e)\Big] \leq 0\Big)$
  - **Calorie inequality constraint:** $\Big(2000 - \Big[[caloriesPerServingColumn]^T * [x]\Big] \leq 0\Big)$
    $= \Big(2000 - \Big[(23)(r) + (171)(p) + (65)(b) + (112)(h) + (188)(e)\Big] \leq 0\Big)$
  - **Fat inequality constraint:** $\Big(50 - \Big[[fatPerServingColumn]^T * [x]\Big] \leq 0\Big)$
    $= \Big(50 - \Big[(0.1)(r) + (0.2)(p) + (0)(b) + (9.3)(h) + (16)(e)\Big] \leq 0\Big)$
  - **Protein inequality constraint:** $\Big(100 - \Big[[proteinPerServingColumn]^T * [x]\Big] \leq 0\Big)$
    $= \Big(100 - \Big[(0.6)(r) + (3.7)(p) + (2.2)(b) + (7.0)(h) + (7.7)(e)\Big] \leq 0\Big)$
  - **Carbs inequality constraint:** $\Big(250 - \Big[[carbsPerServingColumn]^T * [x]\Big] \leq 0\Big)$
    $= \Big(250 - \Big[(6)(r) + (30)(p) + (13)(b) + (0)(h) + (2)(e)\Big] \leq 0\Big)$

(b) Write the problem in vector-matrix form suitable for MATLAB's `linprog`.

- Expressing things as vectors and matrices congruent with `linprog`:
$$\left((f^T)(x)\right) = \begin{bmatrix}0.14 & 0.12 & 0.20 & 0.75 & 0.15\end{bmatrix}\begin{bmatrix}r & p & b & h & e\end{bmatrix}^T$$
$$\left(Ax \le b\right) = \begin{bmatrix}-23 & -171 & -65 & -112 & -188\\ -0.1 & -0.2 & 0 & -9.3 & -16\\ -0.6 & -3.7 & -2.2 & -7.0 & -7.7\\ -6 & -30 & -13 & 0 & -2\end{bmatrix}\begin{bmatrix}r\\ p\\ b\\ h\\ e\end{bmatrix} \le \begin{bmatrix}-2000\\ -50\\ -100\\ -250\end{bmatrix}$$
$$\left(lb \le x \le ub\right) = \left(\begin{bmatrix}0 & 0 & 0 & 0 & 0\end{bmatrix}^T \le \begin{bmatrix}r & p & b & h & e\end{bmatrix}^T \le \begin{bmatrix}\infty & \infty & \infty & \infty & \infty\end{bmatrix}^T\right)$$
- Objective function (coefficients):
```
f = [0.14 0.12 0.20 0.75 0.15];
```
- Constraints (matrix A):
```
A(1,:)  = [-23 -171 -65 -112 -188];
A(2,:)  = [-0.1 -0.2 0 -9.3 -16];
A(3,:)  = [-0.6 -3.7 -2.2 -7.0 -7.7];
A(4,:)  = [-6 -30 -13 0 -2;
```
- RHS of constraints (vector):
```
b = [-2000; -50; -100; -250];
```
- Lower bounds for x (servings vector):
```
lb = zeros(5,1);
```
- Upper bounds for x (servings vector):
```
ub = [inf; inf; inf; inf; inf;];
```

(c) Solve the problem using MATLAB's `linprog`. Provide your code and solution.

For code and output, see **CODE APPENDIX** at end of homework printout.

- The solution is: $x = \begin{bmatrix}r\\ p\\ b\\ h\\ e\end{bmatrix} = \begin{bmatrix}0\\ 7.7147\\ 0\\ 0\\ 9.2800\end{bmatrix}$

(d) Suppose only whole servings can be purchased. Solve the problem using MATLAB's `intlinprog`. Provide your code and solution. Explain how the integer solution is different than the continuous solution.

For code and output, see **CODE APPENDIX** at end of homework printout.

- The solution is: $x = \begin{bmatrix} r \\ p \\ b \\ h \\ e \end{bmatrix} = \begin{bmatrix} 0 \\ 9 \\ 0 \\ 0 \\ 9 \end{bmatrix}$
- With the additional requirement that our serving values must be integers we can see that our integer solution is similar to our continuous solution: the only foods we eat are potatoes and peanut butter, and the number of servings is fairly similar. Explaining this colloquially, we make up for the fact that we can't eat a little over 9 servings of peanut butter and a little under 8 servings of potatoes by eating a little less peanut butter and a little more potatoes in order to keep the number of servings as whole numbers (9 servings for both foods, 0 for the rest).
  It's safe to assume that with the integer solution we will spend more money on food than in the continuous solution. It may also be the case that we eat a greater number of calories, fat, protein, and/or carbs. This is due to the additional inefficiency introduced by having to eat whole number values for the number of servings.

2. Think about an optimization problem that you "solve" in your daily life.

<u>**The problem:**</u> time utility

(a) <u>Define the decision variables.</u> (Note: units for each var is hours)

- z = time spent on Optimization class
- c = time spent on C++ class
- r = time spent on thesis research
- t = time spent on TA assignment
- f = time spent on social life
- e = time spent eating
- s = time spent sleeping
- h = time spent on health (exercise) $\longrightarrow$ vector $x = \begin{bmatrix} z & c & r & t & f & e & s & h \end{bmatrix}^T$

(b) <u>Define the objective function.</u>

- lifeOutput $=$ $\Big[$ OptznOutput + CppOutput + ResearchOutput + TAoutput + SocialLifeOutput + EatingOutput + SleepingOutput + HealthOutput $\Big]$
- lifeOutput$(x) = \left[[rowVectorOfWeights] * \begin{bmatrix} z & c & r & t & f & e & s & h \end{bmatrix}^T \right]$

(c) Define the constraints.

- All values ≥ 0 and ≤ 24:
$\left(\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \leq \begin{bmatrix} z & c & r & t & f & e & s & h \end{bmatrix}^T \leq \begin{bmatrix} 24 & 24 & 24 & 24 & 24 & 24 & 24 & 24 \end{bmatrix}^T \right)$
- 24 hours in a day: $\|x\|_1 = \sum_{i=1}^{len(x)} x_i = 24$
- Sleep: $s \geq 7.0$ (not that this happens, but speaking ideally)
- Exercise: $0.5 \leq h \leq 1.5$
- Eating: $2.0 \leq e \leq 4.0$
- Optzn (Tu,Thr): $z \geq 1.25$
- C++ (MWF): $c \geq \frac{5}{6}$

3. In MATLAB create a noisy data set based on a quadratic function with coefficients all equal to 1.

```
>> x = linspace(0,1)' ;
>> y = 1 + 1*x + 1*x.^2 ;
>> ym = y + 0.1*randn(100,1) ;
```

(a) Plot the quadratic function and the noisy data.

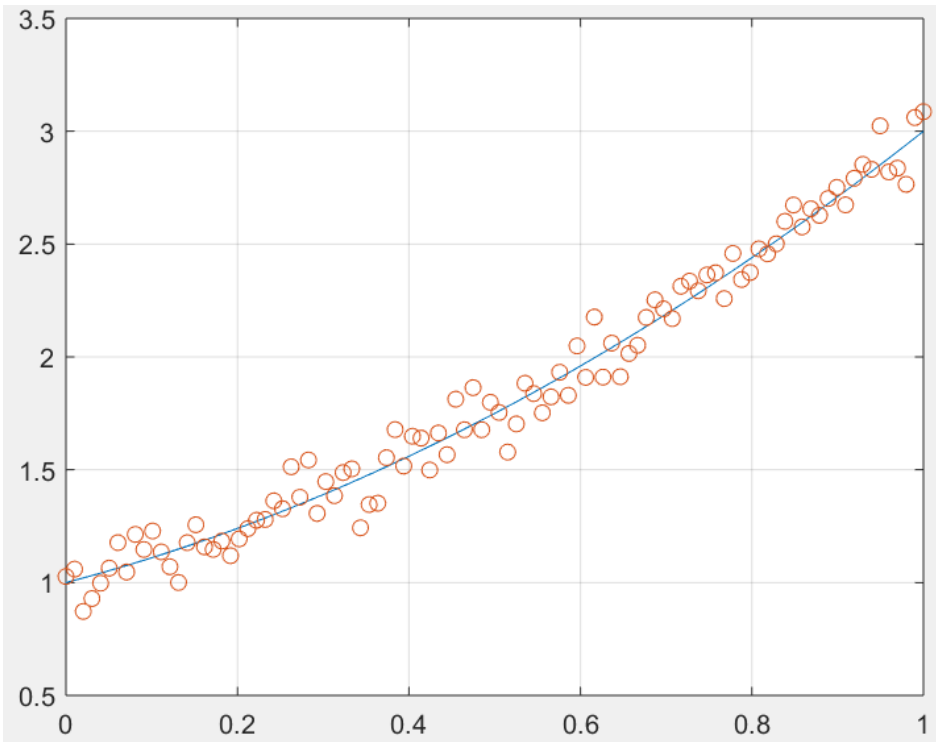

Figure 1: Noisy data and overlaid original quadratic function

(b) Formulate the coefficient estimation problem as an optimization problem.

- We have a set of 100 observed points made up of ordered pairs. Let's denote these pairs as $(x_i, y_i), i \in \{1, 2, \ldots, 100\}$
- Based on the prompt, and a visual inspection of the noisy data, we assume a quadratic model: $\hat{f}(x) = ax^2 + bx + c = \hat{y}$
- Let's define the vector of coefficient estimates (**<u>our control variables</u>**) to be: $d = \begin{bmatrix} a & b & c \end{bmatrix}^T$
- Now let's define errors: $e_i = \big(y_i - \hat{y}_i\big) = \big(y_i - (a{x_i}^2 + bx_i + c)\big)$
- Our goal is to minimize SSE (sum of squared errors):

$$\left[min \sum_{i=1}^{100} (y_i - \hat{y}_i)^2\right] = \left[min \sum_{i=1}^{100} (y_i - \hat{y}_i)(y_i - \hat{y}_i)\right] = \left[min \sum_{i=1}^{100} ({y_i}^2 - 2(y_i)(\hat{y}_i) + {\hat{y}_i}^2)\right] =$$

$$\left[min \sum_{i=1}^{100} \Big({y_i}^2 - 2(y_i)(a{x_i}^2 + bx_i + c) + (a{x_i}^2 + bx_i + c)^2\Big)\right] =$$

$$\left[min \sum_{i=1}^{100} \Big(-2(y_i)(a{x_i}^2 + bx_i + c) + (a{x_i}^2 + bx_i + c)^2\Big)\right]$$

since the ${y_i}^2$ term is constant w.r.t. the coefficient a, b, c (so we can remove it).

- Now let's write a sum in terms of matrix-vector multiplication in order to define our objective:
$\sum_{i=1}^{100}(ax_i{}^2 + bx_i + c) = Md = \begin{bmatrix} x_1{}^2 & x_1 & 1 \\ x_2{}^2 & x_2 & 1 \\ \vdots & \vdots & \vdots \\ x_{100}{}^2 & x_{100} & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}$ for the matrix M and the vector d.
- Now, let's rewrite what we're minimizing from sums to matrices and vectors:
$\left[\sum_{i=1}^{100}\bigg(-2(y_i)(ax_i{}^2 + bx_i + c) + (ax_i{}^2 + bx_i + c)^2\bigg)\right] = \left[\bigg(-2y^TMd + d^TM^TMd\bigg)\right]$
Since it won't affect the values of d that minimizes this quantity, let's multiply by a constant:
$\left[\frac{1}{2}\bigg(-2y^TMd + d^TM^TMd\bigg)\right] = \left[\bigg(\frac{1}{2}d^TM^TMd - y^TMd\bigg)\right]$
- Therefore our <u>**objective function**</u> is: $f(d) = \left(\frac{1}{2}d^TM^TMd - y^TMd\right)$ which we minimize by finding optimal values for $d = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$
- Aside from restricting the equation to being a quadratic, we don't have any <u>**constraints**</u> other than $\big(d \in \mathbb{R}^3\big)$ also written as $\big((a \in \mathbb{R}), (b \in \mathbb{R}), (c \in \mathbb{R})\big)$

(c) Use MATLAB's `quadprog` to solve the problem.

For code and output, see **CODE APPENDIX** at end of homework printout.
I'm not sure how to set a seed for the random number generation in MATLAB, but when I ran the code seen in the appendix I got an answer of:
$d = \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 1.0396 \\ 0.9724 \\ 1.0049 \end{bmatrix}$

(d) Why are the estimated coefficients not all one?

The short answer is because we added noise to the function-generated y values. Expounding on this, by adding noise we created a scenario in which it was nearly certain that coefficients slightly different from $d = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T$ would minimize the SSE (sum of squared errors). Unless the noise added was perfectly balanced (e.g. an "equal amount" of error above the function line and below the function line) we were bound to get something slightly different than $d = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T$.
To verify this, I've included code that shows that the SSE of the modeled quadratic (0.9543) is <u>less than</u> the SSE of the data-generating quadratic (0.9564). This shows that our modeled quadratic describes the noisy data more accurately than the original data-generating quadratic since it takes into account the added noise and minimizes the SSE better than any other set of coefficients, including $\begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T$.

4. For each of the following functions, define the range. Using the range, determine the minimum value and infimum value. If they do not exist, explain why.

   (a) $f(x) = 0$ with domain $\mathbb{R}$
   (b) $f(x) = cos(x)$ with domain $\mathbb{R}$
   (c) $f(x) = x^3$ with domain $\mathbb{R}$
   (d) $f(x) = -e^{-x}$ with domain $[0, -\infty)$
   (e) $f(x) = sin(x)$ with domain $\left[0, \frac{3\pi}{2}\right)$

Function	Domain	Range	Minimum	Infimum
(a) $f(x) = 0$	$\mathbb{R}$	$\{0\}$	0	0
(b) $f(x) = cos(x)$	$\mathbb{R}$	$[-1, 1]$	-1	-1
(c) $f(x) = x^3$	$\mathbb{R}$	$(-\infty, \infty)$	DNE: $f \to -\infty$ as $x \to -\infty$ (always $\exists$ a smaller $-\infty$)	see below
(d) $f(x) = -e^{-x}$	$[0, -\infty)$	$[-1, 0)$	-1	-1
(e) $f(x) = sin(x)$	$\left[0, \frac{3\pi}{2}\right)$	$(-1, 1]$	DNE: can forever find an x s.t. $f(x) \to -1$ more closely	-1

"see below" = DNE: say we pick some value $a \in \mathbb{R}$ s.t. $a = -\infty$. We can then define some $b \in \mathbb{R}$ s.t. $b = (a - 1)$, e.g. b is a 'more negative' $-\infty$ than a since $b < a$. We can keep doing this interminably. Therefore, there is no value in the defined range of $(-\infty, \infty)$ that fits the definition of being the infimum. Here the infiumum would be the greatest element in the range that is $\leq$ to all elements of the range, which is impossible to achieve since there always exists a "more negative" $-\infty$. (This same logic is why $\nexists$ minimum of this function.)

5. Do the following:

(a) Prove the following implication: Let f be a real-valued, differentiable function on an open interval of $\mathbb{R}$. If x minimizes f then $f'(x) = 0$.

- If we have a local minimum at $x^* \implies f(x^*) \leq f(x) \forall x$ s.t. $\mid x - x^* \mid \leq \delta$ where $\delta > 0$ (we define a δ-neighborhood around x^*)
- It must be the case that $f(x^*) \leq f(x^* + \delta)$. Rearrange this to get $f(x^* + \delta) - f(x^*) \geq 0$.
- Since $\delta > 0$ we can say that $\left[\frac{f(x^* + \delta) - f(x^*)}{\delta} \geq \frac{0}{\delta}\right] = \left[\frac{f(x^* + \delta) - f(x^*)}{\delta} \geq 0\right]$
- Since $\delta > 0$ we can take the limit of both sides of the inequality as $\delta \to 0$ from the right:

  $$\left[\lim_{\delta \to 0^+} \left(\frac{f(x^* + \delta) - f(x^*)}{\delta}\right) \geq \lim_{\delta \to 0^+} \left(0\right)\right] = \left[\lim_{\delta \to 0^+} \left(\frac{f(x^* + \delta) - f(x^*)}{\delta}\right) \geq 0\right]$$
  $$= \left[f'(x^*) \geq 0\right]$$

- Although we've specified that $\delta > 0$ we know that in order for a limit to exist the RHS limit and LHS limit must be equal (and exist). For the sake of checking this condition now assume that $\delta < 0$, **which flips the sign of the inequality since we're dividing by a negative**. This gives:

  $$\left[\lim_{\delta \to 0^-} \left(\frac{f(x^* + \delta) - f(x^*)}{\delta}\right) \leq \lim_{\delta \to 0^-} \left(0\right)\right] = \left[\lim_{\delta \to 0^-} \left(\frac{f(x^* + \delta) - f(x^*)}{\delta}\right) \leq 0\right]$$
  $$= \left[f'(x^*) \leq 0\right]$$

- As mentioned above, in order for the limit to exist at x^* it must be the case that $\left[\text{RHS limit at } x^*\right] = \left[\text{LHS limit at } x^*\right]$ which is $\left[f'(x^*) \geq 0\right] = \left[f'(x^*) \leq 0\right]$.
  By ovservation, we can see that this condition only holds for $f'(x^*) = 0$. Therefore we have shown that for a real-valued function f defined on an open interval of $\mathbb{R}$, if x^* minimizes f then $f'(x^*) = 0$, proving the desired result.

(b) Find a non-differentiable function where this condition fails.

The function $f(x) = |x|$; $x^* = 0$ minimizes f (value of 0) but $f'(0)$ DNE.
This is a function for which there is an x^* that minimizes f, but for which $f'(x^*) \neq 0$, in this case since $f'(x^*)$ does not exist since f is not differentiable at $x^* = 0$.

(c) Find a differentiable function defined on a closed interval where this condition fails.

The function $f(x) = x^3$ on $[-1, 1]$. Here $x^* = -1$ minimizes f, but $f'(-1) = 3(-1)^2 = 3 \neq 0$, failing the condition as desired.

6. For each of the following functions defined on $\mathbb{R}$, use the first derivative condition to determine the minimum value (if it exists) and all locations where the minimum occurs.

(a) $f(x) = x^2$

- $f'(x) = 2x$ set $= 0 \to 2x = 0 \to x = 0$ is a critical value.
- Since $f''(x) = 2 > 0 \implies$ minimum of $f(x) = x^2$ occurs at $x = 0$, that minimum of f being $f(0) = 0^2 = 0$.
- ANSWER: min f is 0, min occurs at $x = 0$

(b) $f(x) = x^3$

- $f'(x) = 3x^2$ set $= 0 \to 3x^2 = 0 \to x = 0$ is a critical value.
- $f''(x) = 6x$, since $\nabla^2{}_f = 6x \not> 0, \forall x \in \mathbb{R} \implies$ a min of f does not occur at $x = 0$, therefore $f(x) = x^3$ has no minimum on $\mathbb{R}$.
- ANSWER: min f DNE

(c) $f(x) = cos(x)$

- $f'(x) = -sin(x) \to$ set $-sin(x) = 0 \to sin(x) = 0$. Thus all x s.t. $sin(x) = 0$ are critical points. This includes: $\{x = \pi n\}, n \in \mathbb{Z}$
- $\nabla^2{}_f = f''(x) = -cos(x) \to -cos(x) > 0, \forall x \in \{(2n+1)\pi\}$ where $n \in \mathbb{Z}$, so the function $f(x) = cos(x)$ achieves its minimum of -1 at $x \in \{\ldots, -3\pi, -\pi, \pi, 3\pi, \ldots\}$
- ANSWER: min f is -1, min occurs at $x \in \{\ldots, -3\pi, -\pi, \pi, 3\pi, \ldots\} = \{x = (2n+1)\pi\}, n \in \mathbb{Z}$

(d) $f(x) = (x-a)^2(x+a)^2$

- Expand f giving: $f(x) = (x-a)(x+a)(x-a)(x+a) = (x^2-a^2)(x^2-a^2) = x^4 - 2a^2x^2 + a^4$
- $\left[f'(x) = 4x^3 - 4a^2x\right]$ set $= 0 \to \left[4x(x^2-a^2) = 0\right] \to \left[x(x^2-a^2) = 0\right]$
  $\to \left[f'(x) = 0 \text{ for } x \in \{-a, 0, a\}\right]$ (these are our critical values)
- $\nabla^2{}_f = f''(x) = 12x^2 - 4a^2$ set $f''(x) > 0$. Is $12x^2 - 4a^2 > 0$ for any of $\{-a, 0, a\}$?
- For $-a$: $12(-a)^2 - 4a^2 = 8a^2$ which is > 0 if $a > 0$
- For 0: $12(0)^2 - 4a^2 = -4a^2 \not> 0$
- For a: $12(a)^2 - 4a^2 = 8a^2$ which is > 0 if $a > 0$
- From this, we deduce that the min of f occurs at $-a$ and a, but not at 0. However, let's say that $a = 0$, giving $f(x) = (x^2)(x^2) = x^4$ and we know $argminf(x) = x^4$ is $x = 0$. So we can say that f has a min at $x = 0$ if $a = 0$, BUT this is already covered under a minimum occurring at both a and $-a$ since $[0 = -0]$ is like $[a = -a]$.
- Knowing this, the minimum value of f is
  $\left[f(-a) = (-a-a)^2(-a+a)^2 = 0\right] = \left[f(a) = (a-a)^2(a+a)^2 = 0\right] = 0$
- ANSWER: min f is 0, min occurs at $\{-a, a\}$

7. Do the following:

(a) Use the first derivative condition to find the minimum value and location of the function $f(x,y) = (\alpha - x)^2 + \beta(y - x^2)^2$.

- $f'(x,y) = \nabla_f = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} -2\alpha + 2x - 4\beta xy + 4\beta x^3 \\ 2\beta y - 2\beta x^2 \end{bmatrix}$ set $= \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ giving
$\begin{bmatrix} -2\alpha + 2x - 4\beta xy + 4\beta x^3 \\ 2\beta y - 2\beta x^2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$
- Now multiply both sides by $\dfrac{1}{2}$, giving:
$\begin{bmatrix} -\alpha + x - 2\beta xy + 2\beta x^3 \\ \beta y - \beta x^2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ Multiply both sides of bottom equation by $(2x)$ giving:
$\begin{bmatrix} -\alpha + x - 2\beta xy + 2\beta x^3 \\ 2\beta xy - 2\beta x^3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$
- Now add the LH sides and RH sides together (method of elimination) to get:
$\Big[\big(-\alpha + x - 2\beta xy + 2\beta x^3\big) + \big(2\beta xy - 2\beta x^3\big) = (0) + (0)\Big] =$
$\Big[-\alpha + x = 0\Big] \implies \Big[x = \alpha\Big]$ **But what is y?**
- We know from the bottom equation that $\beta y - \beta x^2 = 0$. Substituting our solved-for $x = \alpha$ we have $\Big[\beta y - \beta(\alpha)^2 = 0\Big] \to \Big[\dfrac{\beta y}{\beta} = \dfrac{\beta\alpha^2}{\beta}\Big] \to \Big[y = \alpha^2\Big]$
- Therefore, the critical point, and assumed minimum w/out checking the 2^{nd}-order condition, is the vector $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \alpha \\ \alpha^2 \end{bmatrix}$
- I found that the Hessian $\nabla^2{}_f = \begin{bmatrix} \dfrac{\partial^2 f}{\partial x^2} & \dfrac{\partial^2 f}{\partial x \partial y} \\ \dfrac{\partial^2 f}{\partial y \partial x} & \dfrac{\partial^2 f}{\partial y^2} \end{bmatrix} = \begin{bmatrix} 2 - 4\beta y + 12\beta x^2 & -4\beta x \\ -4\beta x & 2\beta \end{bmatrix}$

  I tried checking that the Hessian is positive definite, but got stuck at finding eigenvalues of $\lambda = (1-\beta) \pm \sqrt{\beta^2 + 8\beta\alpha^2 + 2\beta + 1}$ and couldn't think of a way to prove that they're both positive. It seems like λ_1 and λ_2 might only be positive for certain pairs of α and β though.

- With the minimum location of $(x,y) = (\alpha, \alpha^2)$ we find the minimum of f to be:
$\Big[f(\alpha,\alpha^2) = (\alpha - \alpha)^2 + \beta(\alpha^2 - (\alpha^2))^2 = (0) + \beta(0)\Big] = 0$
- <u>ANSWER:</u> the minimum value of f is 0, the location of the minimum is $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \alpha \\ \alpha^2 \end{bmatrix}$

(b) Use MATLAB's `fmincon` to solve the problem with $\alpha = 1$ and $\beta = 100$. Try different initial guesses. Provide your code and brief discussion of your observations.

- For code and output, see **CODE APPENDIX** at end of homework printout.
- With values of $\alpha = 1$ and $\beta = 100$ we find that the minimum is achieved at $(x, y) = (1, 1)$, giving a minimum of 0 (the analytic minimum we found in part (a) was 0 as well).
- I tried a few different initial guesses for $x_0 = (x, y)$. If the guess was too far away from the solution of $(1, 1)$ then the solver takes too many iterations to converge and MATLAB gives up.
  For example, if you use $x_0 = (500, 1000)$ like I tried then it won't work. On the other hand, a more proximal guess like $x_0 = (3, 3)$ had no problem converging to the correct answer $(1, 1)$ in a number of iterations acceptable for the `fmincon` function.

8. Do the following:

(a) Use the first derivative condition to find all local minima of $f(x) = x + 2sin(x)$.

- $\left[f'(x) = 1 + 2cos(x)\right]$ set $= 0 \to \left[cos(x) = -\frac{1}{2}\right] \to \left[cos^{-1}(cos(x)) = cos^{-1}\left(\frac{-1}{2}\right)\right]$
  $\to \left[x = \left\{\ldots, -\frac{4\pi}{3}, \frac{2\pi}{3}, \frac{8\pi}{3}, \ldots\right\}\right] \to \left[x = \left\{\frac{(2+6n\pi)}{3}\right\}, n \in \mathbb{Z}\right]$
- $\left[f''(x) = -2sin(x)\right] \to \left[-2sin(x) > 0\right]$ for any of the critical pts in $\{x\}$?
  $\left[\frac{-2sin(x)}{-2} > \frac{0}{-2}\right]$ flip sign $\to \left[sin(x) < 0\right]$ for any of $\{x\}$?
- $sin(x) = \frac{\sqrt{3}}{2}$ $\forall x_i \in \{x\}$ and since $\left[\frac{\sqrt{3}}{2} \not< 0\right] \implies$ the function f has no global minimum.
- Instead, the function has local minima occuring $\forall x_i \in \{x\}$.

  These local minima are $\left\{\ldots, \sqrt{3} - \frac{4\pi}{3}, \sqrt{3} + \frac{2\pi}{3}, \sqrt{3} + \frac{8\pi}{3}, \ldots\right\} = \left\{\sqrt{3} + \frac{(2+6n\pi)}{3}, n \in \mathbb{Z}\right\}$
- Here is a graphical representation of the function on $x \in (-10, 10)$:

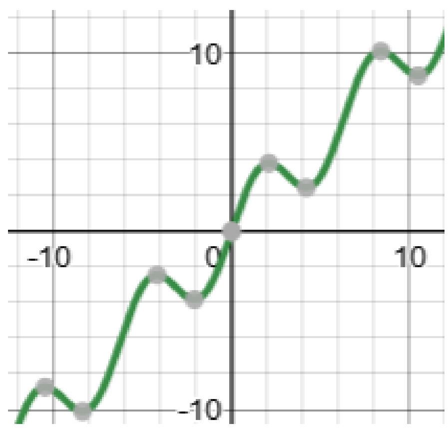

Figure 2: The function $f(x) = x + 2sin(x)$ on $x \in (-10, 10)$

(b) How many local minima are there?

From looking at Figure 2 we can tell that there are an infinite number of local minima since there are infinitely many values in $\mathbb{Z}$. Based on our set of critical points, this implies an infinite amount of values x satisfying $f'(x) = 0$ since the set of critical points maps one-to-one with $\mathbb{Z}$. To be more specific, since the set of critical points maps one-to-one with $\mathbb{Z}$, and since $\mathbb{Z}$ can be mapped one-to-one with $\mathbb{N}$, we can say that there are **countably infinite** local minima.

(c) Does a global minimum exist? If so, what is it?

I'll transcribe the third-to-last bullet from part (a) to answer this:

- $sin(x) = \frac{\sqrt{3}}{2}\ \forall x_i \in \{x\}$ where $\left[\{x\} = \left\{\frac{(2+6n\pi)}{3}\right\}, n \in \mathbb{Z}\right]$ is the set of all critical points.

  Since $\left[\frac{\sqrt{3}}{2} \not< 0\right] \implies \left[\nabla^2{}_f \not> 0 \text{ for the critical points} \in \{x\}\right] \implies$ the function f has no global minimum.

CODE APPENDIX

```
%==========================================================================
%== ASSIGNMENT: hw1
%== AUTHOR: Jared Hansen
%== DUE: Thursday, 09/12/2019
%==========================================================================



clear all; close all; clc;
%==========================================================================
%==========================================================================
%== PROBLEM 1
%==========================================================================
%==========================================================================

%==========================================================================
%== 1(B)
%==========================================================================
% Coefficients of the objective function (price per serving in $ for
% carrots, potatoes, bread, cheese, and peanut butter respectively)
f = [0.14  0.12  0.20  0.75  0.15];
% Creating the matrix that specifies caloric and macro-nutrient content
A(1,:) = [-23   -171   -65   -112   -188];   % >= 2000 calorie constraint
A(2,:) = [-0.1  -0.2   0     -9.3   -16];    % >= 50g fat constraint
A(3,:) = [-0.6  -3.7   -2.2  -7.0   -7.7];   % >= 100g protein constraint
A(4,:) = [-6    -30    -13   0      -2];     % >= 250g carbs constraint
b = [-2000; -50; -100; -250];                % RHS of ineq constraints
lb = zeros(5,1);                   % Lower bounds for # of servings
ub = [inf; inf; inf; inf; inf;];   % Upper bounds for # of servings

%==========================================================================
%== 1(C)
%==========================================================================
% Use linprog function to solve (cont_soln = continuous solution)
cont_soln = linprog(f, A, b, [], [], lb, ub);
% Display the continuous solution
disp("Continuous solution:")
disp(cont_soln);

%==========================================================================
%== 1(D)
%==========================================================================
% Use intlinprog function to solve (int_soln = integer solution)
int_soln = intlinprog(f, [1:5], A, b, [], [], lb, ub);
% Display the integer solution
disp("Integer solution:")
disp(int_soln);
```

```
clear all; close all; clc;
%=========================================================================
%=========================================================================
%== PROBLEM 3
%=========================================================================
%=========================================================================
% DIRECTIONS: in MATLAB create a noisy data set based on a quadratic
% function with coefficients all equal to 1.
x  = linspace(0,1)';
y  = 1 + 1*x + 1*x.^2;
ym = y + 0.1*randn(100,1);

%=========================================================================
%== 3(A)
%=========================================================================
figure, plot(x, y, '-', x, ym, 'o'), grid on

%=========================================================================
%== 3(C)
%=========================================================================
% Matrix containing polynomial terms for each x
M = [x.^2, x, ones(100,1)];
% The quadratic term, H, for the quadprog function
H = (M')*(M);
% The linear term, f, for the quadprog function
f = -((ym')*(M));
% Estimate the coefficients d=[a,b,c] where (y_hat = ax^2 + bx + c)
fitted = quadprog(H,f)
disp(fitted)
% Just for fun, let's see how well our estimate does VS the true model
truth_error = norm((1*x.^2 + 1*x + 1) - (ym))
model_error = norm((fitted(1)*x.^2 + fitted(2)*x + fitted(3)) - (ym))
% The model has a lower error than the true data generating function.
```

```
clear all; close all; clc;
%==========================================================================
%==========================================================================
%== PROBLEM 7
%==========================================================================
%==========================================================================

%==========================================================================
%== 7(B)
%==========================================================================
% NOTE: I did borrow much of this from MATLBAB's documentation for
%       fmincon but tried to make comments to reflect my
 understanding.
%       https://www.mathworks.com/help/optim/ug/fmincon.html

% The Rosenbrock function in terms of the 2-dim vector x = [x(1),
 x(2)]
fctn = @(x) (1-x(1))^2 + 100*(x(2) - x(1)^2)^2;
% This initial guess took too many iterations so MATLAB gave up
% x0 = [500,1000];
% This initial guess allows the the solver to converge to an answer w/
in
% acceptable number of iterations (since it doesn't quit before
 solving)
x0 = [3,3];
% Use fmincon to solve
soln = fmincon(fctn,x0);
% Display the solution: ends up being x = [x(1)=1, x(2)=1] w/o constr
disp(soln)

% With ineq. and eq. constraints of: [x(1)+2x(2) <= 1], [2x(1)+x(2) =
 1]
A = [1,2];     % Matrix for the inequality constraint
b = 1;         % Vector for the inequality constraint
Aeq = [2,1];   % Matrix for the equality constraint
beq = 1;       % Vector for the equality constraint
soln_2 = fmincon(fctn, x0, A, b, Aeq, beq);
disp(soln_2);


Optimal solution found.

Continuous solution:
         0
    7.7147
         0
         0
    9.2800

LP:                Optimal objective value is 2.317755.
```

```
Heuristics:         Found 1 solution using rounding.

                    Upper bound is 2.460000.

                    Relative gap is 4.05%.


Branch and Bound:

   nodes     total   num int        integer       relative

explored  time (s)  solution           fval        gap (%)

       5      0.00         2   2.430000e+00   0.000000e+00


Optimal solution found.

Intlinprog stopped because the objective value is within a gap
 tolerance of the
optimal value, options.AbsoluteGapTolerance = 0 (the default value).
 The intcon
variables are integer within tolerance, options.IntegerTolerance =
 1e-05 (the
default value).

Integer solution:
     0
     9
     0
     0
     9


Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-
decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint
 tolerance.


fitted =

    1.0396
    0.9724
    1.0049

    1.0396
    0.9724
    1.0049
```

```
truth_error =

    0.9564


model_error =

    0.9543


Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-
decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint
 tolerance.

    1.0000    1.0000


Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-
decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint
 tolerance.

    0.4149    0.1701
```

Published with MATLAB® R2019a