CS370 Fall 2023: Assignment 1

Instructors: Justin Wan (email: justin.wan@uwaterloo.ca)
Leili Rafiee Sevyeri (email: leili.rafiee.sevyeri@uwaterloo.ca)

Due October 4, Wednesday, at 10:00pm EDT

Submit all components of your solutions (written/analytical work, code/scripts, figures, plots, output, etc.) to Crowdmark in PDF form for each question (multiple pages are allowed). Note that you may resubmit as often as necessary until the due date - only the final submission will be marked.

You must also separately submit a single zip file containing any and all code/scripts you write to the Assignment 1 DropBox on LEARN, in runnable format (and if necessary, any data needed to reproduce your results).

You have a number of options of how to prepare your solutions. You can typeset your solutions in a word-processing application (MS Word, LaTeX, etc.), you can write on a tablet computer, or you can write on paper and take photos. **It is your responsibility to ensure that your submission is sufficiently legible. This is particularly important if you are submitting photos of handwritten pages.** TAs have the right to take marks off for illegible answers.

Note that your solutions will be judged not only for correctness but also for the quality of your presentation and explanations.

1. **(7 marks)** Floating Point Number System

   Consider the normalized floating-point number system $F(\beta = 4, t = 5, L = -10, U = 10)$, with elements of the form

   $$\pm 0 \,.\, d_1\, d_2\, d_3\, d_4\, d_5 \times 4^p$$

   where $-10 \leq p \leq 10$. The number system uses rounding. Also, this number system is normalized, so $d_1 \neq 0$; the only exception is the zero element, in which all the mantissa digits and exponent digits are zero.

   For the following questions, state your answers in base-4 (you may use base-10 to write exponents).

   (a) (2 marks) What are the smallest and largest positive values in F?

   (b) (2 marks) What is the value of $\frac{(0.0321223)_4}{(10)_4}$ using this number system.

   (c) (1 mark) What is machine epsilon for F?

   (d) (2 marks) What fraction of the normalized numbers in F are smaller in magnitude than 1? You may use base-10 for this answer.

2. **(8 marks)** Floating Point Error Analysis

   Suppose y is a number that can be exactly represented in some floating point system F. Give a bound on the relative error of the floating point calculation

   $$(y \ominus 1) \otimes (y \oplus 1)$$

   with respect to the true value of $y^2 - 1$ in terms of machine epsilon E (assuming no overflow/underflow occurs). Recall that $\ominus$, $\oplus$ and $\otimes$ indicate floating point subtraction, addition, and multiplication, respectively. You may assume the number 1 is exactly represented in F.

3. **(8 marks)** Compute sin(x) using Power Series

   The power series for $\sin(x)$ is

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots$$

   (a) (3 marks) Create a Python function named `PowerSin`. The input is any real number x, and the output is the value of the power series for $\sin(x)$. You should include the following **while** loop in your program:

```
while sum + term != sum do
        ⋮
    sum = sum + term
        ⋮
end while
```

   Here, `sum` is a partial sum of the series, and `term` is the value of a single term in the series.

   (b) (1 mark) Make a plot of $\sin(x)$, $0 \le x \le 2\pi$, using your Python function `PowerSin(x)`. Label the x-axis, y-axis, and title.

   (c) (1 mark) What causes the **while** loop in part (a) to terminate?

   (d) (2 marks) Consider $x = \frac{\pi}{2}$, $\frac{11\pi}{2}$, $\frac{21\pi}{2}$, and $\frac{31\pi}{2}$. Calculate `PowerSin` on each these values, and compare it to the "exact" value of sin (computed using Python's built-in `sin` function). Show the computed values, exact values, errors, and the number of terms used in `PowerSin`. Display these results in table form.

   (e) (1 mark) What do you conclude about the use of floating-point arithmetic and power series to evaluate functions?

4. **(6 marks)** Polynomial Interpolation

   Given 2 constant parameters, u and v, determine whether there exists a unique cubic polynomial, $p(x)$ such that

$$p(0) = 3,\ p'(0) = 4u,\ p(1) = 3 + 5u,\ p''(1) = -4v + 6u.$$

   If such a polynomial exists, determine the coefficients of $p(x)$ in terms of u and v.

5. **(7 marks)** Cubic Spline

   Consider the following alternative representation for a cubic spline, $S(x)$:

$$S(x) = \begin{cases} a + b(x-1) + c(x-1)^2 - \frac{1}{4}(x-1)^2(x-2) & 1 \le x < 2 \\ \\ e + f(x-2) + g(x-2)^2 + \frac{1}{4}(x-2)^2(x-3) & 2 \le x \le 3. \end{cases}$$

   We also wish our cubic spline to satisfy the boundary conditions:

$$\frac{d^2S}{dx^2}(1) = 0\ ,\quad \frac{d^2S}{dx^2}(3) = 0\ .$$

(a) (2 marks) What are the conditions on the coefficients a, b, c, e, f and g such that $S(x)$ interpolates the points (1,1), (2,1), and (3,0)? Deduce the values of a and e.

(b) (1 mark) What is the condition on the coefficients such that $S'(x)$ is continuous at $x = 2$?

(c) (2 marks) Show that enforcing the boundary conditions at $x = 1$ and $x = 3$ leads to $c = -\frac{1}{4}$ and $g = -\frac{1}{2}$.

(d) (1 mark) Compute the values of b and f from part (a).

(e) (1 mark) To ensure that $S(x)$ is a cubic spline, what other condition needs to be checked? (It is not necessary to actually verify this condition for the purpose of this exercise.)

6. **(10 marks)** Spline Construction

(a) (4 marks) Write down the linear system of five equations needed to determine the derivative values s_1, s_2, s_3, s_4, s_5 for the cubic spline $S(x)$ going through the points (-3,0), (0,5), (1,3), (3,2), (4,1) and having a prescribed slope of -1 at the left end (at $x = -3$), and a "free" or natural boundary condition at the right end (at $x = 4$). The system should be given in the form $T\vec{s} = \vec{r}$ where T is a matrix, $\vec{s}$ is the vector of unknown slopes s_i, and $\vec{r}$ is the right-hand-side vector, as described in the course notes, Section 2.5. Show the calculations leading to the entries in your linear system, not just the final system.

(b) (3 marks) Next, use Python (or the mathematical tool of your choice) to actually solve the linear system you constructed in part (a), and then use the results to determine the complete cubic spline $S(x)$. Specifically, determine the a, b, c, d coefficients of each of the polynomials $S_1(x), S_2(x), S_3(x), S_4(x)$ comprising the spline, in the form

$$\begin{aligned} S_1(x) &= a_1 + b_1(x+3) + c_1(x+3)^2 + d_1(x+3)^3 && \text{for } x \in [-3, 0] \\ S_2(x) &= a_2 + b_2(x-0) + c_2(x-0)^2 + d_2(x-0)^3 && \text{for } x \in [0, 1] \\ S_3(x) &= a_3 + b_3(x-1) + c_3(x-1)^2 + d_3(x-1)^3 && \text{for } x \in [1, 3] \\ S_4(x) &= a_4 + b_4(x-3) + c_4(x-3)^2 + d_4(x-3)^3 && \text{for } x \in [3, 4]. \end{aligned}$$

(You can give the final coefficients to 4 significant digits. You may use a calculator or computer to perform the individual calculations needed to find the coefficients. There is no need to include any code for this part. However, in Python/Numpy, you may find the function `numpy.linalg.solve` to be useful for solving linear systems.)

(c) (3 marks) Write a short Jupyter notebook that produces a single plot of the resulting smooth curve by evaluating the polynomials you found over their respective domains at a finely spaced set of x-values. Submit your code and plots.

7. **(10 marks)** Using Splines for Parametric Curves

Create a parametric curve representation of your nickname written in your handwriting. This representation should be based on parametric spline interpolation. Your nickname must have at least two curved segments. Figure 1 below shows an example using 3 coarsly-sampled segments.

Steps:

(a) (2 marks) Draw your nickname and select interpolation points. You can do this on graph paper by hand, or you can use `ginput`. Make sure the interpolation points are not too

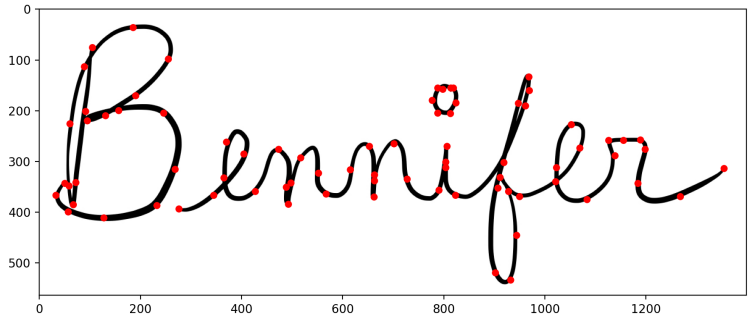

Figure 1: The nickname "Bennifer" written in 3 segments

close together. Your curved segments should be sampled sparingly. Figure 1 shows an example.

(b) (2 marks) Add code to the notebook (`A1Q7.ipynb`) to initialize and store the interpolation points for each segment. For example, you could store the coordinates of the interpolation points for the first segment in the arrays `x1`+ and `y1`, and you would have one `x-y` pair of arrays for each segment. Your data arrays should be hardcoded into your script.

(c) (3 marks) Complete the function `ParametricSpline`; it takes the interpolation points of one segment as input, and outputs a pair of cubic spline functions (one for `x(t)` and one for `y(t)`, along with an array, `t`, that holds the parameter value at the interpolation points. See the function's documentation for more details. The parameter `t` must reflect the pseudo-arclength of the curve. You can use `scipy.make_interp_spline` (and return the spline functions that it creates).

(d) (1 mark) Add lines to the notebook to call `ParametricSpline` for each segment in your nickname.

(e) (2 marks) Add more lines to the notebook to plot your nickname. The plot should graph the parametric spline segments using a refined selection of parameter values (eg. 1000 points per segment). The plot should also display the original interpolation points, and the `x` and `y` axes should be rendered using the same scale ("`plt.axis('equal')`"). Figure 1 is an example of what your output should look like.