I. INTRODUCTION

Numerical methods

The numerical methods, or numerical analysis, are simplified procedures to get *approximate numerical solutions* to equations and problems in algebra, calculus and other fields of mathematics. These procedures should be organized in definite and general steps that are applicable to the problem which they are formulated to solve. These steps are known as algorithms and can be programmed by using a computer language.

Programming languages used in numerical methods

Numerical calculation of science and engineering problems was one of the first applications of computers in the 1950's. The early high-level computer language used for the purpose was FORTRAN. Development of other powerful languages like Pascal, C++, java and Python and integrated computation environments such as MATLAB and Mathematica increased the efficiency of codes and techniques of numerical methods in parallel with the advantages of each language.

Why Python programming language

In this course, Python is used in programming the numerical methods because it has many advantages:

1- It has very simple and intuitive syntax with powerful statements and data structures.
2- Numerous mathematical functions are defined in its standard and third party modules.
3- Vectors and matrices can be created and manipulated by means of operators and functions from the module *NumPy*.
4- Plotting functions of different plot types can be imported from the module *matplotlib*.
5- *SciPy* is a comprehensive module of a large collection of numerical method functions in various topics such as interpolation, integration, linear algebra, etc.
6- It is available for all main operating systems.
7- It has several integrated development environments (IDE) like IDLE, Spyder and PyCharm with powerful code editing, execution and debugging capabilities.
8- Python and its standard and third party modules are available for free download.

The goal of the course

This course aims at teaching how to

1- program the numerical methods *step by step* to create simple and efficient Python codes that output the numerical solution at the required degree of accuracy.
2- create and manipulate arrays (vectors and matrices) by using *NumPy.*
3- use the plotting functions of Python library *matplotlib* to show the results graphically
4- apply numerical analysis functions from *SciPy* on the problems discussed in the course.

Who this course is made for

This course is designed for:

1- the students who currently study their first course in numerical methods and need to understand how the methods are coded and how related Python tools are used to solve problems.
2- the students who need to create their own numerical analysis codes or use Python numerical libraries for their course, project or thesis works.

Course Requirements

This course requires the following:

1- A good background in algebra and calculus, in addition to the basic knowledge about computers.
2- A Python IDE and its libraries NumPy, matplotlib and SciPy installed on the computer that will be used in working the examples and the exercises.
3- No previous experience in programming with Python since all statements and functions will be thoroughly explained.

How to install Python

Python and its IDE IDLE can be downloaded for free from http://www.python.org. The numerical packages NumPy, SciPy and matplotlib can be downloaded for free from http://pypi.python.org (The Python Package Index) as Python Wheel packages (.whl files)

Notes:

1- Python should be installed first.
2- In this course Python 3.6 is used.
3- The packages should be compatible with the installed Python version (cp36).
4- Numpy should be installed first because the other packages depend on its components.
5- SciPy is not available for Windows from pypi.python.org. Windows users can download its .whl file for free from http://www.lfd.uci.edu/~gohlke/pythonlibs/#scipy
6- The Python Wheel packages can be installed in Python path system by using its **pip** command as following:
   a) Open the terminal (command prompt for Windows) and change the directory to the folder where the files are saved by using **cd** command.
   b) Enter the command: **pip install** file_name.whl

Course Topics

I. Introduction
II. Roots of High-Degree Equations
- Simple Iteration Method
- Newton-Raphson's Method
- Bisection Method

III. Interpolation and Curve Fitting
- Lagrange's Method
- Newton's Method
- Fitting with a Straight Line (Linear Regression)
- Fitting with a Polynomial Curve

IV. Numerical Differentiation
- Finite Differences Method

V. Numerical Integration
- Trapezoidal Rule
- Simpson's 1/3 Rule
- Simpson's 3/8 Rule
- Multiple Integrations

VI. Systems of Linear Equations
- Gauss Elimination Method
- Jacobi's Method
- Gauss-Seidel's Method

VII. Ordinary Differential Equations
- Euler's Method
- Second Order Runge-Kutta's Method
- Fourth Order Runge-Kutta's Method
- Higher-Order Ordinary Differential Equations