# MAT 357–Numerical Analysis

Fall 2018

*Prerequisites: MAT 250, MAT 258*

## General Information

Class Schedule: Wednesdays and Fridays 3:30–4:50pm
Classroom: Van Gogh (SR 4E)
Instructor: Michael Daniel Samson
Contact: `mat357@mdvsamson.work`, +65 6577 1944
Class Webpage: Moodle, `mdvsamson.work/mat357`
Office Hours: Tuesdays 5:30–7:30pm, Mondays 2:00–7:00pm or by appointment (through email)

## Description

This course covers the numerical techniques arising in many areas of computer science and applied mathematics. Such techniques provide essential tools for obtaining approximate solutions to non-linear equations arising from the construction of mathematical models of real-world phenomena. Topics of study include root-finding, interpolation, approximation of functions, cubic splines, integration, and differential equations. Further topics may include stability, iterative methods for solving systems of equations, eigenvalue approximation, and the fast Fourier transform.

## Course Objectives and Learning Outcomes

Upon completing this course students should be able to:

- Understand numerical methods and apply them in the following situations:
  - Find roots of functions
  - Interpolate $y(x)$ given some data points $(x_i, y_i)$
  - Find derivatives at a point $x$, given some data points
  - Integrate a function numerically
- Implement numerical algorithms
- Understand the strengths and the pitfalls of the methods discussed

## Textbooks

*Numerical Methods in Engineering with Python3*, Jaan Kiusalaas, Cambridge University Press, ISBN-978-1-107-03385-6

## Outline and Tentative Dates

The following schedule is subject to change.

September 5: Overview, Computational Considerations, Taylor Polynomials
September 7: Rootfinding Methods
    *Interpolation*
September 12: Polynomial Spaces
September 14: Orthogonal Polynomials
September 19, 21: Polynomial Interpolation
September 26, 28: Spline Interpolation: Cubic Splines
October 3, 5: Curve-Fitting: Least-Squares Fit
October 10: **Examination** (discussion on October 12)
    *Numerical Differentiation*
October 17, 19: Finite-Difference Approximations
October 24, 26: Richardson Extrapolation
October 31, November 2: Derivatives by Interpolation
November 6: *Deepavali*
November 7: **Examination** (discussion on November 9)
    *Numerical Integration*
November 14: Newton-Cotes Formulas
November 16: Romberg Integration
November 21, 23: Gaussian Quadrature
November 28: Pseudospectral Methods
November 30: Concepts in Numerical Linear Algebra
December 5, 7: Introduction to Ordinary Differential Equations
December 10–14: **Examination** (schedule to be announced by DigiPen Admin)

## Grading Policy

The examination on week fifteen is *optional*. You must inform the instructor of your decision to not take the final exam *by week fourteen*.

The relative weights of homework, quiz, projects, and examinations are:

10% Homework (weekly)
30% Projects (three, due four weeks after specifications)
60% Examinations (drop the lowest of three)

Grades will be computed out of 40 points. Letter grades will be computed subject to:

35 = at least A
30 = at least B
20 = at least C- (passing)

*To pass the course,*

***have a passing examination average*** *and* ***the course total should be greater than or equal to 20****.*

## Late Policy

Late homework and project submissions **will not** be accepted, unless authorized by the instructor.

## Statement

This course is designed to give an overview of simple numerical methods in common usage. As with most established algorithms, implementations for numerous languages can be readily found, covering most, if not all, of the discussed methods, with varying degrees of effectiveness. The course will focus the underlying mathematics of the methods, and provide some analysis to differentiate between them.

## Last Day to Withdraw

The final date to withdraw from this course is **28 October 2018**. Scores for six (6) homework submissions, one (1) project and one (1) examination should be available before this date. In order to withdraw from a course, in accordance with policy, contact your advisor or the Registrar to begin the withdrawal process—it is *not sufficient* simply to stop attending class or to inform the instructor. The last day for withdrawal from this course is cited in the official catalog.

## Academic Integrity Policy

Academic dishonesty *in any form* will not be tolerated in this course. Cheating, copying, plagiarizing, or any other form of academic dishonesty (including doing someone else's individual assignments) will result in, at the very minimum, a zero on the assignment in question, and could result in a failing grade in the course or even expulsion from DigiPen. In particular, code sources should be cited in the submitted documentation when lifted.

## External Preparation

It is expected that the students in this class spend six (6) hours on average per week for outside classroom activities through the semester, including, but not limited to, homework, reading assignments, project implementation, group discussions, preparation of examinations, etc.

## Disability Support Service

Students who have special needs or medical conditions and require formal accommodations in order to fully participate or effectively demonstrate learning in this class should contact the Student Life & Advising Office (`studentlife.sg@digipen.edu`) at the beginning of each semester. A Student Life & Advising Officer will meet with the student privately to discuss how the accommodations will be implemented.

Name: ______________________________

**It is suggested to implement the rootfinding methods, log the iterations, and submit the log for the problems.**

3. The smallest positive, nonzero root of $\cosh x \cos x - 1 = 0$ lies in the interval $(4, 5)$. Compute this root by Ridder's method.

   The log, for $10^{-6}$ accuracy:

```
a = 4.00000000, f(a) = -18.8498521903
b = 5.00000000, f(b) = 20.0505561817
Iter #1: c = 4.73741114, f(c) = 0.4279543686
Iter #2: c = 4.73034897, f(c) = 0.0177733380
Iter #8: c = 4.73004074, f(c) = -0.0000001830
```

   By the second iteration, the root can be approximated as 4.73.

5. A root of the equation $\tan x - \tanh x = 0$ lies in $(7.0, 7.4)$. Find this root with three decimal place accuracy by the method of bisection.

   The log, for $10^{-4}$ accuracy:

```
a = 7.00000000, f(a) = -0.1285503542
b = 7.40000000, f(b) = 1.0492849164
Iter #1: c = 7.20000000, f(c) = 0.3046220548
Iter #2: c = 7.10000000, f(c) = 0.0648944885
Iter #3: c = 7.05000000, f(c) = -0.0364914712
Iter #4: c = 7.07500000, f(c) = 0.0129175441
Iter #5: c = 7.06250000, f(c) = -0.0120920521
Iter #6: c = 7.06875000, f(c) = 0.0003345637
Iter #7: c = 7.06562500, f(c) = -0.0058980464
Iter #8: c = 7.06718750, f(c) = -0.0027865970
Iter #9: c = 7.06796875, f(c) = -0.0012272343
Iter #10: c = 7.06835938, f(c) = -0.0004466402
Iter #11: c = 7.06855469, f(c) = -0.0000561145
Iter #12: c = 7.06865234, f(c) = 0.0001392055
```

   The root is, approximately, 7.069.

7. Determine the two roots of $\sin x + 3\cos x - 2 = 0$ that lie in the interval $(-2, 2)$. Use secant lines.

   The log, for $5 \times 10^{-3}$ accuracy and secant method, for the negative root:

```
x_0 = -1.00000000, f(x_0) = -1.2205640672;
x_1 = 0.00000000, f(x_1) = 1.0000000000
x_2 = -0.45033603, f(x_2) = 0.2656345897
x_3 = -0.61323153, f(x_3) = -0.1221356829
x_4 = -0.56192447, f(x_4) = 0.0058781406
x_5 = -0.56428039, f(x_5) = 0.0001131160
```

   The log, for $5 \times 10^{-3}$ accuracy and secant method, for the positive root:

```
x_0 = 1.00000000, f(x_0) = 0.4623779024;
x_1 = 2.00000000, f(x_1) = -2.3391430828
x_2 = 1.16504531, f(x_2) = 0.1029329524
x_3 = 1.20023846, f(x_3) = 0.0185319321
x_4 = 1.20796582, f(x_4) = -0.0003384023
x_5 = 1.20782725, f(x_5) = 0.0000010548
```

   The roots are, approximately, $-0.56$ and $1.21$.

19. The speed $v$ of a Saturn V rocket in vertical flight near the surface of earth can be approximated by

$$\begin{aligned} v &= u \ln \frac{M_0}{M_0 - \dot{m}t} - gt, \text{ where} \\ u &= 2510 \text{ m/s} = \text{velocity of exhaust relative to the rocket,} \\ M_0 &= 2.8 \times 10^6 \text{ kg} = \text{mass of rocket at liftoff,} \\ \dot{m} &= 13.3 \times 10^3 \text{ kg/s} = \text{rate of fuel consumption,} \\ g &= 9.81 \text{ m/s}^2 = \text{gravitational acceleration,} \\ t &= \text{time measured from liftoff.} \end{aligned}$$

Determine the time when the rocket reaches the speed of sound (355 m/s).

Setting $f(t) = u \ln \dfrac{M_0}{M_0 - \dot{m}t} - gt - v$, for the values above, and $v = 355$ m / s, the log, for $10^{-6}$ accuracy:

```
a = 73.00000000, f(a) = -2.3836098006
b = 74.00000000, f(b) = 6.1241208164
Iter #1: c = 73.28175842, f(c) = 0.0000027490
Iter #2: c = 73.28175809, f(c) = -0.0000000000
```

The rocket reaches Mach 1 approximately at 73.28 seconds past launch.

23. Determine the coordinates of the two points where the circles $(x-2)^2 + y^2 = 4$ and $x^2 + (y-3)^2 = 4$ intersect. Start by estimating the locations of the points from a sketch of the circles, and then use the Newton-Raphson method to compute the coordinates.

One approach can be to solve one equation for $y = f(x)$, and use that value for $y$ in the other equation to get a function solely in $x$: the first equation gives $y = \pm\sqrt{4-(x-2)^2}$, which, when plugged into the second equation, gives $g_{\pm}(x) = x^2 + (3 \pm \sqrt{4-(x-2)^2})^2 - 4 = x^2 + 9 \pm 6\sqrt{4-(x-2)^2} - (x-2)^2 = 4x + 5 \pm 6\sqrt{4x - x^2}$ whose roots are roots of $G(x) = (4x+5)^2 - 36(4x - x^2) = 52x^2 - 104x + 25$, which has roots at $x = 1 \pm \dfrac{3}{2}\sqrt{\dfrac{3}{13}}$. To determine the $y$-components, which can be determined by noting that the intersection points lie on the line $6y = 4x + 5$, determining $y = \dfrac{3}{2} \pm \sqrt{\dfrac{3}{13}}$. The log, for $10^{-6}$ using the Newton-Raphson method:

```
x_0 = 0.00000000, f(x_0) = 25.0000000000
x_1 = 0.24038462, f(x_1) = 3.0048076923
x_2 = 0.27842016, f(x_2) = 0.0752285214
x_3 = 0.27942261, f(x_3) = 0.0000522556

x_0 = 2.00000000, f(x_0) = 25.0000000000
x_1 = 1.75961538, f(x_1) = 3.0048076923
x_2 = 1.72157984, f(x_2) = 0.0752285214
x_3 = 1.72057739, f(x_3) = 0.0000522556
```

From these, the points are approximately $(0.279423, 1.019616)$ and $(1.720577, 1.980384)$, where the exact values are $\left(1 - \dfrac{3}{2}\sqrt{\dfrac{3}{13}}, \dfrac{3}{2} - \sqrt{\dfrac{3}{13}}\right)$ and $\left(1 + \dfrac{3}{2}\sqrt{\dfrac{3}{13}}, \dfrac{3}{2} + \sqrt{\dfrac{3}{13}}\right)$

**Source**: Jaan Kiusalaas, *Numerical Methods in Engineering with Python 3*, Problem Set 4.1

Name: ______________________________

1. Show that the *Chebyshev polynomials* $T_n(x) = \cos(n \arccos x)$ are a family of polynomials where the degree of $T_n(x)$ is $n$: to be precise, show

$$T_0(x) \equiv 1, \qquad T_1(x) = x, \qquad \text{and that} \qquad T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), \quad n \geq 2.$$

Note that the set $\{T_n(x)\}_{n=0}^N$ is a basis for $\mathbb{P}_N$.

The first two statements are straightforward: $T_0(x) = \cos(0 \arccos x) \equiv \cos 0 = 1$; $T_1(x) = \cos(\arccos x) = x$.

Using the definition $T_n(x) = \cos(n \cos^{-1} x)$, let $x = \cos\theta$, so that $0 \leq \theta \leq \pi$, $T_n(x) = T_n(\cos\theta) = \cos n\theta$, and if $n \geq 1$, $\cos[(n \pm 1)\theta] = \cos\theta\cos n\theta \mp \sin\theta \sin n\theta$, so $2\cos\theta\cos n\theta = \cos[(n+1)\theta] + \cos[(n-1)\theta]$, so $T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$.

2. Given a polynomial of degree $n$, $p(x) = \sum_{k=0}^{n} c_k T_k(x)$, and points $x_0 < x_1 < \cdots < x_n$, find the coefficients $\{\ell_k\}_{k=0}^n$ such that $p(x) = \sum_{k=0}^{n} \ell_k L_k(x)$, where $L_k(x)$ is the Lagrange polynomial associated with $x_k$ for the points given, i.e.

$$L_k(x_j) = \delta_{jk} = \begin{cases} 1, & \text{if } j = k \\ 0, & \text{otherwise.} \end{cases}$$

Since $p(x) \in \mathbb{P}_n[x_0, x_n]$, it is exactly interpolated by the $n+1$ points $x_i$: $p(x) = \sum_{m=0}^{n} p(x_m) L_m(x)$. Thus, $\ell_m = p(x_m) = \sum_{k=0}^{n} c_k T_k(x_m)$.

3. Two functions $f(x), g(x) \in F$, a function space, are said to be *orthogonal with respect to an inner product* $\langle \cdot, \cdot \rangle : F \times F \to \mathbb{R}$ if $\langle f, g \rangle = 0$. An example of an inner product on a vector space is the dot product on $\mathbb{R}^2$ and $\mathbb{R}^3$.

Show that the Chebyshev polynomials are orthogonal with respect to the inner product given below:

$$\langle T_m(x), T_n(x) \rangle_{\omega^{-\frac{1}{2},-\frac{1}{2}}} = \int_{-1}^{1} T_m(x) T_n(x) \omega^{-\frac{1}{2},-\frac{1}{2}} \,\mathrm{d}x = \begin{cases} 0, & \text{if } m \neq n, m, n \geq 0, \\ \pi, & \text{if } m = n = 0, \\ \pi/2, & \text{if } m = n > 0, \end{cases}$$

where $\omega^{-\frac{1}{2},-\frac{1}{2}} = \dfrac{1}{\sqrt{1-x^2}}$.

Letting $x = \cos\theta$, gives $T_n(x) = \cos n\theta$, so that $\mathrm{d}x = -\sin\theta \,\mathrm{d}\theta$ and

$$\langle T_m(x), T_n(x) \rangle_{\omega^{-\frac{1}{2},-\frac{1}{2}}} = \int_{-1}^{1} \frac{T_m(x)T_n(x)}{\sqrt{1-x^2}} \,\mathrm{d}x = \int_{\pi}^{0} \frac{\cos m\theta \cos n\theta}{\sin\theta} (-\sin\theta \,\mathrm{d}\theta) = \int_0^{\pi} \cos m\theta \cos n\theta \,\mathrm{d}\theta.$$

Using $\cos[(m+n)\theta] + \cos[(m-n)\theta] = 2\cos m\theta \cos n\theta$,

$$\langle T_m(x), T_n(x) \rangle_{\omega^{-\frac{1}{2},-\frac{1}{2}}} = \frac{1}{2}\int_0^{\pi} \cos[(m+n)\theta] \,\mathrm{d}\theta + \frac{1}{2}\int_0^{\pi} \cos[(m-n)\theta] \,\mathrm{d}\theta.$$

If $m \neq n$, then $m \pm n \neq 0$ and

$$\langle T_m(x), T_n(x) \rangle_{\omega^{-\frac{1}{2},-\frac{1}{2}}} = \frac{1}{2}\left[\frac{\sin[(m+n)\theta]}{m+n} + \frac{\sin[(m-n)\theta]}{m-n}\right]_0^{\pi} = 0.$$

If $m = n > 0$, then $m - n = 0$ and

$$\langle T_m(x), T_n(x) \rangle_{\omega^{-\frac{1}{2},-\frac{1}{2}}} = \frac{1}{2}\left[\frac{\sin[(m+n)\theta]}{m+n} + \theta\right]_0^{\pi} = \frac{\pi}{2}.$$

If $m = n = 0$, then $m \pm n = 0$ and

$$\langle T_m(x), T_n(x) \rangle_{\omega^{-\frac{1}{2},-\frac{1}{2}}} = \frac{1}{2}\left[2\theta\right]_0^{\pi} = \pi.$$

4. Given a polynomial of degree $n$, $p(x) = \sum_{k=0}^{n} \ell_k L_k(x)$, where $L_k(x)$ is the Lagrange polynomial associated with $x_k$ for the points $x_0 < x_1 < \cdots < x_n$, find the coefficients $\{c_k\}_{k=0}^{n}$ such that $p(x) = \sum_{k=0}^{n} c_k T_k(x)$, where $T_k(x)$ is the Chebyshev polynomial of degree $k$.

   Since the Chebyshev polynomials are orthogonal, $L_m(x) = \sum_{k=0}^{n} t_{mk} T_k(x)$ where

$$t_{mk} = \frac{2 - \delta_{0k}}{\pi} \left\langle L_m(x), T_k(x) \right\rangle_{\omega^{-\frac{1}{2}, -\frac{1}{2}}}, \qquad \delta_{0k} = \begin{cases} 0, & k > 0 \\ 1, & k = 0, \end{cases}$$

   so $p(x) = \sum_{m=0}^{n} \ell_m L_m(x) = \sum_{m=0}^{n} \ell_m \sum_{k=0}^{n} t_{mk} T_k(x) = \sum_{k=0}^{n} \left[ \sum_{m=0}^{n} t_{mk} \ell_m \right] T_k(x)$, which gives $c_k = \frac{2 - \delta_{0k}}{\pi} \sum_{m=0}^{n} \left\langle L_m(x), T_k(x) \right\rangle_{\omega^{-\frac{1}{2}, -\frac{1}{2}}} \ell_m$.

You are currently working with physicists that are using a particle collider for some experiments. They have been testing the monitoring equipment, by passing a single particle and monitoring its motion on a plane, where the particle is at position $(0, 0)$ at the starting time $t = 0$. (The units have been omitted, as the measurements are on such a small scale, and in order to anonymize the computations, somewhat.) They want to model the measured motion from the data, and predict the position of the particle at a given point within the duration of the measurements.

For this project, data is provided in a text file named `partmove.txt`, which has numbers at each line, separated by spaces within lines, and lines separated by carriage returns, and no other characters are included. The first line contains $N$, which is an integer from 1 to 16. The next $N$ lines contains pairs of floating point numbers corresponding to the position $(x_i, y_i)$ at the $i$th measurement, where $0 < t_1 < t_2 < \cdots < t_N$ is the time at which the measurements are taken. The next line will contain $t\ x\ y$, where $t$ time at which the position is to be predicted, and $(x, y)$ the actual position at which the particle is found at time $t$.

1. The measurements are taken at evenly-spaced intervals, with $t_N = 2$. Model the position of the particle using polynomials in each direction, and output the discrepancy between the interpolated position at time $t$ with the given $(x, y)$. **(3)**

2. The measurements are taken at evenly-spaced intervals, with $t_N = 2$. Model the position of the particle using natural cubic splines in each direction, and output the discrepancy between the interpolated position at time $t$ with the given $(x, y)$. **(3)**

3. The measurements are taken $1 + t_i$, where $t_i$ at the roots of $T_N(t)$, the Chebyshev polynomial of degree $N$. Chebyshev polynomials, with degree $n \geq 0$, can be given in the following form

$$T_n(t) = \cos(n \arccos t).$$

Model the position of the particle using polynomials, and output the discrepancy between the interpolated position at time $t$ with the given $(x, y)$. **(3)**

4. The measurements are taken at $1 + t_i$, where $t_i$ the roots of $P_N(t)$, the Legendre polynomial of degree $N$. Legendre polynomials, with degree $n \geq 0$, are given recursively as

$$P_0(t) \equiv 1; \quad P_1(t) = t; \quad (n+1)P_{n+1}(t) + nP_{n-1}(t) = (2n+1)tP_n(t), n > 0.$$

It is known that the roots of the Legendre polynomial $P_n$ are also simple and lie in the interval $[-1, 1]$. It is also known that

$$P_n'(t) = \sum_{\substack{k=0 \\ n+k \text{ even}}}^{n-1} (2k+1)P_k(t),$$

and iterative methods for determining the roots of the Legendre polynomial $P_n$ often start from the roots of the Chebyshev polynomial $T_n$.

Model the position of the particle using polynomials, and output the discrepancy between the interpolated position at time $t$ with the given $(x, y)$. **(3)**

5. The measurements are taken at evenly-spaced intervals, with $t_N = 2$. It is presumed that the model of the position of the particle is dependent only on the initial velocity and an unknown constant force—that is, the position is quadratic in both directions. Output the discrepancy between the interpolated position at time $t$ with the given $(x, y)$. **(3)**

Given data points $\{(x_i, y_i)\}_{i=0}^{n}$, the polynomial interpolant of degree $n$, $P_n(x)$ can be determined from the divided difference tableau generated by:

$$[y_i] = y_i, 0 \le i \le n; \qquad [y_i, \dots, y_{i+k}] = \frac{[y_{i+1}, \dots, y_{i+k}] - [y_i, \dots, y_{i+k-1}]}{x_{i+k} - x_i}, 0 \le i \le n-k, 1 \le k \le n.$$

Thus,

$$\begin{aligned} P_n(x) &= y_0 + [y_0, y_1](x - x_0) + [y_0, y_1, y_2](x - x_0)(x - x_1) + \cdots + [y_0, \dots, y_n](x - x_0) \cdots (x - x_{n-1}) \\ &= (\cdots([y_0, \dots, y_n](x - x_{n-1}) + [y_0, \dots, y_{n-1}])(x - x_{n-2}) + \cdots + [y_0, y_1])(x - x_0) + y_0 \\ &= y_n + [y_{n-1}, y_n](x - x_n) + [y_{n-2}, y_{n-1}, y_n](x - x_n)(x - x_{n-1}) + \cdots + [y_0, \dots, y_n](x - x_n) \cdots (x - x_1). \end{aligned}$$

The generated polynomial is of degree $n$, and to show that this is the polynomial interpolant, inductively:

When $n = 0$, the only data point is $(x_0, y_0)$, and the divided difference scheme gives $P_0(x) \equiv y_0$, which satisfies the conditions.

When $n = 1$, $P_1(x) = P_0(x) + [y_0, y_1](x - x_0) = y_0 + \dfrac{y_1 - y_0}{x_1 - x_0}(x - x_0)$:

$$P_1(x_0) = P_0(x_0) = y_0; \qquad P_1(y_1) = y_0 + \frac{y_1 - y_0}{x_1 - x_0}(x_1 - x_0) = y_0 + y_1 - y_0 = y_1,$$

thus $P_1(x)$ interpolates $\{(x_0, y_0), (x_1, y_1)\}$.

Consider $Q_1(x) = y_1 + [y_0, y_1](x - x_1)$: $Q_1(x_1) = y_1$ and

$$Q_1(x_0) = y_1 + [y_0, y_1](x_0 - x_1) = y_1 + \frac{y_1 - y_0}{x_1 - x_0}(x_0 - x_1) = y_1 - (y_1 - y_0) = y_0.$$

Thus, $Q_1(x) = P_1(x)$, and the conditions are satisfied.

Assume then that conditions are satisfied for any $n$ points: in particular, for the first $n$ points $\{(x_i, y_i)\}_{i=0}^{n-1}$—that is,

$$\begin{aligned} P_{n-1}(x) &= y_0 + [y_0, y_1](x - x_0) + \cdots + [y_0, \dots, y_{n-1}](x - x_0) \cdots (x - x_{n-2}) \\ &= y_{n-1} + [y_{n-2}, y_{n-1}](x - x_{n-1}) + \cdots + [y_0, \dots, y_{n-1}](x - x_{n-1}) \cdots (x - x_1), \end{aligned}$$

with $P_{n-1}(x_i) = y_i$ for $0 \le i \le n-1$. Note that sub-tableaus for consecutive data points interpolates those points:

$$\begin{aligned} P(x) &= y_i + [y_i, y_{i+1}](x - x_i) + \cdots + [y_i, \dots, y_j](x - x_i) \cdots (x - x_{j-1}) \\ &= y_j + [y_{j-1}, y_j](x - x_j) + \cdots + [y_i, \dots, y_j](x - x_j) \cdots (x - x_{i+1}) \end{aligned}$$

interpolates $\{(x_k, y_k)\}_{k=i}^{j}$ for $0 \le i \le j \le n$.

Then $P_n(x) = P_{n-1}(x) + [y_0, \dots, y_n](x - x_0) \cdots (x - x_{n-1})$ and, for $0 \le i \le n-1$, $P_n(x_i) = P_{n-1}(x_i) = y_i$. Also,

$$\begin{aligned} P_n(x_n) &= P_{n-1}(x_n) + [y_0, \dots, y_n](x_n - x_0) \cdots (x_n - x_{n-1}) \\ &= P_{n-1}(x_n) + ([y_1, \dots, y_n] - [y_0, \dots, y_{n-1}])(x_n - x_1) \cdots (x_n - x_{n-1}) \\ &= P_{n-1}(x_n) + ([y_2, \dots, y_n] - [y_1, \dots, y_{n-1}])(x_n - x_2) \cdots (x_n - x_{n-1}) - [y_0, \dots, y_{n-1}](x_n - x_1) \cdots (x_n - x_{n-1}) \\ &\vdots \\ &= P_{n-1}(x_n) + [y_{n-1}, y_n](x_n - x_{n-1}) - [y_{n-2}, y_{n-1}](x_n - x_{n-1}) - \cdots - [y_0, \dots, y_{n-1}](x_n - x_1) \cdots (x_n - x_{n-1}) \\ &= P_{n-1}(x_n) + y_n - y_{n-1} - [y_{n-2}, y_{n-1}](x_n - x_{n-1}) - \cdots - [y_0, \dots, y_{n-1}](x_n - x_1) \cdots (x_n - x_{n-1}), \end{aligned}$$

where

$$P_{n-1}(x_n) = y_{n-1} + [y_{n-2}, y_{n-1}](x_n - x_{n-1}) + \cdots + [y_0, \ldots, y_{n-1}](x_n - x_{n-1}) \cdots (x_n - x_1),$$

so $P_n(x_n) = y_n$, and $P_n(x)$ interpolates $\{(x_i, y_i)\}_{i=0}^n$.

Consider $Q_n(x) = y_n + [y_{n-1}, y_n](x - x_n) + \cdots + [y_0, \ldots, y_n](x - x_n) \cdots (x - x_1)$: note that

$$\begin{aligned} Q_{n-1}(x) &= y_n + [y_{n-1}, y_n](x - x_n) + \cdots + [y_1, \ldots, y_n](x - x_n) \cdots (x - x_2) \\ &= y_1 + [y_1, y_2](x - x_1) + \cdots + [y_1, \ldots, y_n](x - x_1) \cdots (x - x_{n-1}) \end{aligned}$$

should interpolate the $n$ points $\{(x_i, y_i)\}_{i=1}^n$, and $Q_n(x) = Q_{n-1}(x) + [y_0, \ldots, y_n](x - x_n) \cdots (x - x_1)$. Thus, for $1 \leq i \leq n$, $Q_n(x_i) = Q_{n-1}(x_i) = y_i$. So,

$$\begin{aligned} Q_n(x_0) &= Q_{n-1}(x_0) + [y_0, \ldots, y_n](x_0 - x_n) \cdots (x_0 - x_1) \\ &= Q_{n-1}(x_0) - ([y_1, \ldots, y_n] - [y_0, \ldots, y_{n-1}])(x_0 - x_{n-1}) \cdots (x_0 - x_1) \\ &= Q_{n-1}(x_0) - ([y_1, \ldots, y_{n-1}] - [y_0, \ldots, y_{n-2}])(x_0 - x_{n-2}) \cdots (x_0 - x_1) - [y_1, \ldots, y_n](x_0 - x_{n-1}) \cdots (x_0 - x_1) \\ &\vdots \\ &= Q_{n-1}(x_0) + [y_0, y_1](x_0 - x_1) - [y_1, y_2](x_0 - x_1) - \cdots - [y_1, \ldots, y_n](x_0 - x_{n-1}) \cdots (x_0 - x_1) \\ &= Q_{n-1}(x_0) + y_0 - y_1 - [y_1, y_2](x_0 - x_1) - \cdots - [y_1, \ldots, y_n](x_0 - x_{n-1}) \cdots (x_0 - x_1), \end{aligned}$$

where

$$Q_{n-1}(x_0) = y_1 + [y_1, y_2](x_0 - x_1) + \cdots + [y_1, \ldots, y_n](x_0 - x_1) \cdots (x_0 - x_{n-1}),$$

so $Q_n(x_0) = y_0$, and $Q_n \equiv P_n$, so the conditions are satisfied.

Given data points $\{(x_i, y_i)\}_{i=0}^n$ with underlying function $f(x)$ such that $y_i = f(x_i)$ for $0 \leq i \leq n$, where $\dfrac{\mathrm{d}^{n+1}}{\mathrm{d}x^{n+1}} f$ exists on an interval $I$ containing $\{x_i\}_{i=0}^n$. Let $P_n(x)$ be the polynomial of degree $n$ such that $y_i = P_n(x_i)$ for $0 \leq i \leq n$, and $x^*$ be in $I$, $x^* \neq x_i$, $0 \leq i \leq n$.

Let

$$\begin{aligned} g(t) &= f(t) - P_n(t) - [f(x^*) - P_n(x^*)] \frac{(t - x_0)(t - x_1) \cdots (t - x_n)}{(x^* - x_0)(x^* - x_1) \cdots (x^* - x_n)} \\ &= f(t) - P_n(t) - [f(x^*) - P_n(x^*)] \prod_{i=0}^{n} \frac{t - x_i}{x^* - x_i}. \end{aligned}$$

Note that $g(x_i) = g(x^*) = 0$, $0 \leq i \leq n$. Thus, $g(t) = 0$ for $n + 2$ points in $I$; applying Rolle's theorem means that $g'(t) = 0$ for $n + 1$ points in $I$; applying Rolle's theorem means that $g''(t) = 0$ for $n$ points in $I$; $\ldots$; applying Rolle's theorem means that $g^{(n+1)}(\xi) = 0$ for some point in $I$.

But, since $P_n$ is of degree $n$ and $\prod\limits_{i=0}^{n} (t - x_i)$ is of degree $n + 1$,

$$\begin{aligned} g^{(n+1)}(t) = f^{(n+1)}(t) - \frac{[f(x^*) - P_n(x^*)](n+1)!}{\prod\limits_{i=0}^{n} (x^* - x_i)} &\implies 0 = g^{(n+1)}(\xi) = f^{(n+1)}(\xi) - \frac{[f(x^*) - P_n(x^*)](n+1)!}{\prod\limits_{i=0}^{n} (x^* - x_i)} \\ &\implies f(x^*) - P_n(x^*) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^{n} (x^* - x_i). \end{aligned}$$

This indicates that, for any $x \in I$,

$$|f(x) - P_n(x)| = \frac{|f^{(n+1)}(\xi)|}{(n+1)!} \prod_{i=0}^{n} |x - x_i|.$$

**Source**: Richard L. Burden and J. Douglas Faires, *Numerical Analysis*, 9th edition (International), Theorem 3.3

Name: ______________________________

2. Find the zero of $y(x)$ from the following data:

| $x$ | 0 | 0.5 | 1 | 1.5 | 2 | 2.5 | 3 |
|---|---|---|---|---|---|---|---|
| $y$ | 1.8421 | 2.4694 | 2.4921 | 1.9047 | 0.8509 | −0.4112 | −1.5727 |

Use Lagrange's interpolation over (a) three; and (b) four nearest-neighbor data points.
*Hint*: After finishing part (a), part (b) can be computed with a relatively small effort.
The divided difference tableau for the last four points is

$$
\begin{array}{ccccc}
x_i & y_i = [y_i] & [y_i, y_{i+1}] & [y_i, y_{i+1}, y_{i+2}] & [y_0, y_1, y_2, y_3] \\
\hline
3 & -1.5727 & & & \\
 & & -2.323 & & \\
2.5 & -0.4112 & & 0.2012 & \\
 & & -2.5242 & & \dfrac{3089}{7500} \\
2 & 0.8509 & & -0.4166 & \\
 & & -2.1076 & & \\
1.5 & 1.9047 & & &
\end{array}
$$

Thus, using the last three points for (a) gives

$$
\begin{aligned}
p_2(x) &= [0.2012(x-2.5) - 2.323](x-3) - 1.5727 = 0 &&\implies x = \frac{34246 - \sqrt{620477072}}{4024} \approx 2.3327, \\
p_3(x) &= \frac{3089}{7500}(x-2)(x-2.5)(x-3) + p_2(x) = 0 &&\implies x \approx 2.3386.
\end{aligned}
$$

An alternative approach would be to do an inverse interpolation, i.e. determine an interpolant $x = q(y)$ from the divided difference table

$$
\begin{array}{ccccc}
y_i & x_i = [x_i] & [x_i, x_{i+1}] & [x_i, x_{i+1}, x_{i+2}] & [x_0, x_1, x_2, x_3] \\
\hline
-1.5727 & 3 & & & \\
 & & -0.430477830391735 & & \\
-0.4112 & 2.5 & & 0.014157744169435 & \\
 & & -0.396165121622692 & & -0.013795088428626 \\
0.8509 & 2 & & -0.03381329633226977 & \\
 & & -0.474473334598596 & & \\
1.9047 & 1.5 & & &
\end{array}
$$

Thus, using the last three points for (a) gives

$$
\begin{aligned}
x = q_2(0) &\approx [0.014157744169435(0.4112) - 0.430477830391735](1.5727) + 3 \approx 2.3321, \\
x = q_3(0) &\approx q_2(0) - 0.013795088428626(-0.8509)(0.4112)(1.5727) \approx 2.3396.
\end{aligned}
$$

5. Given the data

| $x$ | 0 | 0.5 | 1 | 1.5 | 2 |
|---|---|---|---|---|---|
| $y$ | −0.7854 | 0.6529 | 1.7390 | 2.2071 | 1.9425 |

find $y$ at $x = \pi/4$ and at $\pi/2$. Use the method that you consider to be most convenient.
The divided difference tableau is

| $x_i$ | $y_i = [y_i]$ | $[y_i, y_{i+1}]$ | $[y_i, y_{i+1}, y_{i+2}]$ | $[y_i, y_{i+1}, y_{i+2}, y_{i+3}]$ | $[y_0, y_1, y_2, y_3, y_4]$ |
|---|---|---|---|---|---|
| $0$ | $-0.7854$ | | | | |
| | | $2.8766$ | | | |
| $0.5$ | $0.6529$ | | $-0.7044$ | | |
| | | $2.1722$ | | $\frac{-1329}{3750}$ | |
| $1$ | $1.739$ | | $-1.236$ | | $-\frac{1511}{15000}$ |
| | | $0.9362$ | | $\frac{-1147}{7500}$ | |
| $1.5$ | $2.2071$ | | $-1.4654$ | | |
| | | $-0.5292$ | | | |
| $2$ | $1.9425$ | | | | |

Thus,

$$p(x) = \left[\left(\left[\frac{1511}{15000}\left(x - \frac{3}{2}\right) + \frac{1329}{3750}\right](1-x) - \frac{1761}{2500}\right)\left(x - \frac{1}{2}\right) + \frac{14383}{5000}\right]x - \frac{3927}{5000},$$

thus, $p(\pi/4) \approx 1.3296$ and $p(\pi/2) \approx 2.2013$.

Using the four nearest neighbors for each point:

$$\begin{aligned} p_3\left(\frac{\pi}{4}\right) &= \left(\left[\frac{1329}{3750}\left(1 - \frac{\pi}{4}\right) - \frac{1761}{2500}\right]\left(\frac{\pi}{4} - \frac{1}{2}\right) + \frac{14383}{5000}\right)\frac{\pi}{4} - \frac{3927}{5000} \approx 1.3303 \\ q_3\left(\frac{\pi}{2}\right) &= \left(\left[\frac{1147}{7500}\left(\frac{3}{2} - \frac{\pi}{2}\right) - \frac{618}{500}\right]\left(\frac{\pi}{2} - 1\right) + \frac{10861}{5000}\right)\left(\frac{\pi}{2} - \frac{1}{2}\right) + \frac{6529}{10000} \approx 2.2168. \end{aligned}$$

Using the three middle data points:

$$q_2(x) = \left[\frac{618}{500}(1-x) + \frac{10861}{5000}\right]\left(x - \frac{1}{2}\right) + \frac{6529}{10000},$$

thus, $q_2(\pi/4) \approx 1.3485$ and $q_2(\pi/2) \approx 2.2234$.

6. The points

| $x$ | $-2$ | $1$ | $4$ | $-1$ | $3$ | $-4$ |
|---|---|---|---|---|---|---|
| $y$ | $-1$ | $2$ | $59$ | $4$ | $24$ | $-53$ |

lie on a polynomial. Use the divided difference table of Newton's method to determine the degree of the polynomial.

Arranging the data points in increasing $x$-value:

| $x_i$ | $y_i = [y_i]$ | $[y_i, y_{i+1}]$ | $[y_i, y_{i+1}, y_{i+2}]$ | $[y_i, y_{i+1}, y_{i+2}, y_{i+3}]$ |
|---|---|---|---|---|
| $-4$ | $-53$ | | | |
| | | $26$ | | |
| $-2$ | $-1$ | | $-7$ | |
| | | $5$ | | $1$ |
| $-1$ | $4$ | | $-2$ | |
| | | $-1$ | | $1$ |
| $1$ | $2$ | | $3$ | |
| | | $11$ | | $1$ |
| $3$ | $24$ | | $8$ | |
| | | $35$ | | |
| $4$ | $59$ | | | |

with all remaining entries zero. This indicates that the polynomial is cubic, given by $([(x+1)-7](x+2)+26)(x+4)-53 = [(x-6)(x+2)+26](x+4)-53 = [x^2-4x+14](x+4)-53 = x^3-2x+3$.

7. Use Newton's method to find the polynomial that fits the following points:

| $x$ | $-3$ | $2$ | $-1$ | $3$ | $1$ |
|---|---|---|---|---|---|
| $y$ | $0$ | $5$ | $-4$ | $12$ | $0$ |

Arranging the data points in increasing $x$-value:

| $x_i$ | $y_i = [y_i]$ | $[y_i, y_{i+1}]$ | $[y_i, y_{i+1}, y_{i+2}]$ |
|---|---|---|---|
| $-3$ | $0$ | | |
| | | $-2$ | |
| $-1$ | $-4$ | | $1$ |
| | | $2$ | |
| $1$ | $0$ | | $1$ |
| | | $5$ | |
| $2$ | $5$ | | $1$ |
| | | $7$ | |
| $3$ | $12$ | | |

with all remaining entries zero. This indicates that the polynomial is quadratic, given by $[(x+1)-2](x+3) = (x-1)(x+3) = x^2+2x-3 = (x+1)^2-4$.

18. The table shows the drag coefficient $c_D$ of a sphere as a function of Reynold's number $Re$. Use a polynomial interpolant intersecting four nearest-neighbor data points to find $c_D$ at $Re = 5$, $50$, $500$ and $5000$.

| $Re$ | 0.2 | 2 | 20 | 200 | 2000 | 20000 |
|---|---|---|---|---|---|---|
| $c_D$ | 103 | 13.9 | 2.72 | 0.800 | 0.401 | 0.433 |

For $Re^* = 5$, using the four nearest neighbors, and Neville's method, if $d_i = Re_i - Re^*$,

| $d_i$ | $Re_i$ | $c_i = P[c_i](Re^*)$ | $P[c_i, c_{i+1}](Re^*)$ | $P[c_i, c_{i+1}, c_{i+2}](Re^*)$ | $P[c_0, c_1, c_2, c_3](Re^*)$ |
|---|---|---|---|---|---|
| $-4.8$ | $0.2$ | $103$ | | | |
| | | | $-\frac{673}{5}$ | | |
| $-3$ | $2$ | $13.9$ | | $-\frac{245153}{2475}$ | |
| | | | $\frac{3611}{300}$ | | $-\frac{79439146}{824175}$ |
| $15$ | $20$ | $2.72$ | | $\frac{235579}{19800}$ | |
| | | | $\frac{72}{25}$ | | |
| $195$ | $200$ | $0.8$ | | | |

Thus the interpolated $c_D^* \approx -96.38626$.

Using the previous algorithm, and four nearest-neighbor data points, the iterates are

| $Re$ | $c_D$ |
|---|---|
| $5$ | $-96.3863$ |
| $50$ | $-11.141$ |
| $500$ | $-1.58346$ |
| $5000$ | $56.8063$ |

**Source**: Jaan Kiusalaas, *Numerical Methods in Engineering with Python 3*, Problem Set 3.1

Given data points $\{(x_i, y_i)\}_{i=0}^n$ with $x_0 < x_1 < \cdots < x_n$, the natural cubic spline interpolant $S_n(x)$ is the piecewise interpolant that is cubic on each interval $I_i = [x_{i-1}, x_i]$ with a continuous second derivative on $[x_0, x_n]$—that is, $S_n(x_i) = y_i$, $S_n'(x_i)$ and $S_n''(x_i)$ exist for $0 \le i \le n$—with $S_n''(x_0) = S_n''(x_n) = 0$ (the natural condition).

If $S_n''(x_i) = y_i''$, for $0 \le i \le n$, the simplest interpolant of $S_n''$ on $I_i$ is

$$S_n''(x) = \frac{y_i''(x - x_{i-1}) + y_{i-1}''(x_i - x)}{x_i - x_{i-1}}, \qquad x_{i-1} \le x \le x_i,$$

the linear interpolant of $(x_{i-1}, y_{i-1}'')$ and $(x_i, y_i'')$.

Integrating the interpolant twice gives the cubic interpolant

$$S_n(x) = \frac{y_i''(x - x_{i-1})^3 + y_{i-1}''(x_i - x)^3}{6(x_i - x_{i-1})} + A(x - x_{i-1}) + B(x_i - x), \qquad x_{i-1} \le x \le x_i, \tag{1}$$

where $A$ and $B$ arise from the integration constants. Having $S_n(x)$ interpolate the data points:

$$y_{i-1} = S_n(x_{i-1}) = \frac{y_{i-1}''(x_i - x_{i-1})^3}{6(x_i - x_{i-1})} + B(x_i - x_{i-1}) \qquad \Longrightarrow B = \frac{y_{i-1}}{x_i - x_{i-1}} - \frac{y_{i-1}''}{6}(x_i - x_{i-1}), \tag{2}$$

$$y_i = S_n(x_i) = \frac{y_i''(x_i - x_{i-1})^3}{6(x_i - x_{i-1})} + A(x_i - x_{i-1}) \qquad \Longrightarrow A = \frac{y_i}{x_i - x_{i-1}} - \frac{y_i''}{6}(x_i - x_{i-1}). \tag{3}$$

So, placing $A$ and $B$ into (1) gives, for $x_{i-1} \le x \le x_i$,

$$S_n(x) = \frac{y_i''(x - x_{i-1})[(x - x_{i-1})^2 - (x_i - x_{i-1})^2]}{6(x_i - x_{i-1})} + \frac{y_{i-1}''(x_i - x)[(x_i - x)^2 - (x_i - x_{i-1})^2]}{6(x_i - x_{i-1})} + \frac{y_i(x - x_{i-1}) + y_{i-1}(x_i - x)}{x_i - x_{i-1}}.$$

To determine the values $y_i''$, consider the first derivative at $x_i$, $0 < i < n$:

$$S_n'(x) = \begin{cases} \dfrac{y_i''[3(x - x_{i-1})^2 - (x_i - x_{i-1})^2]}{6(x_i - x_{i-1})} - \dfrac{y_{i-1}''[3(x_i - x)^2 - (x_i - x_{i-1})^2]}{6(x_i - x_{i-1})} + \dfrac{y_i - y_{i-1}}{x_i - x_{i-1}}, & x_{i-1} \le x \le x_i, \\ \dfrac{y_{i+1}''[3(x - x_i)^2 - (x_{i+1} - x_i)^2]}{6(x_{i+1} - x_i)} - \dfrac{y_i''[3(x_{i+1} - x)^2 - (x_{i+1} - x_i)^2]}{6(x_{i+1} - x_i)} + \dfrac{y_{i+1} - y_i}{x_{i+1} - x_i}, & x_i \le x \le x_{i+1}. \end{cases}$$

Thus, for $S_n'(x_i)$ to exist,

$$S_n'(x_i) = \frac{2y_i''(x_i - x_{i-1})^2}{6(x_i - x_{i-1})} + \frac{y_{i-1}''(x_i - x_{i-1})^2}{6(x_i - x_{i-1})} + \frac{y_i - y_{i-1}}{x_i - x_{i-1}} = -\frac{y_{i+1}''(x_{i+1} - x_i)^2}{6(x_{i+1} - x_i)} - \frac{2y_i''(x_{i+1} - x_i)^2}{6(x_{i+1} - x_i)} + \frac{y_{i+1} - y_i}{x_{i+1} - x_i}$$

gives

$$y_{i+1}''(x_{i+1} - x_i) + 2y_i''(x_{i+1} - x_{i-1}) + y_{i-1}''(x_i - x_{i-1}) = 6\left[\frac{y_{i+1} - y_i}{x_{i+1} - x_i} - \frac{y_i - y_{i-1}}{x_i - x_{i-1}}\right]. \tag{4}$$

Since $y_0'' = y_n'' = 0$, this gives $n - 1$ equations in the $n - 1$ unknowns $y_i''$, $0 < i < n$, and the equations (1), (2), (3) and (4) together give the natural cubic interpolant $S_n$ on $[x_0, x_n]$.

**Example**. Using (1), (2), (3) and (4), determine the natural cubic spline interpolant on the data

| $x$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $y$ | 0 | 1 | 0 | 1 | 0 |

Note that $x_i = i + 1$, for $0 \le i \le 4$, and $h = x_i - x_{i-1} = 1$, for $1 \le i \le 4$, which simplifies (1), (2) and (3) to, for $i - 1 \le x \le i$,

$$S_4(x) = \frac{y_i''(x - i)[(x - i)^2 - 1]}{6} + \frac{y_{i-1}''(i + 1 - x)[(i + 1 - x)^2 - 1]}{6} + [y_i(x - i) + y_{i-1}(i + 1 - x)],$$

and simplifies (4) to

$$y''_{i+1} + 4y''_i + y''_{i-1} = 6[y_{i+1} - 2y_i + y_{i-1}] \quad 1 \le i \le 3.$$

The latter gives the system

$$\begin{cases} \qquad y''_2 + 4y''_1 & = 6[-2] = -12, \\ y''_3 + 4y''_2 + \ \ y''_1 & = 6[2] = 12, \\ 4y''_3 + \ \ y''_2 & = 6[-2] = -12. \end{cases} \implies \begin{cases} y''_1 & = -\dfrac{30}{7}, \\ y''_2 & = \dfrac{36}{7}, \\ y''_3 & = -\dfrac{30}{7}. \end{cases}$$

These give the natural cubic interpolant

$$S_4(x) = \begin{cases} -\dfrac{5(x-1)[(x-1)^2-1]}{7} + x - 1, & 1 \le x \le 2 \\ \dfrac{6(x-2)[(x-2)^2-1]}{7} - \dfrac{5(3-x)[(3-x)^2-1]}{7} - x + 3, & 2 \le x \le 3 \\ -\dfrac{5(x-3)[(x-3)^2-1]}{7} + \dfrac{6(4-x)[(4-x)^2-1]}{7} + x - 3, & 3 \le x \le 4 \\ -\dfrac{5(5-x)[(5-x)^2-1]}{7} - x + 5, & 4 \le x \le 5 \end{cases}$$

To show that $S_4$ interpolates the data points:

$$\begin{aligned} S_4(1) &= -\frac{5(1-1)[(1-1)^2-1]}{7} + 1 - 1 = 0, \\ S_4(2) &= -\frac{5(2-1)[(2-1)^2-1]}{7} + 2 - 1 = \frac{6(2-2)[(2-2)^2-1]}{7} - \frac{5(3-2)[(3-2)^2-1]}{7} - 2 + 3 = 1, \\ S_4(3) &= \frac{6(3-2)[(3-2)^2-1]}{7} - \frac{5(3-3)[(3-3)^2-1]}{7} - 3 + 3 = 0, \\ &= -\frac{5(3-3)[(3-3)^2-1]}{7} + \frac{6(4-3)[(4-3)^2-1]}{7} + 3 - 3 = 0, \\ S_4(4) &= -\frac{5(4-3)[(4-3)^2-1]}{7} + \frac{6(4-4)[(4-4)^2-1]}{7} + 4 - 3 = -\frac{5(5-4)[(5-4)^2-1]}{7} - 4 + 5 = 1, \\ S_4(5) &= -\frac{5(5-5)[(5-5)^2-1]}{7} - 5 + 5 = 0. \end{aligned}$$

The derivative of the interpolant is

$$S'_4(x) = \begin{cases} -\dfrac{15(x-1)^2}{7} + \dfrac{12}{7}, & 1 \le x \le 2 \\ \dfrac{18(x-2)^2}{7} + \dfrac{15(3-x)^2}{7} - \dfrac{18}{7}, & 2 \le x \le 3 \\ -\dfrac{15(x-3)^2}{7} - \dfrac{18(4-x)^2}{7} + \dfrac{18}{7}, & 3 \le x \le 4 \\ \dfrac{15(5-x)^2}{7} - \dfrac{12}{7}, & 4 \le x \le 5 \end{cases}$$

To show that $S'_4$ is continuous on $[1, 5]$

$$\begin{aligned} S'_4(2) &= -\frac{15(2-1)^2}{7} + \frac{12}{7} = \frac{18(2-2)^2}{7} + \frac{15(3-2)^2}{7} - \frac{18}{7} = -\frac{3}{7}, \\ S'_4(3) &= \frac{18(3-2)^2}{7} + \frac{15(3-3)^2}{7} - \frac{18}{7} = -\frac{15(3-3)^2}{7} - \frac{18(4-3)^2}{7} + \frac{18}{7} = 0, \\ S'_4(4) &= -\frac{15(4-3)^2}{7} - \frac{18(4-4)^2}{7} + \frac{18}{7} = \frac{15(5-4)^2}{7} - \frac{12}{7} = \frac{3}{7}. \end{aligned}$$

The second derivative of the interpolant is

$$S_4''(x) = \begin{cases} -\dfrac{30(x-1)}{7}, & 1 \le x \le 2 \\ \dfrac{36(x-2)}{7} - \dfrac{30(3-x)}{7}, & 2 \le x \le 3 \\ -\dfrac{30(x-3)}{7} + \dfrac{36(4-x)}{7}, & 3 \le x \le 4 \\ -\dfrac{30(5-x)}{7}, & 4 \le x \le 5 \end{cases}$$

To show that $S_4''$ is continuous on $[1, 5]$,

$$\begin{aligned}
S_4''(1) &= -\frac{30(1-1)}{7} = 0 = y_0'' \\
S_4''(2) &= -\frac{30(2-1)}{7} = \frac{36(2-2)}{7} - \frac{30(3-2)}{7} = -\frac{30}{7} = y_1'', \\
S_4''(3) &= \frac{36(3-2)}{7} - \frac{30(3-3)}{7} = -\frac{30(3-3)}{7} + \frac{36(4-3)}{7} = \frac{36}{7} = y_2'', \\
S_4''(4) &= -\frac{30(4-3)}{7} + \frac{36(4-4)}{7} = -\frac{30(5-4)}{7} = -\frac{30}{7} = y_3'', \\
S_4''(5) &= -\frac{30(5-x)}{7} = 0 = y_4''.
\end{aligned}$$

**Source**: Jaan Kiusalaas, *Numerical Methods in Engineering with Python 3*, Chapter 3

Name: ______________________________

10. Determine the natural cubic spline that passes through the data points

| $x$ | 0 | 1 | 2 |
|---|---|---|---|
| $y$ | 0 | 2 | 1 |

Note that the interpolant consists of two cubics, one valid in $0 \leq x \leq 1$, the other in $1 \leq x \leq 2$. Verify that these cubics have the same first and second derivatives at $x = 1$.

Note that $\Delta x = x_i - x_{i-1} = 1$. The splines are, plugging in the data,

$$\begin{aligned} s_1(x) &= \frac{y_0''}{6}(x-1)[1-(x-1)^2] + \frac{y_1''}{6}x(x^2-1) + 2x, \\ s_2(x) &= \frac{y_1''}{6}(x-2)[1-(x-2)^2] + \frac{y_2''}{6}(1-x)[1-(1-x)^2] - x + 3, \end{aligned}$$

where $y_0'' = y_2'' = 0$ and $-4y_1'' = 6\left(\frac{0-2}{0-1} - \frac{2-1}{1-2}\right) = 18$, so $y_1'' = -\frac{9}{2}$ and

$$\begin{aligned} s_1(x) &= -\frac{3}{4}x^3 + \frac{11}{4}x & &\Longrightarrow s_1'(x) = -\frac{9}{4}x^2 + \frac{11}{4} & &\Longrightarrow s_1''(x) = -\frac{9}{2}x, \\ s_2(x) &= \frac{3}{4}x^3 - \frac{9}{2}x^2 + \frac{29}{4}x - \frac{3}{2} & &\Longrightarrow s_2'(x) = \frac{9}{4}x^2 - 9x + \frac{29}{4} & &\Longrightarrow s_2''(x) = \frac{9}{2}x - 9. \end{aligned}$$

To confirm: $s_1(0) = 0$, $s_1(1) = s_2(1) = 2$, $s_2(2) = 1$, $s_1'(1) = s_2'(1) = \frac{1}{2}$, $s_1''(0) = 0$, $s_1''(1) = s_2''(1) = -\frac{9}{2}$, $s_2''(2) = 0$.

11. Given the data points

| $x$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $y$ | 13 | 15 | 12 | 9 | 13 |

determine the natural cubic spline interpolant at $x = 3.4$.

Note that $\Delta x = x_i - x_{i-1} = 1$. The splines are, plugging in the data,

$$\begin{aligned} s_1(x) &= \frac{y_0''}{6}(x-2)[1-(x-2)^2] + \frac{y_1''}{6}(1-x)[1-(1-x)^2] + 2x + 11, \\ s_2(x) &= \frac{y_1''}{6}(x-3)[1-(x-3)^2] + \frac{y_2''}{6}(2-x)[1-(2-x)^2] - 3x + 21, \\ s_3(x) &= \frac{y_2''}{6}(x-4)[1-(x-4)^2] + \frac{y_3''}{6}(3-x)[1-(3-x)^2] - 3x + 21, \\ s_4(x) &= \frac{y_3''}{6}(x-5)[1-(x-5)^2] + \frac{y_4''}{6}(4-x)[1-(4-x)^2] + 4x - 7, \end{aligned}$$

where $y_0'' = y_4'' = 0$ and

$$\begin{cases} 4y_1'' + y_2'' & = 6(13 - 2(15) + 12) = -30, \\ y_1'' + 4y_2'' + y_3'' & = 6(15 - 2(12) + 9) = 0, \\ y_2'' + 4y_3'' & = 6(12 - 2(9) + 13) = 42. \end{cases} \Longrightarrow \begin{cases} y_1'' & = -\frac{51}{7}, \\ y_2'' & = -\frac{6}{7}, \\ y_3'' & = \frac{75}{7}. \end{cases}$$

So,

$$s_3(3.4) = -\frac{1}{7}\left(-\frac{3}{5}\right)\frac{16}{25} + \frac{25}{14}\left(-\frac{2}{5}\right)\frac{21}{25} - \frac{51}{5} + 21 = \frac{8973}{875} \approx 10.25.$$

12. Compute the zero of the function $y(x)$ from the following data:

| $x$ | 0.2 | 0.4 | 0.6 | 0.8 | 1 |
|---|---|---|---|---|---|
| $y$ | 1.150 | 0.855 | 0.377 | −0.266 | −1.049 |

Use inverse interpolation with the natural cubic spline. *Hint*: Reorder the data so that the values of $y$ are in ascending order.

The inverted data, as hinted, looks like:

| $y$ | −1.049 | −0.266 | 0.377 | 0.855 | 1.150 |
|---|---|---|---|---|---|
| $x$ | 1 | 0.8 | 0.6 | 0.4 | 0.2 |

The splines are, plugging in the data,

$$\begin{aligned}
s_1(y) &= -\frac{x_0''}{6}\left[\frac{(y+0.266)^3}{0.783}-(y+0.266)(0.783)\right]+\frac{x_1''}{6}\left[\frac{(y+1.049)^3}{0.783}-(y+1.049)(0.783)\right]-\frac{y-2.866}{3.915},\\
s_2(y) &= -\frac{x_1''}{6}\left[\frac{(y-0.377)^3}{0.643}-(y-0.377)(0.643)\right]+\frac{x_2''}{6}\left[\frac{(y+0.266)^3}{0.643}-(y+0.266)(0.643)\right]-\frac{y-2.306}{3.215},\\
s_3(y) &= -\frac{x_2''}{6}\left[\frac{(y-0.855)^3}{0.478}-(y-0.855)(0.478)\right]+\frac{x_3''}{6}\left[\frac{(y-0.377)^3}{0.478}-(y-0.377)(0.478)\right]-\frac{y-1.811}{2.39},\\
s_4(y) &= -\frac{x_3''}{6}\left[\frac{(y-1.15)^3}{0.295}-(y-1.15)(0.295)\right]+\frac{x_4''}{6}\left[\frac{(y-0.855)^3}{0.295}-(y-0.855)(0.295)\right]-\frac{y-1.445}{1.475},
\end{aligned}$$

where $x_0'' = x_4'' = 0$ and

$$\begin{cases}
2852x_1''+643x_2'' & = 6000\left(\dfrac{200}{783}-\dfrac{200}{643}\right) = -\dfrac{56000000}{167823},\\
643x_1''+2242x_2''+478x_3'' & = 6000\left(\dfrac{200}{643}-\dfrac{200}{478}\right) = -\dfrac{99000000}{153677},\\
478x_2''+1546x_3'' & = 6000\left(\dfrac{200}{478}-\dfrac{200}{295}\right) = -\dfrac{21960000}{14101}.
\end{cases}
\implies
\begin{cases}
x_1'' & \approx -0.10688,\\
x_2'' & \approx -0.044875,\\
x_3'' & \approx -0.99346.
\end{cases}$$

So, the estimate of the root by the inverse cubic spline interpolation is

$$s_2(0) = \frac{x_1''[1885(0.377^2-0.643^2)]+x_2''[1330(0.266^2-0.643^2)]+13836}{19290} \approx 0.72116.$$

13. Use a cubic spline that has constant second derivatives within its first and last segments (the end segments are parabolic) to determine the cubic spline interpolant at $x = 2.6$, given the data points

| $x$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| $y$ | 1 | 1 | 0.5 | 0 |

The end conditions for this spline are $y''(x_0) = y''(x_1)$ and $y''(x_{n-1}) = y''(x_n)$.

Note that $\Delta x = x_i - x_{i-1} = 1$. The splines are, plugging in the data,

$$\begin{aligned}
s_1(x) &= \frac{y_0''}{6}(x-1)[1-(x-1)^2]+\frac{y_1''}{6}(-x)(1-x^2)+1,\\
s_2(x) &= \frac{y_1''}{6}(x-2)[1-(x-2)^2]+\frac{y_2''}{6}(1-x)[1-(1-x)^2]-\frac{x}{2}+\frac{3}{2},\\
s_3(x) &= \frac{y_2''}{6}(x-3)[1-(x-3)^2]+\frac{y_3''}{6}(2-x)[1-(2-x)^2]-\frac{x}{2}+\frac{3}{2},
\end{aligned}$$

with $y_0''+4y_1''+y_2'' = 5y_1''+y_2'' = 6(1-2(1)+0.5) = -3$ and $y_1''+4y_2''+y_3'' = y_1''+5y_2'' = 6(1-2(0.5)+0) = 0$, due to the end conditions. These give $y_1'' = -\frac{5}{8}$ and $y_2'' = \frac{1}{8}$, hence

$$s(2.6) = s_3(2.6) = \frac{(2.6-3)[1-(2.6-3)^2]+(2-2.6)[1-(2-2.6)^2]}{48}-\frac{2.6}{2}+\frac{3}{2} = \frac{37}{200} = 0.185.$$

17. The table shows the drag coefficient $c_D$ of a sphere as a function of Reynold's number $Re$. Use a natural cubic spline to find $c_D$ at $Re = 5$, 50, 500, and 5000. *Hint*: Use a log-log scale.

| $Re$ | 0.2 | 2 | 20 | 200 | 2000 | 20000 |
|---|---|---|---|---|---|---|
| $c_D$ | 103 | 13.9 | 2.72 | 0.800 | 0.401 | 0.433 |

From the hint,

| $x = \log Re$ | $\log 2 - 1$ | $\log 2$ | $1 + \log 2$ | $2 + \log 2$ | $3 + \log 2$ | $4 + \log 2$ |
|---|---|---|---|---|---|---|
| $y = \log c_D$ | $2 + \log 1.03$ | $1 + \log 1.39$ | $\log 2.72$ | $\log 8 - 1$ | $\log 4.01 - 1$ | $\log 4.33 - 1$ |

Note that $\Delta x = x_i - x_{i-1} = 1$. The splines are, plugging in the data,

$$\begin{aligned}
s_1(x) &= \frac{y_0''}{6}(x - \log 2)[1 - (x - \log 2)^2] - \frac{y_1''}{6}(x - \log 2 + 1)[1 - (x - \log 2 + 1)^2] \\
&\quad + (1 + \log 1.39)(x - \log 2 + 1) - (2 + \log 1.03)(x - \log 2), \\
s_2(x) &= \frac{y_1''}{6}(x - \log 2 - 1)[1 - (x - \log 2 - 1)^2] - \frac{y_2''}{6}(x - \log 2)[1 - (x - \log 2)^2] \\
&\quad + (\log 2.72)(x - \log 2) - (1 + \log 1.39)(x - \log 2 - 1), \\
s_3(x) &= \frac{y_2''}{6}(x - \log 2 - 2)[1 - (x - \log 2 - 2)^2] - \frac{y_3''}{6}(x - \log 2 - 1)[1 - (x - \log 2 - 1)^2] \\
&\quad + (\log 8 - 1)(x - \log 2 - 1) - (\log 2.72)(x - \log 2 - 2), \\
s_4(x) &= \frac{y_3''}{6}(x - \log 2 - 3)[1 - (x - \log 2 - 3)^2] - \frac{y_4''}{6}(x - \log 2 - 2)[1 - (x - \log 2 - 2)^2] \\
&\quad + (\log 4.01 - 1)(x - \log 2 - 2) - (\log 8 - 1)(x - \log 2 - 3), \\
s_5(x) &= \frac{y_4''}{6}(x - \log 2 - 4)[1 - (x - \log 2 - 4)^2] - \frac{y_5''}{6}(x - \log 2 - 3)[1 - (x - \log 2 - 3)^2] \\
&\quad + (\log 4.33 - 1)(x - \log 2 - 3) - (\log 4.01 - 1)(x - \log 2 - 4),
\end{aligned}$$

where $y_0'' = y_5'' = 0$ and

$$\begin{aligned}
4y_1'' + y_2'' &= 6(2 + \log 1.03 - 2(1 + \log 1.39) + \log 2.72) = \log[(28016/19321)^6], \\
y_1'' + 4y_2'' + y_3'' &= 6(1 + \log 1.39 - 2\log 2.72 + \log 8 - 1) = \log[(3475/2312)^6], \\
y_2'' + 4y_3'' + y_4'' &= 6(\log 2.72 - 2(\log 8 - 1) + \log 4.01 - 1) = \log[(6817/4000)^6], \\
y_3'' + 4y_4'' &= 6(\log 8 - 1 - 2(\log 4.01 - 1) + \log 4.33 - 1) = \log[(346400/160801)^6].
\end{aligned}$$

giving

$$\begin{aligned}
\frac{y_1''}{6} &= \log \sqrt[209]{\frac{160801 \cdot 6817^4 \cdot 2312^{15} \cdot 28016^{56}}{346400 \cdot 4000^4 \cdot 3475^{15} \cdot 19321^{56}}} \approx 0.058777, \\
\frac{y_2''}{6} &= \log \sqrt[209]{\frac{346400^4 \cdot 19321^{15} \cdot 4000^{16} \cdot 3475^{60}}{160801^4 \cdot 28016^{15} \cdot 6817^{16} \cdot 2312^{60}}} \approx 0.027876, \\
\frac{y_3''}{6} &= \log \sqrt[209]{\frac{28016^4 \cdot 160801^{15} \cdot 2312^{16} \cdot 6817^{60}}{19321^4 \cdot 346400^{15} \cdot 3475^{16} \cdot 4000^{60}}} \approx 0.032089, \\
\frac{y_4''}{6} &= \log \sqrt[209]{\frac{19321 \cdot 3475^4 \cdot 4000^{15} \cdot 346400^{56}}{28016 \cdot 2312^4 \cdot 6817^{15} \cdot 160801^{56}}} \approx 0.0753.
\end{aligned}$$

For $Re = 5$, $x = \log 5$ and

$$s_2(\log 5) = \frac{y_1''}{6}[(\log 4)^3 - \log 4] + \frac{y_2''}{6}[(\log 2.5)^3 - \log 2.5] - [\log(136/695)](\log 2.5) + \log 2.72,$$

and estimates $c_D$ as $10^{s_2(\log 5)} \approx 6.7484$. For $Re = 50$, $x = 1 + \log 5$ and

$$s_3(1 + \log 5) = \frac{y_2''}{6}[(\log 4)^3 - \log 4] + \frac{y_3''}{6}[(\log 2.5)^3 - \log 2.5] - [\log(5/17)](\log 4) - 1 + \log 8,$$

and estimates $c_D$ as $10^{s_3(1+\log 5)} \approx 1.5908$. For $Re = 500$, $x = 2 + \log 5$ and

$$s_4(2 + \log 5) = \frac{y_3''}{6}[(\log 4)^3 - \log 4] + \frac{y_4''}{6}[(\log 2.5)^3 - \log 2.5] - [\log(401/800)](\log 4) - 1 + \log 4.01,$$

and estimates $c_D$ as $10^{s_4(2+\log 5)} \approx 0.55743$. For $Re = 5000$, $x = 3 + \log 5$ and

$$s_5(3 + \log 5) = \frac{y_4''}{6}[(\log 4)^3 - \log 4] - [\log(433/401)](\log 4) - 1 + \log 4.33,$$

and estimates $c_D$ as $10^{s_5(3+\log 5)} \approx 0.58627$.

**Source**: Jaan Kiusalaas, *Numerical Methods in Engineering with Python 3*, Problem Set 3.1

Given data points $\{(x_i, y_i)\}_{i=0}^{n}$, expected to have some noise, with underlying function $f(x) = \sum\limits_{k=0}^{m} a_k f_k(x)$, the best-fit function by least squares is the function $f^*(x) = \sum\limits_{k=0}^{m} a_k^* f_k(x)$ such that

$$S(a_0^*, a_1^*, \ldots, a_m^*) = \sum_{i=0}^{n} \left( y_i - \sum_{k=0}^{m} a_k^* f_k(x_i) \right)^2$$

is minimized.

From multivariate calculus, the minimum of the function $S$ in the variables $a_k^*$, $0 \le k \le m$ occurs only on points where

$$\frac{\partial S}{\partial a_k^*} = -2 \sum_{i=0}^{n} f_k(x_i) \left( y_i - \sum_{k=0}^{m} a_k^* f_k(x_i) \right) = 0, \qquad 0 \le k \le m.$$

This leads to a system of equations

$$\begin{bmatrix} \sum\limits_{i=0}^{n} [f_0(x_i)]^2 & \sum\limits_{i=0}^{n} f_0(x_i) f_1(x_i) & \cdots & \sum\limits_{i=0}^{n} f_0(x_i) f_m(x_i) \\ \sum\limits_{i=0}^{n} f_0(x_i) f_1(x_i) & \sum\limits_{i=0}^{n} [f_1(x_i)]^2 & \cdots & \sum\limits_{i=0}^{n} f_1(x_i) f_m(x_i) \\ \vdots & \vdots & \ddots & \vdots \\ \sum\limits_{i=0}^{n} f_0(x_i) f_m(x_i) & \sum\limits_{i=0}^{n} f_1(x_i) f_m(x_i) & \cdots & \sum\limits_{i=0}^{n} [f_m(x_i)]^2 \end{bmatrix} \begin{bmatrix} a_0^* \\ a_1^* \\ \vdots \\ a_m^* \end{bmatrix} = \begin{bmatrix} \sum\limits_{i=0}^{n} y_i f_0(x_i) \\ \sum\limits_{i=0}^{n} y_i f_1(x_i) \\ \vdots \\ \sum\limits_{i=0}^{n} y_i f_m(x_i) \end{bmatrix} \qquad (\star)$$

For linear regression, where $f(x) = ax + b$, i.e., $f_k(x) = x^k$, $a = a_1^*$ and $b = a_0^*$, the system is

$$\begin{cases} a \sum\limits_{i=0}^{n} x_i^2 + b \sum\limits_{i=0}^{n} x_i &= \sum\limits_{i=0}^{n} x_i y_i \\ a \sum\limits_{i=0}^{n} x_i + b(n+1) &= \sum\limits_{i=0}^{n} y_i \end{cases} \implies \begin{cases} a &= \dfrac{\sum\limits_{i=0}^{n} x_i (y_i - \bar{y})}{\sum\limits_{i=0}^{n} x_i (x_i - \bar{x})} \\ b &= \bar{y} - a\bar{x} \end{cases} \text{ where } \bar{x} = \frac{1}{n+1} \sum_{i=0}^{n} x_i, \bar{y} = \frac{1}{n+1} \sum_{i=0}^{n} y_i.$$

Note that

$$\begin{aligned} \sum_{i=0}^{n} x_i (y_i - \bar{y}) &= \sum_{i=0}^{n} x_i \left( y_i - \frac{1}{n+1} \sum_{j=0}^{n} y_j \right) = \sum_{i=0}^{n} \left( x_i y_i - \frac{1}{n+1} \sum_{j=0}^{n} x_i y_j \right) \\ &= \sum_{i=0}^{n} x_i y_i - \frac{1}{n+1} \sum_{i=0}^{n} \sum_{j=0}^{n} x_i y_j = \sum_{j=0}^{n} x_j y_j - \frac{1}{n+1} \sum_{j=0}^{n} \sum_{i=0}^{n} x_i y_j \\ &= \sum_{j=0}^{n} \left( x_j y_j - \frac{1}{n+1} \sum_{i=0}^{n} x_i y_j \right) = \sum_{j=0}^{n} y_j \left( x_j - \frac{1}{n+1} \sum_{i=0}^{n} x_i \right) = \sum_{j=0}^{n} y_j (x_j - \bar{x}). \end{aligned}$$

From the view of linear algebra, consider that, for the best-fit curve $f(x) = \sum\limits_{k=0}^{m} a_k^* f_k(x)$, for each data point $(x_i, y_i)$, there is an error $\omega_i = y_i - f(x_i)$, which gives rise to a system (which is overdetermined) $C\vec{a} + \vec{w} = \vec{y}$, where $C = [c_{ij} = f_j(x_i)]$, $\vec{a} = \langle a_0^*, \ldots, a_m^* \rangle^{\mathrm{T}}$, $\vec{w} = \langle \omega_0, \ldots, \omega_n \rangle^{\mathrm{T}}$ and $\vec{y} = \langle y_0, \ldots, y_n \rangle^{\mathrm{T}}$. Pre-multiplying the vector equation by $C^{\mathrm{T}}$ gives $C^{\mathrm{T}} C\vec{a} + C^{\mathrm{T}} \vec{w} = C^{\mathrm{T}} \vec{y}$—the matrix equation $(\star)$ is $C^{\mathrm{T}} C\vec{a} = C^{\mathrm{T}} \vec{y}$, which indicates that $\vec{w}$ is in the nullspace of $C^{\mathrm{T}}$ and $C^{\mathrm{T}} \vec{y}$ is in the column space of $C$, and is thus the projection of $\vec{y}$ to the column space of $C$, $C(C^{\mathrm{T}} C)^{-1} C^{\mathrm{T}} \vec{y} = C\vec{a} = \vec{y} - \vec{w}$.

**Source**: Jaan Kiusalaas, *Numerical Methods in Engineering with Python 3*

Name: ____________________________________________

3. Three tensile tests were carried out on an aluminum bar, In each test, the strain was measured at the same values of stress. The results were

| Stress (MPa) | 34.5 | 69.0 | 103.5 | 138.0 |
|---|---|---|---|---|
| Strain (Test 1) | 0.46 | 0.95 | 1.48 | 1.93 |
| Strain (Test 2) | 0.34 | 1.02 | 1.51 | 2.09 |
| Strain (Test 3) | 0.73 | 1.10 | 1.62 | 2.12 |

where the units of strain are mm / m. Use linear regression to estimate the modulus of elasticity of the bar (modulus of elasticity = stress / strain).

The least-squares best fit line for the $n = 12$ data points is $y = a + bx$, where

$$\bar{x} = \frac{\sum x_i}{n} = \frac{3(34.5) + 3(69.0) + 3(103.5) + 3(138.0)}{12} = \frac{345}{4}$$
$$\bar{y} = \frac{\sum y_i}{n} = \frac{0.46 + 0.95 + 1.48 + 1.93 + 0.34 + 1.02 + 1.51 + 2.09 + 0.73 + 1.10 + 1.62 + 2.12}{12} = \frac{307}{240}$$
$$b = \frac{\sum x_i(y_i - \bar{y})}{\sum x_i(x_i - \bar{x})} = \frac{35351}{2380500} \qquad \Longrightarrow \qquad a = \bar{y} - b\bar{x} = -\frac{1}{600}$$

So the modulus of elasticity is approximated by $\frac{1}{b} \times 10^6 \frac{\text{Pa} \cdot \text{m}}{\text{mm}} = \frac{23805}{35351} \times 10^8 \frac{\text{Pa} \cdot \text{m}}{\text{mm}} \approx 6.7339 \times 10^7 \frac{\text{Pa} \cdot \text{m}}{\text{mm}}$.

9. Fit a straight line and a quadratic to the data in the following table.

| $x$ | 1.0 | 2.5 | 3.5 | 4.0 | 1.1 | 1.8 | 2.2 | 3.7 |
|---|---|---|---|---|---|---|---|---|
| $y$ | 6.008 | 15.722 | 27.130 | 33.772 | 5.257 | 9.549 | 11.098 | 28.828 |

Which is a better fit?

The systems used for best-fit curves use the following values:

$$\sum_{i=0}^{7} x_i = 19.8, \qquad \sum_{i=0}^{7} x_i^2 = 58.4, \qquad \sum_{i=0}^{7} x_i^3 = 191.964, \qquad \sum_{i=0}^{7} x_i^4 = 668.9284,$$
$$\sum_{i=0}^{7} y_i = 137.364, \qquad \sum_{i=0}^{7} x_i y_i = 429.4061, \qquad \sum_{i=0}^{7} x_i^2 y_i = 1462.63437.$$

Solving the corresponding systems, the best-fit curves are approximately $y = 9.438544x - 6.189895$, with least square error approximately 30.20147, and $y = 2.108118x^2 - 1.068896x + 4.405674$, with least square error approximately 3.304259. Even without the computation, the best-first parabola cannot be worse then the best-fit line.

13. Determine $a$ and $b$ for which $f(x) = a\sin(\pi x/2) + b\cos(\pi x/2)$ fits the following data in the least-squares sense.

| $x$ | −0.50 | −0.19 | 0.02 | 0.20 | 0.35 | 0.50 |
|---|---|---|---|---|---|---|
| $y$ | −3.558 | −2.815 | −1.995 | −1.040 | −0.068 | 0.677 |

The least-squares error is given by $S = \sum[y_i - f(x_i)]^2$, which is minimized when

$$\begin{aligned}
\frac{\partial S}{\partial a} &= 0 = -2\sum \sin\left(\frac{\pi x_i}{2}\right)\left[y_i - a\sin\left(\frac{\pi x_i}{2}\right) - b\cos\left(\frac{\pi x_i}{2}\right)\right] \\
&\Longrightarrow 2\sum y_i \sin\left(\frac{\pi x_i}{2}\right) = a\sum[1 - \cos(\pi x_i)] + b\sum \sin(\pi x_i) \\
\frac{\partial S}{\partial b} &= 0 = -2\sum \cos\left(\frac{\pi x_i}{2}\right)\left[y_i - a\sin\left(\frac{\pi x_i}{2}\right) - b\cos\left(\frac{\pi x_i}{2}\right)\right] \\
&\Longrightarrow 2\sum y_i \cos\left(\frac{\pi x_i}{2}\right) = a\sum \sin(\pi x_i) + b\sum[1 + \cos(\pi x_i)]
\end{aligned}$$

Solving for $a$ and $b$ gives

$$a = 2\frac{\left[\sum y_i \sin\left(\frac{\pi x_i}{2}\right)\right]\left(\sum[1+\cos(\pi x_i)]\right) - \left[\sum y_i \cos\left(\frac{\pi x_i}{2}\right)\right]\left[\sum \sin(\pi x_i)\right]}{\left(\sum[1-\cos(\pi x_i)]\right)\left(\sum[1+\cos(\pi x_i)]\right) - \left[\sum \sin(\pi x_i)\right]^2},$$

$$b = 2\frac{\left[\sum y_i \cos\left(\frac{\pi x_i}{2}\right)\right]\left(\sum[1-\cos(\pi x_i)]\right) - \left[\sum y_i \sin\left(\frac{\pi x_i}{2}\right)\right]\left[\sum \sin(\pi x_i)\right]}{\left(\sum[1+\cos(\pi x_i)]\right)\left(\sum[1-\cos(\pi x_i)]\right) - \left[\sum \sin(\pi x_i)\right]^2}.$$

So, to get $f(x)$,

$$\sum \sin(\pi x_i) \approx 0.97950, \qquad \sum \cos(\pi x_i) \approx 3.0881 \implies \sum[1 \pm \cos(\pi x_i)] \approx 6 \pm 3.0881$$

$$\sum y_i \cos\left(\frac{\pi x_i}{2}\right) \approx -7.7688 \qquad \sum y_i \sin\left(\frac{\pi x_i}{2}\right) \approx 3.4027$$

$$a \approx 3.0218, \qquad b \approx -2.0354.$$

14. Determine $a$ and $b$ so that $f(x) = ax^b$ fits the following data in the least-squares sense.

| $x$ | 0.5 | 1.0 | 1.5 | 2.0 | 2.5 |
|---|---|---|---|---|---|
| $y$ | 0.49 | 1.60 | 3.36 | 6.44 | 10.16 |

*Hint*: Let $F(x) = \ln a + b \ln x$ be the least-squares best fit of $(\ln x_i, \ln y_i)$. Let the residuals be $r_i = y_i - f(x_i)$ and $R_i = \ln y_i - F(x_i)$. Then, assuming $r_i \ll y_i$, $R_i \approx r_i/y_i$.

The least-squares best fit line for the $n = 12$ log-log data points $(\ln x_i, \ln y_i)$ weighted by $y_i$ is $\ln y = \ln a + b \ln x$ where

$$\sum y_i^2 = \frac{1587889}{10000}$$

$$\widehat{\ln x} = \frac{\sum y_i^2 \ln x_i}{\sum y_i^2} \approx 0.80448 \qquad \widehat{\ln y} = \frac{\sum y_i^2 \ln y_i}{\sum y_i^2} \approx 2.0863$$

$$b = \frac{\sum y_i^2 (\ln x_i)(\ln y_i - \widehat{\ln y})}{\sum y_i^2 (\ln x_i)(\ln x_i - \widehat{\ln x})} \approx 2.0677 \qquad a = \exp\left(\widehat{\ln y} - b\widehat{\ln x}\right) \approx 1.5264$$

18. Linear regression can be extended to data that depend on two or more variables (called multiple linear regression). If the dependent variable is $z$ and independent variables are $x$ and $y$, the data to be fitted

| $x$ | $y$ | $z$ |
|---|---|---|
| 0 | 0 | 1.42 |
| 0 | 1 | 1.85 |
| 1 | 0 | 0.78 |
| 2 | 0 | 0.18 |
| 2 | 1 | 0.60 |
| 2 | 2 | 1.05 |

Instead of a straight line, the fitting function now represents a plane

$$z = a + bx + cy, \qquad \text{where} \qquad \begin{bmatrix} n & \sum x_i & \sum y_i \\ \sum x_i & \sum x_i^2 & \sum x_i y_i \\ \sum y_i & \sum x_i y_i & \sum y_i^2 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \sum z_i \\ \sum x_i z_i \\ \sum y_i z_i \end{bmatrix}.$$

Find the fitting plane for the given data.

The data is to be fit by least squares to a function $z = a + by + cx$. This indicates that the parameters $a$, $b$ and $c$ should be chosen such that

$$S(a, b, c) = \sum_{i=0}^{5} (z_i - a - bx_i - cy_i)^2$$

is minimized for the data set $\{(x_i, y_i, z_i)\}_{i=0}^{5}$ given above. Optimizing $S$ with respect to $a$, $b$ and $c$ leads to the system

$$\begin{aligned}
\frac{\partial S}{\partial a} &= -2\sum_{i=0}^{5}(z_i - a - bx_i - cy_i) &= 0 \Longrightarrow \quad \sum_{i=0}^{5} z_i &= 6a + b\sum_{i=0}^{5} x_i + c\sum_{i=0}^{5} y_i,\\
\frac{\partial S}{\partial b} &= -2\sum_{i=0}^{5}x_i(z_i - a - bx_i - cy_i) &= 0 \Longrightarrow \quad \sum_{i=0}^{5} x_i z_i &= a\sum_{i=0}^{5} x_i + b\sum_{i=0}^{5} x_i^2 + c\sum_{i=0}^{5} x_i y_i,\\
\frac{\partial S}{\partial c} &= -2\sum_{i=0}^{5}y_i(z_i - a - bx_i - cy_i) &= 0 \Longrightarrow \quad \sum_{i=0}^{5} y_i z_i &= a\sum_{i=0}^{5} y_i + b\sum_{i=0}^{5} x_i y_i + c\sum_{i=0}^{5} y_i^2.
\end{aligned}$$

The solution to the system is

$$a = \frac{6077}{4300}, \qquad b = -\frac{668}{1075}, \qquad c = \frac{3763}{8600} \Longrightarrow z \approx 1.4132558 - 0.621395x + 0.437558y.$$

**Source**: Jaan Kiusalaas, *Numerical Methods in Engineering with Python 3*, Problem Set 3.2

You are currently working with physicists that are using a particle collider for some experiments. They have been testing the monitoring equipment, by passing a single particle and monitoring its motion on a plane, where the particle is at position $(0, 0)$ at the starting time $t = 0$. (The units have been omitted, as the measurements are on such a small scale, and in order to anonymize the computations, somewhat.) They want to model the measured motion from the data for analysis within the duration of the measurements.

For this project, data is provided in a text file named `partmove.txt`, which has numbers at each line, separated by spaces within lines, and lines separated by carriage returns, and no other characters are included. The first line contains $N$, which is an integer from 1 to 16. The next $N$ lines contains three pairs of floating point numbers corresponding to the position $(x_i, y_i)$, the velocity $(x'_i, y'_i)$, and the acceleration $(x''_i, y''_i)$ at the $i$th measurement, where $0 < t_1 < t_2 < \cdots < t_N$ is the time at which the measurements are taken.

1. The measurements are taken at evenly-spaced intervals, with $t_N = 2$. Output, in a file named `partvecterr.txt` the discrepancy, as absolute error, between the computed velocity from the position and the given velocity, the computed acceleration from the velocity and the given acceleration, and the computed acceleration from the position and the given acceleration at each measurement at time $t_i$. The output should be one time measurement per line, measurements in the order given in the input file, lines separated by carriage returns, a single space between any two numbers on a line, in the order $(\Delta x'_i, \Delta y'_i)$, $(\Delta x''_i, \Delta y''_i)$, $(\Delta x''_i, \Delta y''_i)$, as indicated above. **(4)**

2. The measurements are taken $1 + t_i$, where $t_i$ at the roots of $T_N(t)$, the Chebyshev polynomial of degree $N$. (The initial data point at time $t = 0$ can be discarded.) Chebyshev polynomials, with degree $n \geq 0$, can be given in the following form

$$T_n(t) = \cos(n \arccos t).$$

   Output, in a file named `partvecterr.txt` the discrepancy, as absolute error, between the computed velocity from the position and the given velocity, the computed acceleration from the velocity and the given acceleration, and the computed acceleration from the position and the given acceleration at each measurement at time $t_i$. The output should be one time measurement per line, measurements in the order given in the input file, lines separated by carriage returns, a single space between any two numbers on a line, in the order $(\Delta x'_i, \Delta y'_i)$, $(\Delta x''_i, \Delta y''_i)$, $(\Delta x''_i, \Delta y''_i)$, as indicated above. **(4)**

3. The measurements are taken at $1 + t_i$, where $t_i$ the roots of $P_N(t)$, the Legendre polynomial of degree $N$. (The initial data point at time $t = 0$ can be discarded.) Legendre polynomials, with degree $n \geq 0$, are given recursively as

$$P_0(t) \equiv 1; \quad P_1(t) = t; \quad (n+1)P_{n+1}(t) + nP_{n-1}(t) = (2n+1)tP_n(t), n > 0.$$

   It is known that the roots of the Legendre polynomial $P_n$ are also simple and lie in the interval $[-1, 1]$. It is also known that

$$P'_n(t) = \sum_{\substack{k=0 \\ n+k \text{ even}}}^{n-1} (2k+1)P_k(t),$$

   and iterative methods for determining the roots of the Legendre polynomial $P_n$ often start from the roots of the Chebyshev polynomial $T_n$.

   Output, in a file named `partvecterr.txt` the discrepancy, as absolute error, between the computed velocity from the position and the given velocity, the computed acceleration from the velocity and the given acceleration, and the computed acceleration from the position and the given acceleration at each measurement at time $t_i$. The output should be one time measurement per line, measurements in the order given in the input file, lines separated by carriage returns, a single space between any two numbers on a line, in the order $(\Delta x'_i, \Delta y'_i)$, $(\Delta x''_i, \Delta y''_i)$, $(\Delta x''_i, \Delta y''_i)$, as indicated above. **(4)**

4. The measurements are taken at evenly-spaced intervals, with $t_N = 2$. The input file does not contain acceleration data. It is presumed that the model of the position of the particle is dependent

only on the initial velocity and an unknown constant force—that is, the position is quadratic in both directions. Output, in a file named `partvecterr.txt` the discrepancy, as absolute error, between the computed velocity from the position and the given velocity. The output should be one time measurement per line, measurements in the order given in the input file, lines separated by carriage returns, a single space between any two numbers on a line. The last line of the output file is a pair of numbers indicating the acceleration due to the force. $(x_i'', y_i'')$ **(3)**

Given data points $\{(x_i, y_i)\}$ on a uniform grid, i.e. $x_i = x_0 + ih$ with underlying function $f(x)$ such that $y_i = f(x_i)$, Taylor series expansions centered on $x = x_i$ provide approximations of derivatives $f^{(n)}(x_i)$, given a sufficient number of data points, through finite difference schemes. In particular, the first finite difference schemes for evaluating the derivative $f^{(n)}(x_i)$ require at least $n+1$ data points.

The Taylor series expansion about $x = x_i$, assuming infinite differentiability of $f$, is

$$f(x) = f(x_i) + f'(x_i)(x - x_i) + f''(x_i)\frac{(x - x_i)^2}{2} + \cdots + f^{(n)}(x_i)\frac{(x - x_i)^n}{n!} + \cdots . \tag{1}$$

A finite difference scheme on a uniform grid makes use of Taylor series expansions to approximate derivatives. For example, the first-order first forward finite difference scheme approximates $f'(x_i)$ by using the data point $(x_i, f(x_i))$ and a forward point, $(x_{i+1} = x_i + h, f(x_{i+1}))$. The Taylor series expansion (1) at the latter point gives:

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + f''(x_i)\frac{h^2}{2} + f'''(x_i)\frac{h^3}{6} + \cdots + f^{(n)}(x_i)\frac{h^n}{n!} + \cdots . \tag{2}$$

Thus,

$$f(x_{i+1}) - f(x_i) = f'(x_i)h + f''(x_i)\frac{h^2}{2} + \cdots = f'(x_i)h + \mathcal{O}(h^2) \Longrightarrow f'(x_i) = \frac{f(x_i + h) - f(x_i)}{h} + \mathcal{O}(h),$$

where the truncation error $\mathcal{O}(h)$ corresponds to the dominant remainder term, i.e. assuming $h$ is small enough such that terms with higher powers of $h$ are significantly smaller.

Likewise, the first-order backward finite difference scheme approximates $f'(x_i)$ by using the data point $(x_i, f(x_i))$ and a backward point, $(x_{i-1} = x_i - h, f(x_{i-1}))$. The Taylor series expansion (1) at the latter point gives:

$$f(x_{i-1}) = f(x_i) - f'(x_i)h + f''(x_i)\frac{h^2}{2} - f'''(x_i)\frac{h^3}{6} + \cdots + f^{(n)}(x_i)\frac{(-1)^n h^n}{n!} + \cdots . \tag{3}$$

Thus,

$$f(x_i) - f(x_{i-1}) = f'(x_i)h - f''(x_i)\frac{h^2}{2} + \cdots = f'(x_i)h + \mathcal{O}(h^2) \Longrightarrow f'(x_i) = \frac{f(x_i) - f(x_i - h)}{h} + \mathcal{O}(h).$$

Finally, the first-order central finite difference scheme approximates $f'(x_i)$ by using a forward point $(x_{i+1}, f(x_{i+1}))$ and a backward point, $(x_{i-1}, f(x_{i-1}))$. The Taylor series expansions (2)–(3) allow

$$f(x_{i+1}) - f(x_{i-1}) = 2f'(x_i)h + f'''(x_i)\frac{h^3}{3} + \cdots = 2f'(x_i)h + \mathcal{O}(h^3) \Longrightarrow f'(x_i) = \frac{f(x_i + h) - f(x_i - h)}{2h} + \mathcal{O}(h^2).$$

This early example illustrates that the central scheme on the uniform grid often provides better accuracy through lower truncation error, compared to either forward or backward schemes.

For higher-order schemes, a general approach can be described: for the second-order first forward finite difference scheme, to approximate $f'(x_i)$ by using the data point $(x_i, f(x_i))$ and two forward points, $(x_{i+1} = x_i + h, f(x_{i+1}))$ and $(x_{i+2} = x_i + 2h, f(x_{i+2}))$. The Taylor series expansion (1) at the latter point gives:

$$f(x_{i+2}) = f(x_i) + 2f'(x_i)h + 2f''(x_i)h^2 + f'''(x_i)\frac{2h^3}{3} + \cdots + f^{(n)}(x_i)\frac{2^n h^n}{n!} + \cdots . \tag{4}$$

(2) and (4) can be used to determine an approximation for $f''(x_i)$ by generating a system as follows

$$\begin{aligned} af(x_{i+2}) &\approx af(x_i) + 2af'(x_i)h + 2af''(x_i)h^2 \\ bf(x_{i+1}) &\approx \underline{bf(x_i) + bf'(x_i)h + f''(x_i)\frac{bh^2}{2}} \\ &- cf(x_i) + 0f'(x_i)h + f''(x_i)h^2 \end{aligned}$$

such that

$$af(x_i+2h)+bf(x_i+h)+cf(x_i) \approx f''(x_i)h^2 \Longrightarrow f''(x_i) \approx \frac{af(x_i+2h)+bf(x_i+h)+cf(x_i)}{h^2},$$

and the truncation error can be determined when the parameters $a$, $b$ and $c$ are determined.

The system determined by the above scheme is

$$\begin{cases} a+b+c &= 0, \\ 2a+b &= 0, \\ 4a+b &= 2 \end{cases} \Longrightarrow \begin{cases} a &= 1, \\ b &= -2, \\ c &= 1, \end{cases}$$

and, by (2) and (4),

$$f(x_i+2h)-2f(x_i+h)+f(x_i) = f''(x_i)h^2+f'''(x_i)\frac{h^3}{3}+\cdots \Longrightarrow f''(x_i) = \frac{f(x_i+2h)-2f(x_i+h)+f(x_i)}{h^2}+\mathcal{O}(h).$$

Similarly determined are the second-order first backward finite difference scheme

$$f(x_i)-2f(x_i-h)+f(x_i-2h) = f''(x_i)h^2+\mathcal{O}(h^3) \Longrightarrow f''(x_i) = \frac{f(x_i)-2f(x_i-h)+f(x_i-2h)}{h^2}+\mathcal{O}(h).$$

and the second-order first central finite difference scheme

$$f(x_i+h)-2f(x_i)+f(x_i-h) = f''(x_i)h^2+\mathcal{O}(h^4) \Longrightarrow f''(x_i) = \frac{f(x_i+h)-2f(x_i)+f(x_i-h)}{h^2}+\mathcal{O}(h^2).$$

**Source**: Jaan Kiusalaas, *Numerical Methods in Engineering with Python 3*, Chapter 5

Name: ____________________________________________

2. Given the first backward finite difference approximations for

$$f'(x) = \frac{f(x) - f(x-h)}{h} + \mathcal{O}(h) \qquad \text{and} \qquad f''(x) = \frac{f(x) - 2f(x-h) + f(x-2h)}{h^2} + \mathcal{O}(h),$$

derive the first backward finite difference approximation for $f'''(x)$ using the operation $f'''(x) = [f''(x)]'$.

The derivation follows

$$\begin{aligned} f'''(x) = [f''(x)]' &\approx \frac{f''(x) - f''(x-h)}{h} \\ &\approx \frac{1}{h}\left[\frac{f(x) - 2f(x-h) + f(x-2h)}{h^2} - \frac{f(x-h) - 2f(x-2h) + f(x-3h)}{h^2}\right] \\ &\approx \frac{f(x) - 3f(x-h) + 3f(x-2h) - f(x-3h)}{h^3}. \end{aligned}$$

Expanding by Taylor series gives

$$f(x) - 3f(x-h) + 3f(x-2h) - f(x-3h) = f'''(x)h^3 + \mathcal{O}(h^4),$$

thus

$$f'''(x) = \frac{f(x) - 3f(x-h) + 3f(x-2h) - f(x-3h)}{h^3} + \mathcal{O}(h).$$

6. Use finite difference approximations of $\mathcal{O}(h^2)$ to compute $f'(2.36)$ and $f''(2.36)$ from the following data:

| $x$ | 2.36 | 2.37 | 2.38 | 2.39 |
|---|---|---|---|---|
| $f(x)$ | 0.85866 | 0.86289 | 0.86710 | 0.87129 |

Using the second forward finite difference scheme for $f'(x)$, with $h = 0.01$,

$$f'(2.36) \approx \frac{4f(2.37) - f(2.38) - 3f(2.36)}{0.02} = 50[4(0.86289) - (0.86710) - 3(0.85866)] = \frac{53}{125} = 0.424.$$

Since the first forward finite difference scheme for $f''(x)$ also has $\mathcal{O}(h)$ error, using the second forward finite difference scheme and the first central difference scheme gives

$$\begin{aligned} f''(x) = [f'(x)]' &\approx \frac{4f'(x+h) - f'(x+2h) - 3f'(x)}{2h} \\ &\approx \frac{1}{2h}\left[4\frac{f(x+2h) - f(x)}{2h} - \frac{f(x+3h) - f(x+h)}{2h} - 3\frac{4f(x+h) - f(x+2h) - 3f(x)}{2h}\right] \\ &\approx \frac{5f(x) - 11f(x+h) + 7f(x+2h) - f(x+3h)}{4h^2}. \end{aligned}$$

Expanding by Taylor series gives

$$5f(x) - 11f(x+h) + 7f(x+2h) - f(x+3h) = 4f''(x)h^2 - 3f'''(x)h^3 + \mathcal{O}(h^4).$$

Again, using the second forward finite difference scheme and the first central difference scheme gives

$$\begin{aligned} f''(x) = [f'(x)]' &\approx \frac{4f'(x+h) - f'(x+2h) - 3f'(x)}{2h} \\ &\approx \frac{1}{2h}\left[4\frac{4f(x+2h) - f(x+3h) - 3f(x+h)}{2h} - \frac{f(x+3h) - f(x+h)}{2h} - 3\frac{4f(x+h) - f(x+2h) - 3f(x)}{2h}\right] \\ &\approx \frac{9f(x) - 23f(x+h) + 19f(x+2h) - 5f(x+3h)}{4h^2}. \end{aligned}$$

Expanding by Taylor series gives

$$9f(x) - 23f(x+h) + 19f(x+2h) - 5f(x+3h) = 4f''(x)h^2 - f'''(x)h^3 + \mathcal{O}(h^4).$$

These two schemes can be combined to have

$$\begin{aligned} f''(x) &\approx \frac{3[9f(x) - 23f(x+h) + 19f(x+2h) - 5f(x+3h)] + [5f(x) - 11f(x+h) + 7f(x+2h) - f(x+3h)]}{16h^2} \\ &\approx \frac{2f(x) - 5f(x+h) + 4f(x+2h) - f(x+3h)}{h^2}. \end{aligned}$$

Expanding by Taylor series gives

$$2f(x) - 5f(x+h) + 4f(x+2h) - f(x+3h) = f''(x)h^2 + \mathcal{O}(h^4),$$

thus

$$\begin{aligned} f''(x) &= \frac{2f(x) - 5f(x+h) + 4f(x+2h) - f(x+3h)}{h^2} + \mathcal{O}(h^2) \\ f''(2.36) &\approx \frac{2f(2.36) - 5f(2.37) + 4f(2.38) - f(2.39)}{0.0001} = -\frac{1}{5}. \end{aligned}$$

7. Estimate $f'(1)$ and $f''(1)$ from the following data:

| $x$ | 0.97 | 1.00 | 1.05 |
|---|---|---|---|
| $f(x)$ | 0.85040 | 0.84147 | 0.82612 |

Expanding by Taylor series around $x = 1$ gives

$$\begin{aligned} f(0.97) &= f(1) - f'(1)(0.03) + f''(1)\frac{(0.03)^2}{2} + k_1(0.03)^3, \\ f(1.05) &= f(1) + f'(1)(0.05) + f''(1)\frac{(0.05)^2}{2} + k_2(0.05)^3. \end{aligned}$$

Thus, the following approximations can be made:

$$\begin{aligned} f'(1) &\approx \frac{f(1.05) - f(0.97)}{0.05 - -0.03} = \frac{0.82612 - 0.85040}{0.08} = -\frac{607}{2000} = -0.3035; \\ f''(1) &\approx \frac{3f(1.05) - 8f(1) + 5f(0.97)}{[3(0.05)^2 + 5(0.03)^2]/2} = \frac{6(0.82612) - 16(0.84147) + 10(0.85040)}{3(0.0025) + 5(0.0009)} = -\frac{7}{30}. \end{aligned}$$

8. Given the data

| $x$ | 0.84 | 0.92 | 1.00 | 1.08 | 1.16 |
|---|---|---|---|---|---|
| $f(x)$ | 0.431711 | 0.398519 | 0.367879 | 0.339596 | 0.313486 |

calculate $f''(1)$ as accurately as you can.

It would seem that the most accurate central finite difference approximation that can be achieved with five equally-spaced data points has truncation error $\mathcal{O}(h^4)$ as given above:

$$\begin{aligned} f''(x) &\approx \frac{-f(x+2h) + 16f(x+h) - 30f(x) + 16f(x-h) - f(x-2h)}{12h^2}, \\ f''(1) &\approx \frac{-f(1.16) + 16f(1.08) - 30f(1) + 16f(0.92) - f(0.84)}{12(0.08)^2} \\ &\approx \frac{625[-0.431711 + 16(0.398519) - 30(0.367879) + 16(0.339596) - 0.313486]}{48} = 0.3681380208\bar{3}. \end{aligned}$$

10. Using five significant figures in the computations, determine $\mathrm{d}(\sin x)/\mathrm{d}x$ at $x = 0.8$ from (a) the first forward difference approximation and (b) the first central difference approximation. In each case, use $h$ that gives the most accurate result (this requires experimentation).

    If the data points are $x = 0.8$ and $x = 0.8 \pm h$, then the approximations to $\dfrac{\mathrm{d}\sin x}{\mathrm{d}x} = \cos x$, by trigonometric identities, are

$$
\begin{aligned}
\cos x &\approx \frac{\sin(0.8+h) - \sin(0.8)}{h} = \frac{\sin h}{h}\cos(0.8) - \left(\frac{1-\cos h}{h}\right)\sin(0.8) = d_f(h) \text{ by forward difference,} \\
&\approx \frac{\sin(0.8+h) - \sin(0.8-h)}{2h} = \frac{\sin h}{h}\cos(0.8) = d_c(h) \text{ by central difference.}
\end{aligned}
$$

    The accuracy of the approximations $|\cos(0.8) - d(h)|$ based on the schemes and $h = 10^{-t}$ for $t = 1, \ldots, 10$ is given on the next page. Note that, for $t > 8$ for the forward difference scheme, and for $t > 5$, the run-off error (multiplying by less than ten when $h$ is divided by ten) has exceeded the truncation error, thus providing the same error for both schemes. Also note how the truncation error drops linearly (dividing by ten when $h$ is divided by ten) for $d_f$ and quadratically (dividing by 100 when $h$ is divided by ten) for $d_c$.

    Using the values given, the best approximations, up to five significant digits, for $\cos(0.8)$ are attained for lowest $t$ given by $d_f(10^{-6}) \approx 0.69671 \approx d_c(10^{-3})$.

**Source**: Jaan Kiusalaas, *Numerical Methods in Engineering with Python 3*, Problem Set 5.1

Name: ______________________________

11. Use polynomial interpolation to compute $f'$ and $f''$ at $x = 0$, using the data

| $x$ | $-2.2$ | $-0.3$ | $0.8$ | $1.9$ |
|---|---|---|---|---|
| $f(x)$ | 15.180 | 10.962 | 1.920 | $-2.040$ |

Given that $f(x) = x^3 - 0.3x^2 - 8.56x + 8.448$, gauge the accuracy of the result.

The divided difference tableau for the four data points is

| $x_i$ | $y_i = [y_i]$ | $[y_i, y_{i+1}]$ | $[y_i, y_{i+1}, y_{i+2}]$ | $[y_0, y_1, y_2, y_3]$ |
|---|---|---|---|---|
| 1.9 | $-2.04$ | | | |
| | | $-3.6$ | | |
| 0.8 | 1.92 | | 2.1 | |
| | | $-8.22$ | | 1 |
| $-0.3$ | 10.962 | | $-2$ | |
| | | $-2.22$ | | |
| $-2.2$ | 15.18 | | | |

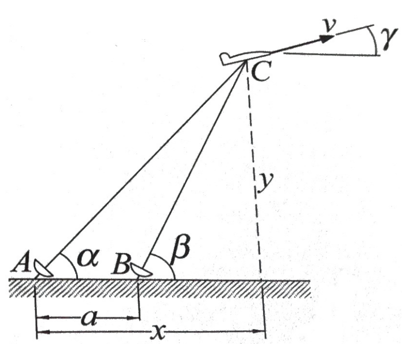
That gives an interpolant (which should be exact)

$$\begin{aligned}
p_3(x) &= -2.04 - 3.6(x - 1.9) + 2.1(x - 1.9)(x - 0.8) + (x - 1.9)(x - 0.8)(x + 0.3),\\
p_3'(x) &= -3.6 + 4.2(x - 1.35) + (x - 0.8)(x + 0.3) + (x - 1.9)(x + 0.3) + (x - 1.9)(x - 0.8),\\
p_3''(x) &= 4.2 + 2(x - 0.8) + 2(x - 1.9) + 2(x + 0.3) = 4.2 + 6(x - 0.8).
\end{aligned}$$

So,

$$\begin{aligned}
f'(0) = -8.56 = p_3'(0) &= -3.6 + 4.2(-1.35) + (-0.8)(0.3) + (-1.9)(0.3) + (-1.9)(-0.8)\\
f''(0) = -0.6 = p_3''(1) &= 4.2 + 6(-0.8).
\end{aligned}$$

13.




The radar stations $A$ and $B$, separated by the distance $a = 500$ m, track the plane $C$ by recording the angles $\alpha$ and $\beta$ at one-second intervals. If three successive readings are

| $t$ (s) | 9 | 10 | 11 |
|---|---|---|---|
| $\alpha$ | 54.80° | 54.06° | 53.34° |
| $\beta$ | 65.59° | 64.59° | 63.62° |

calculate the speed $v$ of the plane and the climb angle $\gamma$ at $t = 10$ s. The coordinates of the plane can be shown to be

$$x = a\frac{\tan\beta}{\tan\beta - \tan\alpha}, \qquad y = a\frac{\tan\alpha\tan\beta}{\tan\beta - \tan\alpha}.$$

It is possible to derive both $v$ and $\gamma$ as from the rate of changes of $x$ and $y$: $v = \sqrt{(x')^2 + (y')^2}$ and $\gamma = \tan^{-1}\dfrac{y'}{x'}$. What remains is to determine an approximation for the derivatives $x'$ and $y'$—this will be done by differentiation of the polynomial interpolant.

First, determine the values for $x$ and $y$ at the given time: by the formulas given,

| $t$ (s) | 9 | 10 | 11 |
|---|---|---|---|
| $\alpha$ (°) | 54.80 | 54.06 | 53.34 |
| $\beta$ (°) | 65.59 | 64.59 | 63.62 |
| $x$ (km) | 1.40192 | 1.45050 | 1.49864 |
| $y$ (km) | 1.98735 | 2.00084 | 2.01351 |

The divided difference tableau for the three $x$-data points (left) and $y$-data points (right) are

| $t_i$ | $x_i = [x_i]$ | $[x_i, x_{i+1}]$ | $[x_0, x_1, x_2]$ | $y_i = [y_i]$ | $[y_i, y_{i+1}]$ | $[y_0, y_1, y_2]$ |
|---|---|---|---|---|---|---|
| 11 | 1.49864 | | | 2.01351 | | |
| | | 0.04814 | | | 0.01267 | |
| 10 | 1.45050 | | −0.00072 | 2.00084 | | −0.00041 |
| | | 0.04958 | | | 0.01349 | |
| 9 | 1.40192 | | | 1.98735 | | |

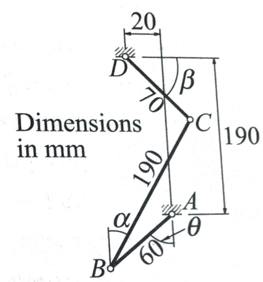
This gives

$$x(t) = 1.49864 + 0.04814(t-11) - 0.00072(t-11)(t-10) \implies x'(t) = 0.04814 - 0.00144(t-10.5),$$
$$y(t) = 2.01351 + 0.01267(t-11) - 0.00041(t-11)(t-10) \implies y'(t) = 0.01267 - 0.00082(t-10.5).$$

So $x'(10) = 48.88$ m / s and $y'(10) = 13.08$ m / s, so $v(10) \approx 50.6$ m / s and $\gamma(10) \approx 14.981°$.

It is also possible to interpolate on $d = a\dfrac{\sec\alpha\tan\beta}{\tan\beta - \tan\alpha}$, with $v = d'$ and $\gamma = \alpha'$.

14.




Geometric analysis of the linkage given resulted in the following table relating the angles $\theta$ and $\beta$:

| $\theta$ (deg) | 0 | 30 | 60 | 90 | 120 | 150 |
|---|---|---|---|---|---|---|
| $\beta$ (deg) | 59.96 | 56.42 | 44.10 | 25.72 | −0.27 | −34.29 |

Assuming that the member $AB$ of the linkage rotates with the constant angular velocity $\mathrm{d}\theta/\mathrm{d}t = 1$ rad / s, compute $\mathrm{d}\beta/\mathrm{d}t$ in rad / s at the tabulated values of $\theta$. Use cubic spline interpolation.

By chain rule $\dfrac{\mathrm{d}\beta}{\mathrm{d}t} = \dfrac{\mathrm{d}\beta}{\mathrm{d}\theta}\dfrac{\mathrm{d}\theta}{\mathrm{d}t}$, so determining $\dfrac{\mathrm{d}\beta}{\mathrm{d}\theta}$ is required. The natural cubic spline interpolant, given $h = -30°$, is

$$s_1(\theta) = \frac{\beta_1''\theta(\theta^2 - 900) + 6[56.42\theta - 59.96(\theta - 30)]}{180}$$
$$s_2(\theta) = \frac{\beta_1''(\theta-60)[900-(\theta-60)^2] - \beta_2''(\theta-30)[900-(\theta-30)^2] + 6[44.1(\theta-30) - 56.42(\theta-60)]}{180}$$
$$s_3(\theta) = \frac{\beta_2''(\theta-90)[900-(\theta-90)^2] - \beta_3''(\theta-60)[900-(\theta-60)^2] + 6[25.72(\theta-60) - 44.1(\theta-90)]}{180}$$
$$s_4(\theta) = \frac{\beta_3''(\theta-120)[900-(\theta-120)^2] - \beta_4''(\theta-90)[900-(\theta-90)^2] - 6[25.72(\theta-120) + 0.27(\theta-90)]}{180}$$
$$s_5(\theta) = \frac{\beta_4''(\theta-150)[900-(\theta-150)^2] + 6[0.27(\theta-150) - 34.29(\theta-120)]}{180}$$

where

$$\begin{aligned}
4\beta_1'' + \beta_2'' &= \frac{59.96 - 2(56.42) + 44.10}{150} = -\frac{439}{7500}, \\
\beta_1'' + 4\beta_2'' + \beta_3'' &= \frac{56.42 - 2(44.10) + 25.72}{150} = -\frac{101}{2500}, \\
\beta_2'' + 4\beta_3'' + \beta_4'' &= \frac{44.10 - 2(25.72) - 0.27}{150} = -\frac{761}{15000}, \\
\beta_3'' + 4\beta_4'' &= \frac{25.72 + 2(0.27) - 34.29}{150} = -\frac{803}{15000}.
\end{aligned}$$

The piecewise functions give

$$\begin{aligned}
s_1'(\theta) &= \frac{\beta_1''(\theta^2 - 900) + 2\beta_1''\theta^2 + 6(56.42 - 59.96)}{180} = \frac{\beta_1''(\theta^2 - 300) - 7.08}{60}, \\
s_3'(\theta) &= \frac{\beta_2''[300 - (\theta - 90)^2] - \beta_3''[300 - (\theta - 60)^2] - 36.76}{180}, \\
s_5'(\theta) &= \frac{\beta_4''[900 - (\theta - 150)^2] - 2\beta_4''(\theta - 150)^2 + 6[0.27 - 34.29]}{180} = \frac{\beta_4''[300 - (\theta - 150)^2] - 68.04}{60},
\end{aligned}$$

and the system provides

$$\begin{aligned}
\beta_1'' &= -\frac{439 \cdot 14}{209 \cdot 75 \cdot 25} + \frac{303}{209 \cdot 500} - \frac{761}{209 \cdot 15 \cdot 250} + \frac{803}{209 \cdot 15000} = -\frac{42319}{3135000}, \\
\beta_2'' &= \frac{439}{209 \cdot 500} - \frac{303}{209 \cdot 125} + \frac{761 \cdot 2}{209 \cdot 15 \cdot 125} - \frac{803}{209 \cdot 150 \cdot 25} = -\frac{2371}{522500}, \\
\beta_3'' &= -\frac{439}{209 \cdot 15 \cdot 125} + \frac{303 \cdot 4}{209 \cdot 15 \cdot 125} - \frac{761}{209 \cdot 250} + \frac{803}{209 \cdot 1000} = -\frac{27431}{3135000}, \\
\beta_4'' &= \frac{439}{209 \cdot 7500} - \frac{303}{209 \cdot 75 \cdot 25} + \frac{761}{209 \cdot 1000} - \frac{803 \cdot 7}{209 \cdot 15 \cdot 125} = -\frac{35099}{3135000}.
\end{aligned}$$

Thus,

$$\begin{aligned}
\left.\frac{\mathrm{d}\beta}{\mathrm{d}t}\right|_{\theta=0^\circ} &\approx s_1'(0)\,\frac{\mathrm{rad}}{\mathrm{s}} = \left(-5\beta_1'' - \frac{177}{1500}\right)\frac{\mathrm{rad}}{\mathrm{s}} = -\frac{8519}{156750}\,\frac{\mathrm{rad}}{\mathrm{s}} \approx -0.0543\ \frac{\mathrm{rad}}{\mathrm{s}}, \\
\left.\frac{\mathrm{d}\beta}{\mathrm{d}t}\right|_{\theta=30^\circ} &\approx s_1'(30)\,\frac{\mathrm{rad}}{\mathrm{s}} = \left(10\beta_1'' - \frac{177}{1500}\right)\frac{\mathrm{rad}}{\mathrm{s}} = -\frac{19828}{78375}\,\frac{\mathrm{rad}}{\mathrm{s}} \approx -0.253\ \frac{\mathrm{rad}}{\mathrm{s}}, \\
\left.\frac{\mathrm{d}\beta}{\mathrm{d}t}\right|_{\theta=60^\circ} &\approx s_3'(60)\,\frac{\mathrm{rad}}{\mathrm{s}} = \left(-10\beta_2'' - 5\beta_3'' - \frac{919}{1500}\right)\frac{\mathrm{rad}}{\mathrm{s}} = -\frac{328259}{627000}\,\frac{\mathrm{rad}}{\mathrm{s}} \approx -0.524\ \frac{\mathrm{rad}}{\mathrm{s}}, \\
\left.\frac{\mathrm{d}\beta}{\mathrm{d}t}\right|_{\theta=90^\circ} &\approx s_3'(90)\,\frac{\mathrm{rad}}{\mathrm{s}} = \left(5\beta_2'' + 10\beta_3'' - \frac{919}{1500}\right)\frac{\mathrm{rad}}{\mathrm{s}} = -\frac{45323}{62700}\,\frac{\mathrm{rad}}{\mathrm{s}} \approx -0.723\ \frac{\mathrm{rad}}{\mathrm{s}}, \\
\left.\frac{\mathrm{d}\beta}{\mathrm{d}t}\right|_{\theta=120^\circ} &\approx s_5'(120)\,\frac{\mathrm{rad}}{\mathrm{s}} = \left(5\beta_4'' - \frac{1701}{1500}\right)\frac{\mathrm{rad}}{\mathrm{s}} = -\frac{746117}{627000}\,\frac{\mathrm{rad}}{\mathrm{s}} \approx -1.19\ \frac{\mathrm{rad}}{\mathrm{s}}, \\
\left.\frac{\mathrm{d}\beta}{\mathrm{d}t}\right|_{\theta=150^\circ} &\approx s_5'(150)\,\frac{\mathrm{rad}}{\mathrm{s}} = \left(-10\beta_4'' - \frac{1701}{1500}\right)\frac{\mathrm{rad}}{\mathrm{s}} = -\frac{32041}{31350}\,\frac{\mathrm{rad}}{\mathrm{s}} \approx -1.02\ \frac{\mathrm{rad}}{\mathrm{s}}.
\end{aligned}$$

16. The relationship between stress $\sigma$ and strain $\varepsilon$ of some biological materials in uniaxial tension is $\dfrac{\mathrm{d}\sigma}{\mathrm{d}\varepsilon} = a + b\sigma$, where $a$ and $b$ are constants ($\mathrm{d}\sigma/\mathrm{d}\varepsilon$ is called the *tangent modulus*). The following table gives the results of a tension test on such a material:

| Strain $\varepsilon$ | Stress $\sigma$ (MPa) |
|---|---|
| 0 | 0 |
| 0.05 | 0.252 |
| 0.10 | 0.531 |
| 0.15 | 0.840 |
| 0.20 | 1.184 |
| 0.25 | 1.558 |
| 0.30 | 1.975 |
| 0.35 | 2.444 |
| 0.40 | 2.943 |
| 0.45 | 3.500 |
| 0.50 | 4.115 |

Determine the parameters $a$ and $b$ by linear regression.

The values $\delta = \dfrac{\mathrm{d}\sigma}{\mathrm{d}\varepsilon}$ are approximated by finite difference schemes with error $\mathcal{O}(h^2)$:

$$\sigma'(0) \approx \frac{-3\sigma(0) + 4\sigma(0.05) - \sigma(0.1)}{0.1}; \qquad \sigma'(0.5) \approx \frac{\sigma(0.4) - 4\sigma(0.45) + 3\sigma(0.5)}{0.1};$$
$$\sigma'\left(\frac{n}{20}\right) \approx \frac{\sigma([n+1]/20) - \sigma([n-1]/20)}{0.1}, \qquad 1 \le n \le 9.$$

These are

| Strain $\varepsilon$ | Stress $\sigma$ (MPa) | Tangent modulus $\delta = \dfrac{\mathrm{d}\sigma}{\mathrm{d}\varepsilon}$ |
|---|---|---|
| 0 | 0 | 5.49 |
| 0.05 | 0.252 | 5.31 |
| 0.10 | 0.531 | 5.88 |
| 0.15 | 0.840 | 6.53 |
| 0.20 | 1.184 | 7.18 |
| 0.25 | 1.558 | 7.91 |
| 0.30 | 1.975 | 8.86 |
| 0.35 | 2.444 | 9.68 |
| 0.40 | 2.943 | 10.56 |
| 0.45 | 3.500 | 11.72 |
| 0.50 | 4.115 | 12.88 |

The parameters $a$ and $b$ are determined by linear regression:

$$\bar{\sigma} = \frac{\sum \sigma_i}{n} = \frac{9671}{5500}; \quad \bar{\delta} = \frac{\sum \delta_i}{n} = \frac{92}{11} \Longrightarrow b = \frac{\sum \sigma_i(\delta_i - \bar{\delta})}{\sum \sigma_i(\sigma_i - \bar{\sigma})} = \frac{709991640}{400469947} \approx 1.77,$$
$$\Longrightarrow a = \bar{\delta} - b\bar{\sigma} = \frac{577764420578}{110129235425} \approx 5.25.$$

It is also possible to discard the end-values to generate a slightly different line.

⋆ Find, for the data points $(x-h, f(x-h))$, $(x, f(x))$, $(x+k, f(x+k))$, the approximation of $f''$ using the polynomial interpolant of the given points, and determine the truncation error of the approximation at each data point.

From Newton's divided-difference scheme, the interpolant is

$$p(y) = f(x-h) + \frac{f(x) - f(x-h)}{h}(y-x+h) + \frac{hf(x+k) - (h+k)f(x) + kf(x-h)}{hk(h+k)}(y-x)(y-x+h),$$

thus $f''(y) \approx p''(y) \equiv \dfrac{2hf(x+k) - 2(h+k)f(x) + 2kf(x-h)}{hk(h+k)}$. From the Taylor series expansions, the leading term of the truncation error is as follows:

$$f''(x) \approx p''(x) + \frac{h-k}{3} f'''(x), \qquad f''(x-h) \approx p''(x-h) + \frac{2h+k}{3} f'''(x-h),$$
$$f''(x+k) \approx p''(x+k) + \frac{h+2k}{3} f'''(x+k).$$

As long as $h \neq k$, $f''(y) = \dfrac{2hf(x+k) - 2(h+k)f(x) + 2kf(x-h)}{hk(h+k)} + \mathcal{O}(h+k)$, $y = x, x-h, x+k$.

**Source**: Jaan Kiusalaas, *Numerical Methods in Engineering with Python 3*, Problem Set 5.1