Engineering Calculus 1
Mini Project 3: Advanced use of MS Excel (and similar tools)

April 2023

This project will introduce you to algorithm complexity, not just as a computer science topic but as a measure of how difficult it can be to follow the steps of a particular solution method. We will simulate budgeting constraints and work on finding a "good enough" solution to problems when a perfect solution is just out of reach.

You have a week to complete this project. Do not wait until the last day. The sooner you start, the more opportunities you will have to ask for help if you need it. Reach out if any of the instructions are not clear or you would like any feedback.

Before We Begin...

This time most of the work can be done by hand, and where appropriate we will use MS Excel (or equivalent). Though you may have only used it before to view spreadsheets and maybe compute values using formulas, there are more advanced applications like curve fitting. Now, there are more robust tools for this purpose, but we're focusing on using what is "good enough" and for the proposed data set this will do.

It will be up to you how much you want to do by hand versus finding ways to have Excel to do the work for you. For example, Excel curve fitting methods may output coefficients only, so you have to translate those into functions before continuing to work with them. While there are ways to compute derivatives with Excel, you may find that it is easier to do these by hand (given we will be working mostly with polynomials). The decision-making process you follow here is part of the exercise.

- Look at some of the tools for interpolation and curve-fitting. If you are graphing data you can use trendlines (which offers the option to not just find the line of best fit but a variety of functions). Without graphing, you can use `linest`.
- Think about how you would divide your data into pieces to fit curves piecewise. Conveniently, the number of data points provided is highly divisible (360) so many different numbers of groups are possible. Many solutions are possible here, just think of a convenient way to tag all the data points in a particular group so you can then interpolate over one group at a time.

When you present the work **make sure to include the exact formulas and settings you used. Your results should be reproducible.**

1 Comparing Algorithmic Complexity

Have you ever thought about how many steps it takes to complete a task? While this concept is very important in computer science, being able to break down projects into their main components and estimating the work that goes into each of them is an important part of project management. We will introduce the topic algorithmically by comparing ways to order lists.

Suppose you have a list of n elements.

$$[a_1, a_2, \ldots, a_n]$$

These could be names, phone numbers, any kind of data that can be ordered in a sensible way (for example using alphabetical order or numerical order).

$$[\text{Tran}, \text{Carlos}, \text{Mai}, \ldots, \text{Rahul}]$$

$$[14323, 9342, 64324, \ldots, 7805]$$

This list may be out of order. How would you sort it and how long would it take to do?

One method is to search through all n elements of the list, find the first, and set it aside. Since we have to look at n objects we can think of this as taking n steps. After finding the first object, we are left with a list of $n-1$ elements to sort. We can repeat this process and add up the number of steps it took:

$$\text{number of steps} = n + (n-1) + (n-2) + \ldots + 1$$

Now, to save on the time it takes to write this out we can use sigma notation:

$$\text{number of steps} = \sum_{i=1}^{n} i$$

1. Compute this sum for several values of n and write down your results below.

   - $\sum_{i=1}^{1} i =$
   - $\sum_{i=1}^{2} i =$
   - $\sum_{i=1}^{3} i =$
   - $\sum_{i=1}^{4} i =$
   - $\sum_{i=1}^{5} i =$

   To find a pattern, we can multiply all terms by 2.

   - $2\sum_{i=1}^{1} i =$
   - $2\sum_{i=1}^{2} i =$
   - $2\sum_{i=1}^{3} i =$
   - $2\sum_{i=1}^{4} i =$

- $2\sum_{i=1}^{5} i =$

2. How do the values depend on the upper limits of the sums? (i.e. how does $2\sum_{i=1}^{3} i$ depend on 3?) Now let's generalize to the nth sum:

   $2\sum_{i=1}^{n} i =$

   With this we can solve for the sum we wanted initially and get a function that describes the complexity in terms of the size of the input: $f(n) = \sum_{i=1}^{n} i =$

   This expression is the algorithmic complexity of this particular sorting method. Now, as n grows increasingly large we will not care about all of the terms in the expression, only the "fastest growing" ones. This is best represented with asymptotic notation.

   Find a function $g(n) = n^k$ for some positive integer k, and two constants c_1 and c_2 so that for $n \geq 2$ we have

   $$c_1 g(n) \leq f(n) \leq c_2 g(n)$$

   $g(n) =$

   $c_1 =$

   $c_2 =$

   You can try graphing tools and changing the variable names (using x instead of n to get an idea).

   The fact that $g(n)$ bounds $f(n)$ is denoted by $f(n) = \Theta(g(n))$[1]. We call this an *asymptotically tight bound* for $f(n)$. You may find, where these bounds are used, that instead of Θ authors will write $O(g(n))$. This is *big O notation* and technically is used to indicate a function $g(n)$ such that for some constant c and starting at some large enough value of n (we chose $n = 2$ above) we have $f(n) \leq cg(n)$.

   Technically this means that $4n + 2$ is $O(n^2)$, as linear functions grow more slowly than quadratic functions. It can more accurately be bound by other linear functions so that $4n + 2 = \Theta(n)$. However, you will find authors writing $4n + 2$ is $O(n)$ to indicate that $4n + 2$ grows roughly as a linear function (that is, they use $O(n)$ to mean $\Theta(n)$). Following in this tradition of bad notation, we can say that the algorithmic complexity of our sorting algorithm above, $f(n) = O(\qquad)$.

   Suppose there was another algorithm to sort lists of n elements, and that its complexity was $h(n) = O(n \log(n))$. We would then like to decide which of the two algorithms to use for very large lists.

   To compare the growth of two functions F and G, we compute

   $$\lim_{x\to\infty} \frac{F(x)}{G(x)}$$

   If this limit is infinite, it will indicate that $F(X)$ grows faster than $G(x)$ (and as x gets large takes longer to sort). If the limit is 0, it will indicate that $G(x)$ grows faster than $F(x)$.

3. Using the bounding function $g(n)$ you found above, compute the limits:

[1] This is the capital Greek letter theta.

(a) $\lim\limits_{n \to \infty} \dfrac{g(n)}{n \log(n)}$

(b) $\lim\limits_{n \to \infty} \dfrac{n \log(n)}{g(n)}$

Which of the two algorithms is faster? Explain.

If there are three algorithm, A, B, and C, with complexities $O(n^3)$, $O(n \log(n))$, and $O(2^n)$, respectively. Sort them by run time.

2 Signals

When observing phenomena that have a “switch”-like behavior (think of things that only activate when a measurement exceeds a threshold value) we often model them with so-called *indicator functions*. One such function is the Heaviside step function:

$$H(x) = \begin{cases} 1 & x > 0 \\ 0 & x \le 0 \end{cases}$$

For example, if a switch is on when $H(x) = 1$ and when it is on we can observe changes in outputs following $f(x) = \sin(x)$ then we can model the overall behavior with $H(x)f(x)$ where the output is 0 while the switch is off but follows $f(x)$ when the switch is on.

Inconveniently, step functions are not continuous and therefore not differentiable. So to work with them we often approximate them with continuous and differentiable functions when we want to incorporate them into calculations.

Let’s consider a few different candidates for this purpose.

7. Show that all of the following functions asymptotically approximate $H(x)$. That is, show that their end behavior is the same as that of $H(x)$.

(a) $\dfrac{e^x}{1+e^x}$

(b) $\dfrac{1}{\pi}\arctan(x)+\dfrac{1}{2}$

(c) $\dfrac{1}{2}\tanh(x)+\dfrac{1}{2}$ where $\tanh(x)=\dfrac{\sinh(x)}{\cosh(x)}$, $\sinh(x)=\dfrac{e^x-e^{-x}}{2}$, $\cosh(x)=\dfrac{e^x+e^{-x}}{2}$.

8. Compute $\lim_{n\to\infty} \sqrt[2n+1]{n}$.

Use the fact that $\sqrt[2n+1]{-x} = -\sqrt[2n+1]{x}$ for any x to compute $\lim_{n\to-\infty} \sqrt[2n+1]{n}$

Given this asymptotic behavior, we could use

$$g(x) = \frac{1}{2}x^{\frac{1}{2n+1}} + \frac{1}{2}$$

with a large value of n to approximate the Heaviside function. Unlike the previous alternatives, this has a disadvantage. Can you figure out what it is?

9. Since one of the nice features of the Heaviside function is its abrupt change, we'd like to approximate it with a function that is as steep as possible when making the transition between 0 and 1. Which of the three functions initially considered is steepest when $x = 0$? Show all your work.

10. How could you make the approximation better? I.e. Starting with the function with the steepest slope, how could you make it even steeper at $x = 0$ while preserving the end behavior?

3 Noisy Signals

In math courses we deal with functions that are continuous and exactly fit where we want them to. Real-life data rarely shows this behavior. For starters, many measurements are taken at discrete time intervals: that is, we know where a particle is 1min, 2min, 4min after starting an experiment but not necessarily at every time point in between. Moreover, it is experimentally very difficult to completely isolate and control all the possible contributions to what we are measuring. This results in what we call *noise*. Even for behaviors we can very accurately predict with formulas we can expect that some experimental observations will have some degree of error. You can still see trends and patterns, but they are hidden by data values that don't fit perfectly.

It's good to get acquainted with two ways of handling this problem: interpolation and curve fitting.

Interpolation, in mathematics, roughly refers to "connecting the dots." We may not have exact data about what happens in between measurements but if we can find a function that fits through the data we do have we can predict what happens elsewhere. For example, you may be familiar with exercises where you have to find the next number in a sequence: 2, 5, 8, ______. Whether you realize it or not, you are probably finding a function $f(n)$ that fits the first three points and then computing $f(4)$. We can make this more difficult by asking to fill a gap in a sequence: 4,9, ______, 25, ... In a bigger scale instead of assuming our domain consists of natural numbers (as is the case with sequences) we connect the existing data points with a function that contains those particular values. Various interpolation methods exist and are beyond the scope of this course, all we need to know here is that they exist, they help fill in blanks in our data, and they are forced to include existing data points.

When data is noisy, however, we may not want to contain the points exactly and instead only find a function that is as close as possible to our experimental values. This is known as curve-fitting. The fine art of finding what type of function will best fit a particular data set is again beyond the scope of the course, so we will rely on the real analysis result that says we can approximate almost anything "nice" with polynomials. The math behind minimizing the distances between a function and the data is something we will leave to our chosen tool, Excel (or any equivalent).

To ground the math in a real-life application we will consider action potentials. Action potentials are sequences of voltage changes across a cell membrane that occur in response to stimuli. For example, these changes are how neurons receive and transmit information from the outside world. When a doctor tests your patellar reflex by knocking your knee, a strong enough stimulus will cause your leg to jerk. The reason your knee doesn't jerk whenever you gently place your hand on it is that the sequence of steps that results in the activation of your muscles doesn't start until the stimulus exceeds some threshold value (see the Heaviside function uses above). In general, action potentials have this shape:

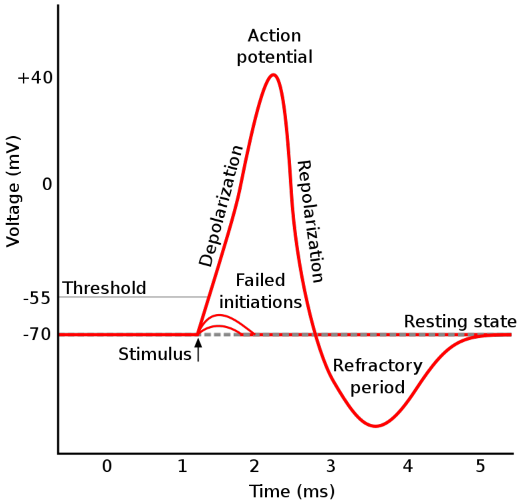

Figure 1: Original by en:User:Chris 73, updated by en:User:Diberri, converted to SVG by tiZom, CC BY-SA 3.0 http://creativecommons.org/licenses/by-sa/3.0/, via Wikimedia Commons

A series of these is what you see in electrocardiograms. You may notice they rarely look this smooth. In fact, depending on a lot of factors the shape we get from actual experimental data is unlikely to be this smooth, and sampling more doesn't really solve the problem. **You can download a sample signal file in `.csv` format from Canvas.**

As you can see in Figure 2, interpolation is not a good way to find where the extreme values take place.

We will take for granted that we can fit the data with cubic functions (i.e. polynomials of degree 3). However, the accuracy of the curve fitting will depend on how much of the data we consider at a time (all? half? a third?), and each time we curve fit we will incur a computational cost. It would not be ideal to split the data into a lot of pieces to curve fit both because of the cost and because we will end up fitting functions to the noise instead of smoothing it out.

The goal then, is to find a piecewise cubic curve that approximates our data well enough to do calculus on (so that we can then find more or less precise values for the minimum and maximum).

While we can make good estimates by looking at the graph, remember that if this were part of a large data set you would want to let the computer handle the process and the computer doesn't see like we do. What we have to do is find a good set of steps the computer can follow to find these maximum and minimum values.

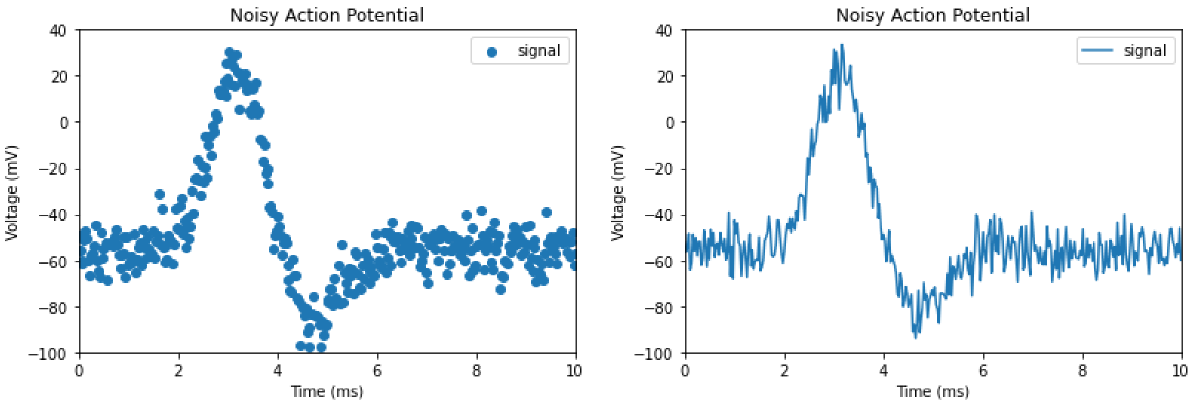

Figure 2: Scatter plot with data points and a bad interpolation.

The costs assigned here are somewhat arbitrary but meant to simulate an actual budget in a project. You have a budget of $100.

While the cost table doesn't explicitly tell you what to do with derivatives, evaluated values, etc. you are free to use any of the techniques discussed in class so far.

Tool	**Cost ($)**
Curve fitting	5
Differentiating	1
Solving an equation	3
Evaluating a function	2

If you split your data into three pieces, for example, then you will have to curve-fit 3 times, which will cost $3 \cdot 5 = 15$ points. It would be cheapest to only curve fit once and do optimization on whatever Excel outputs but it is likely to be a bad estimate, and you will be under budget. Normally a good thing, but consider that if you budget a project at $100 and only spend $15, your future projects will probably not get 100% funded. It will be assumed that you can always make do with less than what is given to you. You want to use as much of your budget as you can without going over.

11. The first thing you will need is some way to divide your data (from the `.csv` file on Canvas) into chunks. Remember that the computer doesn't see what you see, so unless you can find a formula that would work with all similar problems you should avoid grouping visually. Paste a screenshot with your proposed solution

to the data grouping problem and explain your reasoning. Make sure to include any relevant formulas.

12. Once you have a group, you should find a way to fit a curve through the data. Choose a method and paste your proposed solution below, with an example of how you would interpret the output.

13. Describe the steps you would follow to optimize the polynomial within its restricted domain.

14. Would you optimize over every interval? Explain your reasoning.

15. Write out the steps you would follow to find the minimum and maximum of the data set. Combine all the steps above and make sure you are under budget.

16. Paste a graph of your piecewise polynomial curve and discuss how good an approximation it is. How confi-

dent are you in the values you obtained for the maximum and minimum?

17. How would your answer change if your budget was \$50? \$200? If you were proposing this project, how much budget would you ask for? Explain.

Challenge

Find a continuous function that approximates the data as closely as possible. It may not necessarily be polynomial. You can use graphing tools like GeoGebra or Desmos. Make sure to include the formula for your function and explain how you created it.

Mini Project 3: Advanced Use of MS Excel
Rubric

	1	2	3	4	5
Technology use	All answers are missing or incorrectly use the tools	A majority of answers are missing or incorrectly use the tools	Some answers are missing or incorrectly use the tools	Most answers correctly use the tools	All answers correctly use the tools
Calculus concepts	The work shows serious misunderstandings when using calculus concepts	The work shows some misunderstandings when using calculus concepts	The explanations given are generally correct but incomplete	Most explanations are correct and complete	All explanations are correct and complete
Math work	Many solutions are incorrect or incomplete	A few solutions are incorrect or incomplete	Some solutions are missing steps or have small errors	A few solutions are missing steps or have small errors	All solutions are correct and complete
Analysis	All answers are left blank	A majority of answers are left blank or show very shallow analysis	About half of the answers are left blank or show very shallow analysis	Most questions are answered in depth	All spaces are answered in depth
Clarity	It is hard to read/follow the work	Some of the work is hard to read/follow	The organization/tidiness leaves room for improvement but is readable	The work is generally easy to read/follow	It is very easy to read/follow the work done
Challenge	Not attempted	Attempted but the graph doesn't approximate the data or there is no explanation.	The graph approximates the data set but the explanation is not clear.	The graph approximates the data set and the explanation has some error.	The graph approximates the data and the explanation is adequate.