Project Euler - 101

Supporting math for problem 101

Hilmir Vilberg Arnarsson

19th of July 2022

Introduction

We want to find the optimum polynomial, i.e. of lowest degree, that models a sequence u_n up to some term m. Let us call this polynomial $O(m,n)$. The first thing to notice about this polynomial is that it will be of degree $m-1$ since we need exactly m values to compute a polynomial of degree $m-1$ indirectly by the fundamental theorem of algebra.

Observations

For a given optimum polynomial of degree m, we have that $O(m,n) = P(x)$ where $P(x) = a_0x^{m-1} + a_1x^{m-2} + \cdots + a_{m-2}x + a_{m-1}$. We want to determine $a_0, a_1, \ldots, a_{m-1}$ and we know from u_n that $P(1) = u_1, P(2) = u_2, \ldots, P(m) = u_m$. This means we have m equations and m variables to determine. This should ring bells if you know anything about linear algebra. We can set up a matrix and then solve it to attain $a_0, a_1, \ldots, a_{m-1}$. We get:

$$\begin{cases} P(1) & = u_1 \\ P(2) & = u_2 \\ & \vdots \\ P(m) & = u_m \end{cases} \iff \left[\begin{array}{cccc|c} 1 & 1^1 & \ldots & 1^{m-1} & u_1 \\ 1 & 2^1 & \ldots & 2^{m-1} & u_2 \\ \vdots & & & \vdots & \vdots \\ 1 & m^1 & \ldots & m^{m-1} & u_m \end{array}\right]$$

This is easily solvable using python coupled with its module numpy.linalg. Once $a_0, a_1, \ldots, a_{m-1}$ are obtained, it is simple to calculate $P(m+1)$ which gives the value we are interested in. Then we iterate through values of m from 1 to 10.