Homework 3
APPM 4600 Numerical Analysis, Fall 2025

Due date: Friday, September 12, before midnight, via Gradescope. **Instructor**: Prof. Becker
Revision date: 9/11/2025

Theme: Calc III review, Hölder's inequality, Lipschitz continuity, global optimization

Instructions Collaboration with your fellow students is OK and in fact recommended, although direct copying is not allowed. The internet is allowed for basic tasks (e.g., looking up definitions on wikipedia) but it is not permissible to search for proofs or to *post* requests for help on forums such as `http://math.stackexchange.com/` or to look at solution manuals. Please write down the names of the students that you worked with. Please also follow our AI policy.

An arbitrary subset of these questions will be graded.

Turn in a PDF (either scanned handwritten work, or typed, or a combination of both) to **Gradescope**, using the link to Gradescope from our Canvas page. Gradescope recommends a few apps for scanning from your phone; see the Gradescope HW submission guide.

We will primarily grade your written work, and computer source code is *not* necessary (and you can use any language you want). You may include it at the end of your homework if you wish (sometimes the graders might look at it, but not always; it will be a bit easier to give partial credit if you include your code). For nicely exporting code to a PDF, see the APPM 4600 HW submission guide FAQ.

Background: Matrix Methods and Calc review We'll write $\boldsymbol{x}$ (not x) to emphasize when $\boldsymbol{x}$ is a vector, and use x_i to denote its i^{th} entry. For $\boldsymbol{x} \in \mathbb{R}^n$, we define the family of ℓ_p norms $\|\boldsymbol{x}\|_p = \left(\sum_{i=1}^n |x_i|^p\right)^{1/p}$ for $1 \le p < \infty$ and $\|\boldsymbol{x}\|_\infty = \max_{1\le i\le n} |x_i|$. The Euclidean norm is the case $p = 2$. For vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, in matrix methods you learned the Cauchy-Schwarz inequality: $|\boldsymbol{a}^\top \boldsymbol{b}| \le \|\boldsymbol{a}\|_2\|\boldsymbol{b}\|_2$. In fact, this is a special case of the Hölder inequality, which says $|\boldsymbol{a}^\top \boldsymbol{b}| \le \|\boldsymbol{a}\|_p\|\boldsymbol{b}\|_q$ for any $1 \le p \le \infty$ with q defined by solving $\frac{1}{p} + \frac{1}{q} = 1$, with the convention that $\frac{1}{\infty} = 0$. In particular, $|\boldsymbol{a}^\top \boldsymbol{b}| \le \|\boldsymbol{a}\|_1\|\boldsymbol{b}\|_\infty$.

We say that a function $f : \mathbb{R}^n \to \mathbb{R}$ is Lipschitz continuous (with respect to the p-norm) if there is a constant $L \in \mathbb{R}$ such that

$$(\forall \boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n) \quad |f(\boldsymbol{a}) - f(\boldsymbol{b})| \le L\|\boldsymbol{a} - \boldsymbol{b}\|_p \tag{1}$$

and we refer to any such constant L as a "Lipschitz constant." (If L is a Lipschitz constant, then clearly any $\tilde{L} > L$ is also a Lipschitz constant; we usually look for the smallest constant we can find).

We also have multivariate Taylor series, and in particular, we have the remainder form, below specialized to the 1-term Taylor series:

$$f(\boldsymbol{b}) = \underbrace{f(\boldsymbol{a})}_{\text{1 term Taylor series}} + \underbrace{\nabla f(\boldsymbol{\xi})^\top(\boldsymbol{b} - \boldsymbol{a})}_{\text{remainder}} \tag{2}$$

for some $\boldsymbol{\xi}$ on the line segment between $\boldsymbol{a}$ and $\boldsymbol{b}$.

Problem 1: Brute force optimization. "Bruce force" optimization of a function $f : \mathbb{R}^n \to \mathbb{R}$ means evaluating f on a large dataset of points $\{\boldsymbol{x}_k\}_{k=1}^n$ for a large n (usually points from a grid), and then just choosing the best value of $f(\boldsymbol{x}_k)$. For this problem, we'll always use a dataset of equally spaced grid points. For a domain $\Omega \subset \mathbb{R}^n$, denote $f^\star = \min_{\boldsymbol{x}\in\Omega} f(\boldsymbol{x})$. Finding the *exact* solution is asking for too much, so instead we'll look for an ϵ-solution, meaning a feasible point $\boldsymbol{x} \in \Omega$ such that $f(\boldsymbol{x}) \le f^\star + \epsilon$.

a) Let $f : \mathbb{R} \to \mathbb{R}$ be a univariate function, and suppose it is L Lipschitz continuous (with respect to the absolute value). Let the domain be $\Omega = [0, R]$, and choose n equally spaced grid points (including the end points). In terms of L and R, how large must n be in order to guarantee that the brute force method returns an ϵ-solution?

b) Now let $f : \mathbb{R}^d \to \mathbb{R}$ be a multivariate function of dimension $d \geq 1$, and suppose it is L Lipschitz continuous (with respect to the ℓ_∞) norm. Let the domain be $\Omega = [0, R]^d$. Choose a uniform grid on Ω with m equally spaced points in each dimension (again, including end points). In terms of L, R and d, how large must n be in order to guarantee that the brute force method returns an ϵ-solution?

Problem 2: Lipschitz continuity of the Michalewicz function in 1D and 2D. The Michalewicz function in dimension d is defined as

$$f(\boldsymbol{x}) = -\sum_{i=1}^{d} \sin(x_i) \sin^{20}\left(\frac{i x_i^2}{\pi}\right).$$

For dimensions $d = 1$ and $d = 2$, prove that f is Lipschitz continuous (with respect to the ℓ_∞ norm) on the domain $x_i \in [0, \pi]$ for $i = 1, \ldots, d$. *Hints: use Eq. (2) and Hölder's inequality, so focus on computing ∇f and then bounding it in the appropriate norm. Any finite bound works and will get you full credit, though of course smaller bounds are better. It's not much harder to give a bound that works for all $d \geq 1$, though you don't have to do this.*

Problem 3: Brute force optimization of the Michalewicz function

a) Using your results from Problem 1 and Problem 2, if we want to find an ϵ-solution to the Michalewicz function over the domain $\Omega = [0, \pi]^d$, how many function evaluations n do we need? Specifically, for $d = 1, 2, 3, 4$ and 5, write down the actual number n needed for both $\epsilon = 0.1$ and $\epsilon = 10^{-4}$. *Since you didn't have to work out the Lipschitz constants for $d > 2$, you can just suppose they are all $L = 100$ for simplicity, and use that in your calculation.*

b) **Programming.** Using the programming language of your choice, run the brute-force optimization on the 2D Michalewicz function and return an approximation of $f^\star$ that is accurate to at least $\epsilon = 10^{-1}$. Turn in the relevant parts of your code. *Hint: in Python, use `np.linspace` and `np.meshgrid` to help make the grid; in Matlab, use `linspace` and `meshgrid`.*

c) **Optional — not graded.** According to the web, for $d = 2$, $f^\star = -1.8013$ (at $(2.20, 1.57)$). Can you confirm this with your code? i.e., can you find an ϵ-solution for $\epsilon = 10^{-4}$?

For $d = 10$, $f^\star = -9.66015$. If you wanted to confirm this via brute force, i.e., using $\epsilon = 10^{-4}$, how many function evaluations would you need? If each function evaluation took 1 ms, how long would this take?

For reference, the essential snippets of Python code used to make the 2D plots in Fig. 1 are below:

```
import numpy as np
from matplotlib import pyplot as plt
def Michalewicz_2D(x, y):
    return -np.sin(x)*np.sin(x**2/np.pi)**(20) -np.sin(y)*np.sin(2*y**2/np.pi)**(20)
m      = 100  # grid points *per* dimension
grid1D = np.linspace(0,np.pi, m)
x,y    = np.meshgrid(grid1D,grid1D)
n      = x.size # total grid points overall
print(f'Size of grid is {n=}')
f = Michalewicz_2D(x,y) # evaluate the function
fig, ax = plt.subplots(subplot_kw={projection: 3d})
surf = ax.plot_surface(x,y,f, cmap='coolwarm')
```

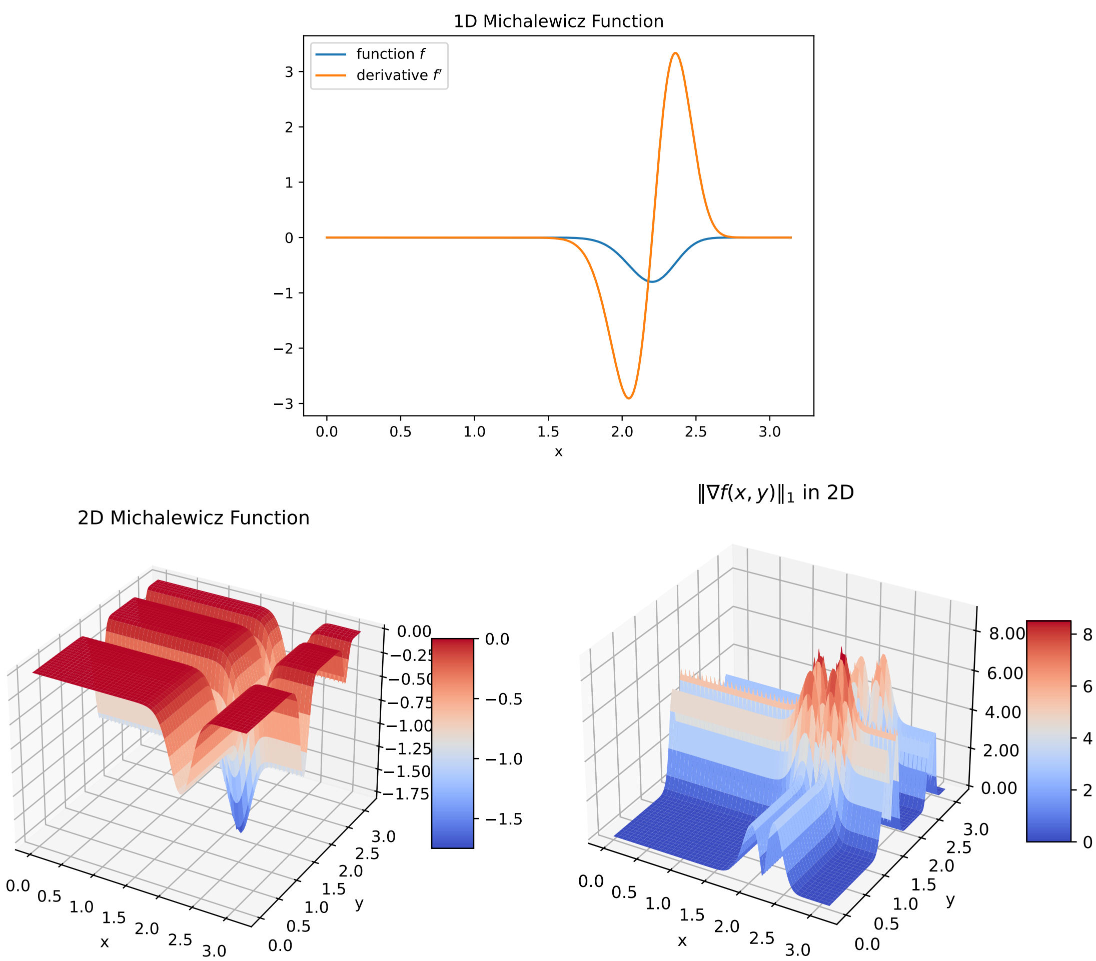

Figure 1: Plots of the Michalewicz function (and its derivative, or ℓ_1 norm of its gradient) in 1D and 2D. Partial snippets of code to the make the plot are at the end of the homework.