Numeric Methods Benchmark Analysis

1st Santiago Valencia Arango
dept. of science
EAFIT
Medellín, Colombia
svalenciaa@eafit.edu.co

2nd Juan Manuel Young Hoyos
dept. of science
EAFIT (of Aff.)
Medellín, Colombia
jmyoungh@eafit.edu.co

***Abstract*—The idea of this project is to test out if it is worth it to use C++ instead of Python for algorithms of some numeric methods, like finding roots of non-linear equations using Newton-Raphson's method.**

***Index Terms*—Numeric methods, C++, Python, algorithms, insert**

I. Introduction

Why this project? the aim of this project is to be able to analyze and draw a conclusion to the question, is it worth spending more time programming in C ++? Or is it more profitable to do this type of algorithms using Python and its libraries in terms of development time and program performance?

II. What will we test?

A. Algorithms

In this project we will only find roots of an equation using:

- Newton-Rapshon method.
- Bisection method.
- Secant method.
- Regula-Falsi method.
- Fixed-point iteration method.

B. G++ compiler

g++ (Ubuntu 9.3.0-17ubuntu1 20.04) 9.3.0

C. Python interpreter

Python 3.8.5

III. Newton-Raphson comparison

A. Variables

Equation: $x^3 + x^2 + 3$
Initial point: -20
Tolerance: 0.00001
Maximum number of iterations: 20

B. C++ Time

After 10 tries the average execution time in C++ is 0.003 seconds

C. Python Time

After 10 tries the average execution time in Python is 0.915 seconds

IV. Bisection comparison

A. Variables

Equation: $3x^3 - 2x - 5$
Lower end: 0
upper end: 2
Tolerance: 0.00001
Maximum number of iterations: 20

B. C++ Time

After 10 tries the average execution time in C++ is 0.004 seconds

C. Python Time

After 10 tries the average execution time in Python is 0.287 seconds

V. Secant comparison

A. Variables

Equation: $x^3 + 1$
Lower end: 0
upper end: 2
Tolerance: 0.00001
Maximum number of iterations: 20

B. C++ Time

After 10 tries the average execution time in C++ is 0.0035 seconds

C. Python Time

After 10 tries the average execution time in Python is 0.655 seconds

VI. Regula-Falsi comparison

A. Variables

Equation: $\cos(x) - xe^x$
Lower end: 0
upper end: 2
Tolerance: 0.00001
Maximum number of iterations: 20

B. C++ Time

After 10 tries the average execution time in C++ is 0.0055 seconds

C. Python Time

After 10 tries the average execution time in Python is 0.368 seconds

VII. Newton-Raphson comparison

A. Variables

Equation: $\frac{2-e^x+x^2}{3}$
Initial point: 0
Tolerance: 0.00001
Maximum number of iterations: 20

B. C++ Time

After 10 tries the average execution time in C++ is 0.0083 seconds

C. Python Time

After 10 tries the average execution time in Python is 0.423 seconds

VIII. Conclusion

In conclusion the construction of root equation methods in both python and c++ are relatively simple, the comparison was based on discovering the performance of these numerical methods in a interpreted language vs a compiled language, in this case the comparison between python and c++ gives us some pretty clear results, c++ is much more efficient executing the algorithms, and not constituting a very high development complexity with respect to python, because normally due to the convenience of the libraries in python usually the development of these methods are faster, but if you are looking for efficiency in the results should be developed with c++.

With respect to the level of knowledge that you must have to develop these algorithms should be much higher in the case of c++, compared to python.

A topic that was not treated in this project and that is raised as future work is the development of these methods in python but to use an extension of c++ that allows the execution of functions and declaration of variables with c++ allowing the facility of python and the great power of the compiler of c++ and its optimizations.

References

[1] Burden, Richard L. and Faires, Duglas. Análisis Numérico. Editorial Thomson. 9 Edición 2011.