**Chad Callender**
**MATH 3533**
**Honors Project: May 5, 2021**

---

# 1 Introduction

A simple Google search of "Sikes Lake land area" yields few results pertaining to the dimensions of the lake. Sikes Lake is a local place of interest in the city of Wichita Falls, TX (WF). Maintained by Midwestern University (or MSU Texas), Sikes Lake serves as the backdrop for many of the university's oldest traditions. Additionally, many patrons take advantage of the 1.1-mile track along most of the lake's perimeter for exercise or leisurely walks. As one of the more popular landmarks of the WF area, there should be more readily available information about Sikes Lake. In this project, we will calculate the surface area of Sikes Lake using Gaussian Quadrature. Furthermore, this project includes an error analysis of the approximations used in calculation.

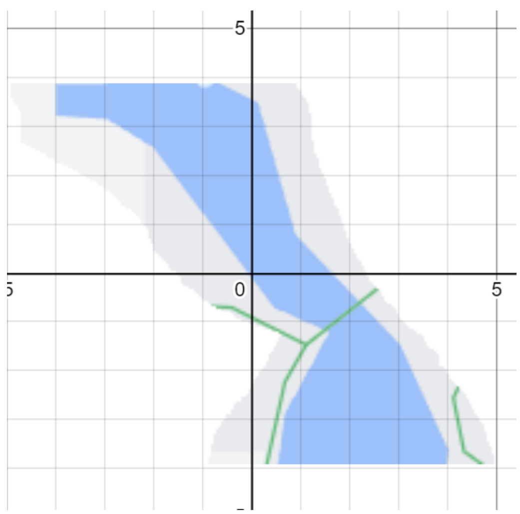
# 2 Method

Suppose we want to find the surface area (SA) of a region $R$ bounded by the curves $y = c(x)$, $y = d(x)$ and the lines $x = a$, $x = b$. The SA is given by the double integral

$$\iint\limits_R 1\, dA = \int_a^b \int_{c(x)}^{d(x)} 1\, dy\, dx.$$

Here, the region $R$ is Sikes Lake shown in the map (from Google Maps) below.


To approximate the curves $y = c(x)$ and $y = d(x)$, we use Lagrange Interpolating polynomials, $P_n(x)$, with points taken from the edge of the lake. Now, to get points on the edge of the lake, we use Desmos, a graphing software. **First**, we take a snip of the screen using a snipping tool. **Second**, we save the snip as an image and upload it to Desmos, producing the result below.




**Third**, we pick points $\{(x_0, y_0), (x_1, y_1), ..., (x_n, y_n)\}$ on the edge of the lake by zooming in on the grid.

Now, we can use these points to construct a table using Newton's divided differences to find $P_k(x)$ of the form

$$P_n(x) = a_0 + \sum_{i=1}^{n} f\left[x_0, x_1, ..., x_i\right] \prod_{j=0}^{i-1} (x - x_j)$$

where $f[x_i] = y_i$ and $f[x_0, ..., x_n] = \frac{f[x_1,...,x_n] - f[x_0,...,x_{n-1}]}{x_n - x_0}$. We calculate the numbers $f[x_0, ..., x_n]$ with the Python code below.

main.py

```python
import numpy as np

def divDiff(x, y, x0 = None):
    n = len(x)
    F = np.zeros((n,n))
    for i in range(n):
        F[i,0] = y[i]

    for i in range(1,n):
        for j in range(1,i+1):
            F[i,j] = (F[i][j-1] - F[i-1][j-1]) / (x[i] - x[i-j])
    print(F)
    return

x = [.1, .87, 3.0432, 4.001]
y = [3.5, .84, -1.4595, -3.61]
divDiff(x, y)
```

Console Shell

```
[[ 3.5         0.          0.          0.        ]
 [ 0.84       -3.45454545  0.          0.        ]
 [-1.4595     -1.05811706  0.81422547  0.        ]
 [-3.61       -2.24524953 -0.37915441 -0.3059164 ]]
>
```

Here, $f[x_0]$ is the first number in the first column, $f[x_0, x_1]$ is the second number in the second column, $f[x_0, x_1, x_2]$ is the third number in the third column, and so on. In general, $f[x_0, ..., x_n]$ is the $a_{n+1,n+1}$ entry of the matrix in the code snippet.

To approximate the SA, we use Gaussian Quadrature with $n = m = 5$ given by

$$\int_a^b \int_{c(x)}^{d(x)} f(x, y)\, dy\, dx$$
$$\approx \int_a^b \frac{d(x) - c(x)}{2} \sum_{j=1}^{n} c_{n,j} f\left(x, \frac{(d(x) - c(x))r_{n,j} + d(x) + c(x)}{2}\right) dx,$$

where $c_{n,j}$ and $r_{n,j}$ are taken from the table below.

| $n$ | Roots $r_{n,i}$ | Coefficients $c_{n,i}$ |
|---|---|---|
| 2 | 0.5773502692 | 1.0000000000 |
| | −0.5773502692 | 1.0000000000 |
| 3 | 0.7745966692 | 0.5555555556 |
| | 0.0000000000 | 0.8888888889 |
| | −0.7745966692 | 0.5555555556 |
| 4 | 0.8611363116 | 0.3478548451 |
| | 0.3399810436 | 0.6521451549 |
| | −0.3399810436 | 0.6521451549 |
| | −0.8611363116 | 0.3478548451 |
| 5 | 0.9061798459 | 0.2369268850 |
| | 0.5384693101 | 0.4786286705 |
| | 0.0000000000 | 0.5688888889 |
| | −0.5384693101 | 0.4786286705 |
| | −0.9061798459 | 0.2369268850 |

Since we are approximating SA, $f(x, y) = 1$, so the approximation becomes.

$$A = \int_a^b \int_{c(x)}^{d(x)} 1\, dy\, dx \approx \int_a^b \frac{d(x) - c(x)}{2} \sum_{i=1}^{n} c_{n,j}\, dx.$$

This formula gives us the SA of the lake from the map. We calculate the vertical and horizontal scales, say $v$ and $h$, using Google Map's measure distance function and dividing that result by the corresponding distance on the graph in Desmos. Finally, we calculate the product of the vertical scale, the horizontal scale, and the result of the double integral to get the surface area of the physical lake (in square feet). That is,

$$\text{real area } = v \cdot h \cdot A.$$

Here, $v = 88.48235$ and $h = 94.82367$. The Python code below produces the results of Gaussian Quadrature.

main.py

```python
import math
# Source: Numerical Analysis, 10th Ed. by Burden,
# Faires, Burden.

# roots and coefficients of Legendre Polynomials
# source: p. 232, table 4.12
r = [(0.5773502692, -0.5773502692), (0.7745966692, 0, -0.7745966692), (0.8611363116, 0.3399810436, -0.3399810436, -0.861136311), (0.9061798459,0.5384693101, 0, -0.5384693101, -0.9061798459)]

co = [(1, 1), (0.5555555556, 0.8888888889, 0.5555555556), (0.3478548451, 0.6521451549, 0.6521451549, 0.3478548451), (0.2369268850, 0.4786286705, 0.5688888889, 0.4786286705, 0.2369268850)]

def f(x,y):
    return 1

def c(x):
    return 3.244 - 0.33147491*(x+4.008) - 0.22086254*(x+4.008)*(x+2.035) + 0.08194589*(x+4.008)*(x+2.035)*(x-.461)

```

Console | Shell

```
area: 56781.755783987515
>
```

main.py

```python
def d(x):
    return 3.5-3.45454545*(x-.1)+0.81422547*(x-.1)*(x-.87)-0.3059164*(x-.1)*(x-.87)*(x-3.0432)

def gauss_q(a,b,m,n):
    # source: p. 244, algorithm 4.5
    h1 = (b - a) / 2
    h2 = (b + a) / 2
    J = 0

    for i in range(1,m+1):
        JX = 0
        x = h1*r[m-2][i-1] + h2
        d1 = d(x)
        c1 = c(x)
        k1 = (d1 - c1) / 2
        k2 = (d1 + c1) / 2

        for j in range (1,n+1):
            y = k1*r[n-2][j-1] + k2
            Q = f(x,y)
            JX += co[n-2][j-1] * Q

        J += co[m-2][i-1] * k1 * JX
```

Console | Shell

```
area: 56781.755783987515
>
```

main.py

```
        J += co[m-2][i-1] * k1 * JX

    J = h1 * J
    print(J)

    return J

gauss_q(-.8026, 3.49, 5, 5)

area1 = gauss_q(-.8026, 3.49, 5, 5) * 94.82367 * 88.48235
# 94.82367 is the horizontal scale factor; 88.48235 is the vertical scale factor

print("area:", area1)
```

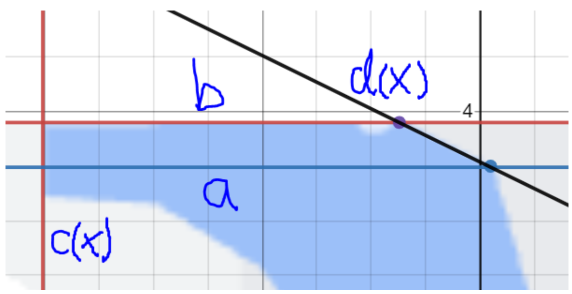
Console Shell

```
6.767611332282333
6.767611332282333
area: 56781.755783987515
```

Overall, we will split the map of the lake into small sections and add all the SAs to get the total SA of Sikes Lake.

# 3 Results

## 3.1 Area 1

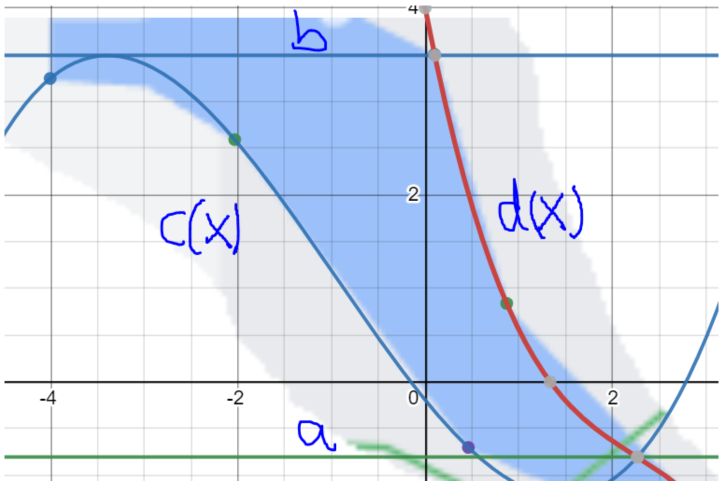



$a = 3.49$, $b = 3.9$, $c(x) = -4.02$,

$d(x) = 3.9\left(\frac{x-.1021}{-.74-.1021}\right) + 3.49\left(\frac{x+.74}{.1021+.74}\right)$ interpolates at the points $(.1, 3.5), (-.74, 3.9)$.

$A_1 = 47700.49398241914$ sq. ft.

### 3.2 Area 2




$a = -.8026$, $b = 3.49$,

Points:

$$(-4.008, 3.244), (-2.035, 2.59), (.461, -.701), (1.564, -1.2195)$$

$c(x) = 3.244 \; - 0.33147491\,(x + 4.008) - 0.22086254\,(x + 4.008)\,(x + 2.035)$
$+0.08194589\,(x + 4.008)\,(x + 2.035)\,(x - .461)$,

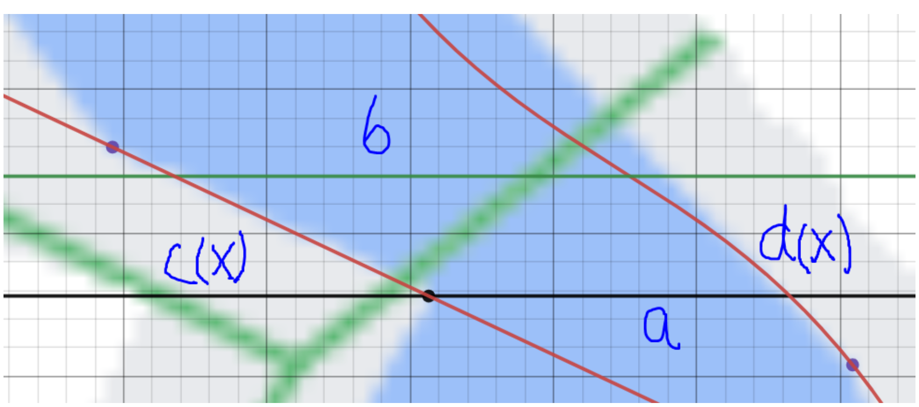
Points:

$$(.1, 3.5), (.87, .84), (3.0432, -1.4595), (4.001, -3.61),$$

$d(x) = 3.5 \; - 3.45454545\,(x - .1) + \; 0.81422547\,(x - .1)\,(x - .87)$
$-0.3059164\,(x - .1)\,(x - .87)\,(x - 3.0432)$

$A_2 = 56781.755783987515$ sq. ft.

### 3.3 Area 3




$a = -1.2195$, $b = -.8026$,

Points:

$$(.461, -.701), (1.564, -1.2195)$$

$c(x) = \ -0.701 - 0.4700816\,(x - .461),$

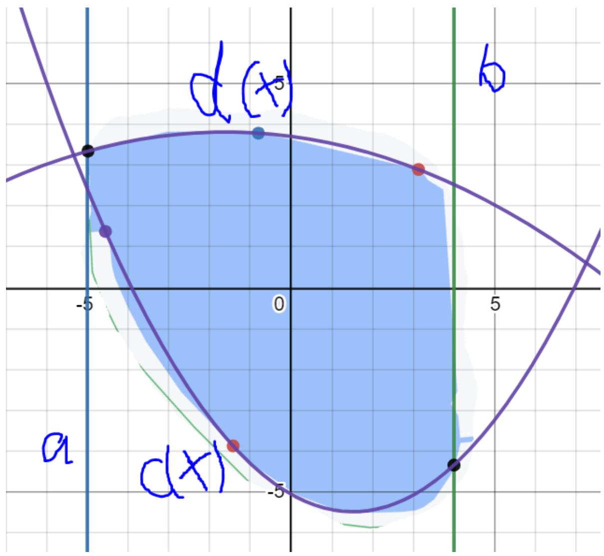
Points:

$$(.1, 3.5), (.87, .84), (3.0432, -1.4595), (4.001, -3.61)$$

$d(x) = 3.5 \ - 3.45454545\,(x - .1) + \ 0.81422547\,(x - .1)\,(x - .87)$
$-0.3059164\,(x - .1)\,(x - .87)\,(x - 3.0432)$

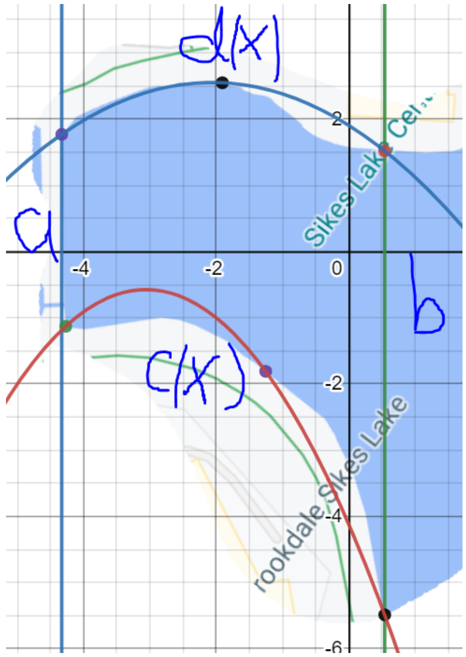
$A_3 = 40869.12476470251$ sq. ft.

### 3.4 Area 4



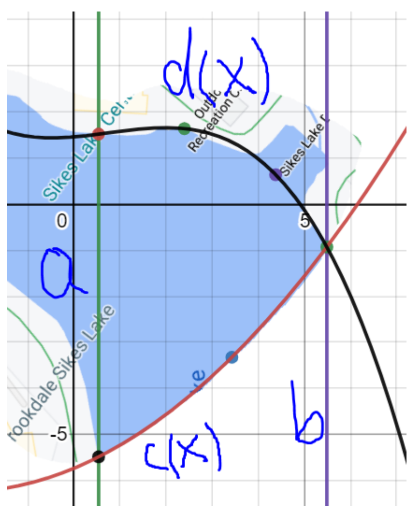

$A_4 = 90229.52926461812$ sq. ft.

## 3.5 Area 5



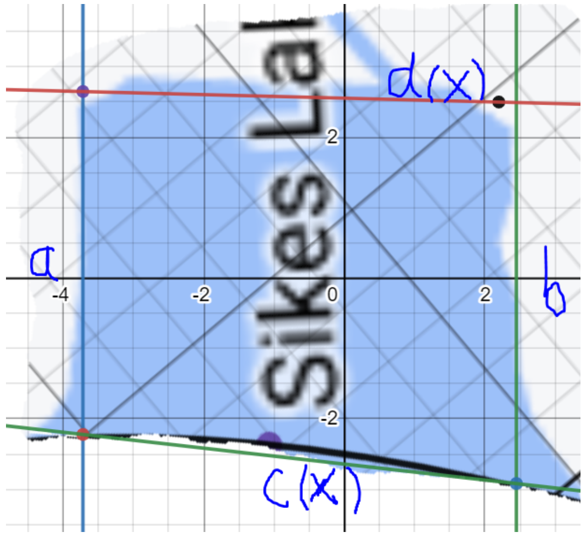

$A_5 = 152776.27493172447$ sq. ft.

## 3.6 Area 6



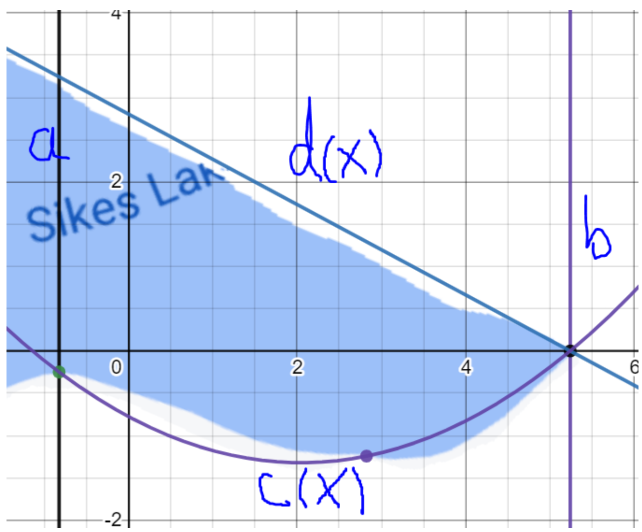

$A_6 = 177069.75725968968$ sq. ft.

### 3.7 Area 7



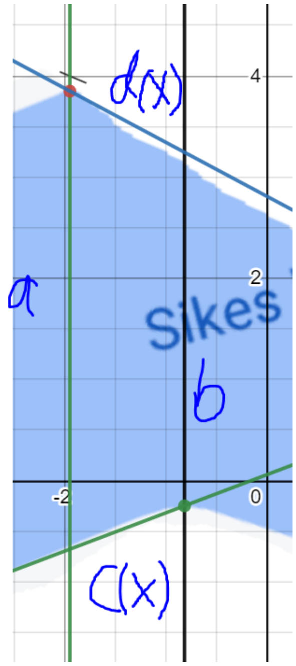

$A_7 = 8997.527072092824$ sq. ft.

### 3.8 Area 8



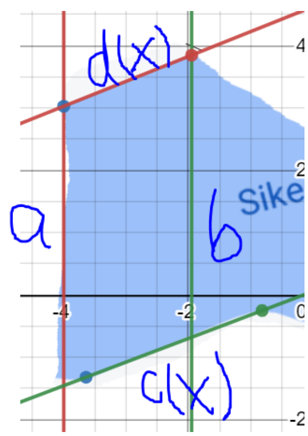

$A_8 = 86430.90807922228$ sq. ft.

### 3.9 Area 9




$A_9 = 25416.189244795183$ sq. ft.

### 3.10 Area 10

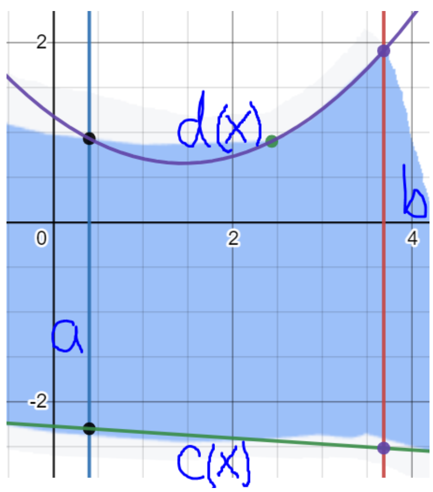



$A_{10} = 51803.07614934796$ sq. ft.

### 3.11 Area 11

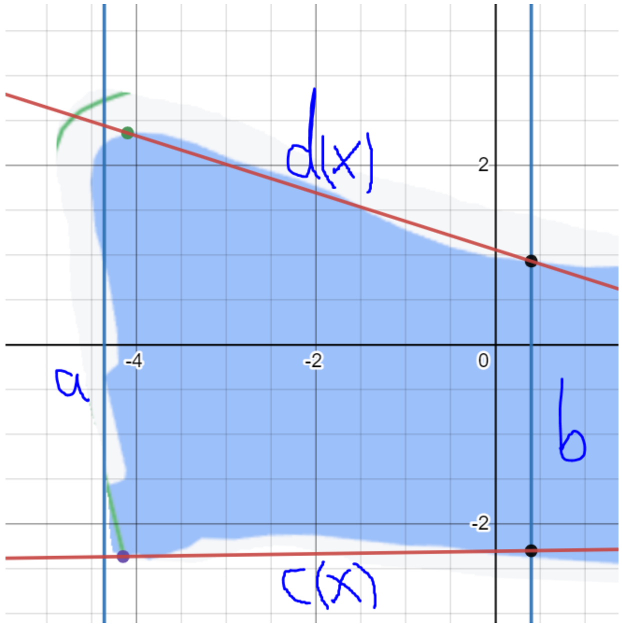

$A_{11} = 13673.755090010338$ sq. ft.

### 3.12 Area 12




$A_{12} = 57753.97104756622$ sq. ft.

### 3.13 Area 13




$A_{13} = 99611.17469695958$ sq. ft.

### 3.14 Total SA

$SA = 909113.5374$ sq. ft.

## 4 Error Analysis

The Wichitan, the official news outlet for MSU Texas, cites that Sikes Lake has a surface area of 21.5 acres which is approximately 936540 sq. ft. We calculated the surface area of the lake to be 909113.5374 sq. ft. This figure represents an absolute error of 27426.46263 sq. ft. and a relative error of 0.02928488119.

## 5 References

Burden, R. L., Faires, D. J., & Burden, A. M. (2015).*Numerical Analysis* (10th ed.). Cengage Learning.
Clancy, J. (2018, Nov. 25). *Sikes Lake Needs Dredging.* The Wichitan. https://thewichitan.com/57533/news/sikes-lake-is-in-need-of-dredging/.