

Mathematical Functions C Code Generator

Vishal Limgire, Mahesh Nikam, Diksha Jadhav, Pooja Pawar

Department of Computer Engineering, Trinity Academy of Engineering Pune, Maharashtra, India

ABSTRACT

The standard mathematical library known as libm offers a small set of functions in a small set of precisions. Performance of libm functions is of critical importance, in particular in scientific and financial computing libm for each new processor, a time-consuming and error-prone task. A first objective of this project is therefore to automate libm development, up to the point where libms generated in a fraction. Applications often require functions not present in the libm. In such case, a composition of libm functions may be very inefficient, or simply not possible (function defined by a differential equation or by interpolation of a set of data points for instance). Even the libm functions are oared in a limited choice of implementations which will most of the time poorly match the actual needs of an application in terms of range, accuracy, performance and resource consumption. The set of functions and contexts available to programmers. The process of implementing a mathematical function into code is composed of several steps; typically range reduction, approximation, evaluation (additions, multiplications, and pre-computed constant values). Approach of this project is therefore to unify the understanding of function, sharing common mathematical, methodological, and implementation tools. In particular this will lead to on-demand generation for the same breadth of targets and with the same guarantees on numerical quality.

Keywords: Processors, Code generation, Elementary function approximation, Interfaces, Reusable Software, Reusable libraries.

I. INTRODUCTION

Mathematical function c code generation is concept in which directly c code will be generate on the given function when user will input. This will be more efficient to developer, mathematician, scientist and students for their respective use. It will help user reduce problem of time, effort and hand code error and also get more result that is accurate in digit and precision. This application will successfully overcome the drawbacks of previous application.

Mathematical function c code generation it helps save time, effort and also is a hand code error. We need to set up some strategies to encourage user will easily handle tool that mean user friendly. These may include:

1. Input as mathematical function and given range.
2. Generate mathematical c code for given function and range[1].

The number of functions of interest is huge. There is an even larger number of implementation contexts: required precision, target hardware, performance/cost constraints. This defines an enormous set of implementations of interest, far too large to manually implement them all [2]. The current solutions consist in living with a restricted set of functions/precisions (the libm approach), and living with implementations that are widely sub-optimal for most contexts. For instance, the code of some functions in open source libms is now twenty years old. As long as each function must be implemented manually, the set of functions available to programmers will be limited [3]. Programmers, however, do need application-specific functions not present in the libm.

This implementation is likely to be much less accurate and much less efficient than that of an expert. We must insist again on the accuracy issue this is not only a problem of performance, but also of the validity of the result [4].

II. METHODS AND MATERIAL

A. System Design

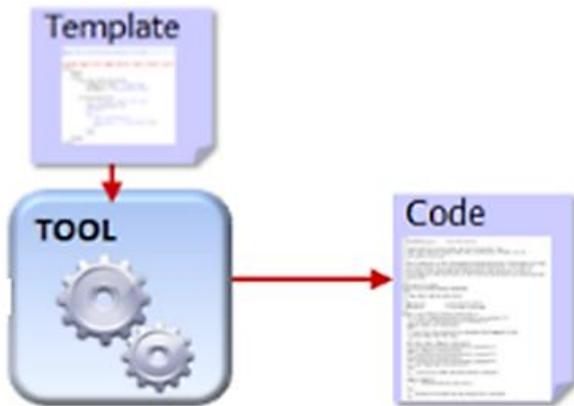


Figure -1: The proposed tool suite

We are currently focusing on two main use cases where we need a libm generator, both illustrated by Figure 1. The first one targets the widest audience of programmers. It is a push-button approach that will try to generate code on a given domain and for a given precision for an arbitrary univariate function with continuous derivatives, some order. The function may be provided as a mathematical expression, or even as an external library that is used as a black box. Call this approach the open-ended approach, in the sense that the function that can be input to the generator is arbitrary – which does not mean that the generator will always succeed in handling it. Section III will describe how this generator has evolved from simple polynomial approximations to the generation of more sophisticated evaluations schemes, including attempts to range reduction. Here, the criterion of success is that the generated code is better than whatever other approach the programmer would have to use (composition of libm function, numerical integration if the function is defined by an integral, etc). “Better” may mean faster, or more accurate, or better behaved in corner cases, etc [3].

B. Back end Process

To overcome the drawbacks of previous system we proposed a new system or tool to overcome previous problems. So in our tool we are mainly dealing with accuracy speed, and length of the code which resulted in failure of previous system. so user will provide the any mathematical functions such as arithmetic, trigonometric, exponential etc. as a input and in result he/she will get the corresponding code for that

mathematical functions and this code is useful for developers, mathematicians and researchers for their further implementations.

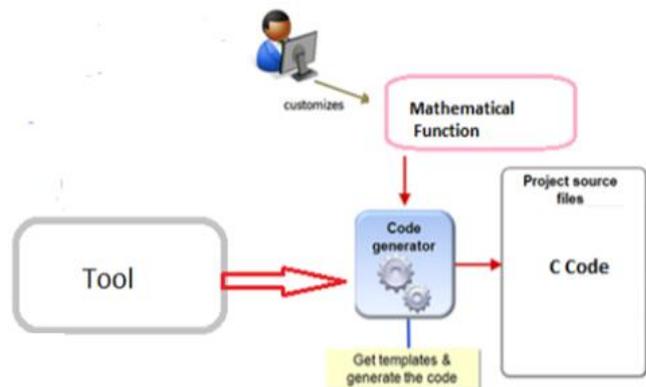


Figure2. Tool Back End Process

Our tool will contains a input screen where user will provide the mathematical functions with parameters such as angle and precision and this function is go in database and it will check whether that function is present in database or not. If that function is present in database then it will produce the code corresponding to the functions and if that function is not present in the database then it display the error message incorrect function. This tool is easy to handle as compare to MATLAB .we tried to develop an tool which will help in giving small amount of code to the users as compare to MATLAB. The main aim or goal of our tool is to provide a code, which will help to developers, researchers, mathematician and student.

C. Software Implementation

✓ General Overview

In our system we are generating the c code for mathematical functions. To understand the challenge of writing code generator for mathematical function manually it is important to note that how they are implemented. In this application the first target is programmer. It is a push button approach that will try to generate code. In this we embed the existing hand crafted code in a framework to make it more generic. We describe a python framework is manage back end code generation we present it form point of view of developer. All the framework is implemented in python language, Python variables as being the output of generated code.

✓ Python Language

1) Free: As in speech and as in beer. (It won't cost you a thing, and you are allowed to view and modify the source.)

2) Beautiful programming language: Python was created to be a generic language that is easy to read, while Matlab started as a matrix manipulation package to which they added a programming language. As you become more familiar with Python, you will be amazed with how well it is designed. There is only one word for that: Beautiful.

3) Powerful: Because it's so well designed, it's easier than other languages to transform your ideas into code. Further, Python comes with extensive standard libraries, and has a powerful data types such as lists, sets and dictionaries. These really help to organize your data.

4) Namespaces: Matlab supports namespaces for the functions that you write, but the core of Matlab is without namespaces; every function is defined in the global namespace. Python works with modules, which you need to import if you want to use them. (For example from skimage import morphology.) Therefore Python starts up in under a second. Using namespaces gives structure to a program and keeps it clean and clear. In Python everything is an object, so each object has a namespace itself. This is one of the reasons Python is so good at introspection.

5) Portability: Because Python is for free, your code can run everywhere. Further, it works on Windows, Linux, and OS X.

6) Class and function definitions: Functions and classes can be defined anywhere. In one file (whether it is a module or a script) you can design as many functions and classes as you like. You can even define one in the command shell if you really want to.

7) Great GUI toolkits: With Python you can create a front-end for your application that looks good and works well. You can chose any of the major GUI toolkits like TKinter or Qt. Pyzo comes with PySide (a wrapper for Qt) [2].

✓ Processor Description :

Modern processor such as IA 32 and ARM exhibits lots a variety of parallelism and the capabilities of hardware to extract their parallelism. Generating optimized code for variety of processor is very challenging so we want to optimized code not only for processor but also for context that include the processor. For this we define for each processor a class that provides information to optimization steps and also provide code generation service. So for constructing this tool we are using 2GB of RAM and Core to Duo processor.

✓ Platform Independent :

This tool is platform independent i.e. run on both ubuntu and Microsoft windows .Only difference is in their file format for windows batch file is used while for ubuntu shell script is used.

- 1) Batch File: Batch file is kind of Microsoft windows file. It consist of series of command executed by command line interpreter. It is stored in plain text. When batch file is run the shell program read the file and execute its command. The extension for batch file is .bat
- 2) Shell Script: It is a computer program run by UNIX shell. Operation perform by this are file manipulation, program execution, printing text etc.

✓ GCC Compiler or G++ Compiler :

Compiler used for generating code is G++. The compiler that take C or C++ source code and compile it into binary file that can be executed to actually run the program. For compiling your source code must reside in the directory that your source file are. By using G++ we can compile multiple files by compiling separate module and linking them into single file.

III.RESULTS AND DISCUSSION

As a result at the end we can say that open-ended code generation has reached a certain level of maturity. It is able to generate the code as quickly as possible. The produced code is available quickly and reduce cost, time effort and hand code error.

IV.CONCLUSION

A good case study for working on such integration is the correct rounding of elementary functions. It presents several challenges, such as function evaluation in larger-than-standard precisions or less common formats. Ideally, we wish we could have a generator where we have a clear separation (as on Figure 1) between a front-end building an approximation scheme, and a back end implementing it on a given target technology.

V. FUTURE SCOPE

1. More functions we can added with respect to time and need.
2. Complex operation such as integration and derivation is implemented.

VI.REFERENCES

- [1] Nicolas Brunie, Olga Kupriianova "Code generators for mathematical functions" 2015 IEEE 22nd Symposium on Computer Arithmetic.
- [2] M. Dukhan, "PeachPy: A Python framework for developing high performance assembly kernels," in Python for High Performance and Scientific Computing, 2013.
- [3] C. S. Anderson, S. Story, and N. Astafiev, "Accurate math functions on the Intel IA-32 architecture: A performance driven design," in 7th Conference on Real Numbers and Computers, 2006, pp. 93–105.
- [4] "IEEE standard for floating-point arithmetic," IEEE 754-2008, also ISO/IEC/IEEE 60559:2011, Aug. 2008.