AM 10 – SYLLABUS

Each week consists of an **asynchronous** component, where a basic math topic is reviewed as motivation, a **synchronous** component, where the code is developed to handle the relevant mathematical operations and a **lab** consisting of examples and applications.

MODULE 1 : Basics

Week 0:
Math: Motivation: why do we need computation in science and engineering?
Code: Simple code statements. Files and directories. Print and read statements.
Lab: Introduction to the Python environment and Jupyter Notebooks.

Week 1:
Math: Number systems: integer, rational, irrational, base-10, base-2 systems of numbers. Precision, machine precision, rounding error.
Code: Numbers in computers, how you represent a number (integer, float, complex). Range of values in 32-bit, 64-bit machines. Logical variables. If/else statements.
Lab: Simple operations: addition, multiplication, division, powers, exponential, logarithm, trigonometric functions. Introduction to numpy libraries.

Week 2:
Math: What are scalars, vectors, tensors (2^{nd} rank as example).
Code: Arrays, elements of an array. Lists and Dictionaries.
2D arrays, matrices, operation with arrays, dot product, matrix-vector multiplication.
Array creation, operations: linspace, zeros, arange, sum, ave, max, min.
Lab: Playing with arrays: indexing, slicing. Reading / writing data to files.

Week 3:
Math: Sequences and Series of numbers: motivation for loop operations
Code: Loops: for, while. Break statement. Nested loops. Basics of plotting - plot in matplotlib.
Lab: applications of loops, conditional operations.

Week 4:
Math: Functions of real variables. The normalized gaussian, the step (theta) function, the delta function: motivation for defining new or parametric functions.
Code: Definition (def) of special-purpose functions. More on plotting, composite plots, labels.
Lab: Examples of basic functions and their plots (trigonometric, exponential, gaussian, parametric plots of various shapes and forms).

MODULE 2 : Elementary

Week 5:

Math: Derivatives: slope as tangent. Taylor expansion, propagation of values.
Code: Pseudocode: translating equations to computer code. Numerical differentiation.
Lab: Application to derivatives of known and unknown (data only) functions -bacteria population

Week 6:
Math: Root finding, roots of simple polynomials (2^{nd} and 3^{rd} degree)
Code: Numerical root finding: bracketing, Newton-Raphson method
Lab: Newton-Raphson, application to known and unknown (data only) function

Week 7:
Math: Integrals: area under the curve.
Code: Numerical integration of functions: trapezoidal rule and Simpson's rule methods
Lab: Application of numerical integration to known and unknown (data-only) functions

Week 8:
Math: Approximating data with functions: mean square error, optimization of fit
Code: Curve fitting with linear fit. Functions in higher dimensions: visualization with contours
Lab: polynomial fit, curve_fit routine, arbitrary number of parameters

MODULE 3 : Advanced

Week 9:
Math: What is a differential equation? First-order Ordinary Differential Equations: exponential growth and decay.
Code: Euler's and modified Euler's methods.
Lab: application to epidemics – SIR model

Week 10:
Math: Second-order Differential Equations: sinusoidal functions. Eigenfunctions. Harmonic oscillator. Wave equation: extension to partial differential equations.
Code: Numerical solution for 2^{nd} order equations. Second order Runge-Kutta.
Lab: Oscillations and waves. Animation of plots.

Week 11:
Math: Sampling and Random numbers. Monte Carlo integration: calculation of pi
Code: calculation of simple integrals of single-variable functions.
Lab: Calculation of single-variable and multi-variable integrals.

Week 12:
Math: What is a simulation? Example: partial differential equations: the diffusion equation in 1 and 2 spatial dimensions.
Code: Random walk in 1 spatial dimension, averages over many samples, diffusion in 1D.
Lab: Random walk in 2 dimensions, connection to diffusion 2D diffusion constant.