Programming with Python in Numerical Analysis

by Matthew L. Greiss

The field of numerical analysis makes use of many different algorithms in order to approximate the solutions for a wide variety of problems. A number of these algorithms can be quite long and may require several calculations. Previously, calculations such as these had to be done by hand, and could thus take minutes, hours, or even days to complete. Now, with the advent of modern programming, devices such as computers, tablets, and even phones, can complete these calculations in a matter of seconds. The programs that are used to perform such calculations can be written in a number of different languages, one of which is Python.

This paper will cover ten programs that have been written in the programming language Python and are designed to solve problems or complete tasks within the field of numerical analysis. The main point of this paper will be to examine the methodology behind each program and how the source code executes upon those ideas. Since this paper's main topic is the source code of these programs, a pdf of it will accompany this paper. It would be unreasonable to cover every character of the source code, so insignificant elements, as well as those whose purpose is immediately apparent, will not be discussed. While the accompanying pdf will contain the entirety of the source code, it will still be helpful to include excerpts from it, as this will make certain explanations clearer. When source code is presented it may be abridged or slightly altered for the sake of clarity.

The first program will convert binary machine numbers into base-10 numbers. These are numbers that computers use to store base-10 numbers. The second, third, fourth, and fifth programs, are all root-finding algorithms. The second is a bisection method program, and it repeatedly bisects intervals until they converge upon a root. The third is a fixed-point iteration program and it will isolate one of a function's variables, and find a fixed point for the resulting expression. The fourth is a Newton-Raphson method program and it will iterate an expression that is comprised of both a function and its derivative. The fifth is a secant method program and it will iterate a recurrence relation. The sixth is an Euler method program, and it will solve ordinary differential equations by generating a series of discrete points. The final four programs all use interpolation to approximate functions. The seventh is a Lagrange polynomial program and it produces a polynomial by generating a series of terms, each constructed from a combination of data points, and adding them together. The eighth is a Newton polynomial program and it produces a polynomial by taking the divided differences of adjacent data points. The ninth is a Hermite interpolation program and it approximates coordinates of a function by taking the divided differences of a function's data points and its derivatives data points. The tenth and final program is a cubic spline interpolation program, and it produces a piece-wise function, comprised of cubic splines, in order to approximate functions.
Specifically, they are the bisection method, fixed-point iteration, Newton-Raphson method, and secant method.