

# Using Scientific Software engineering to implement the transcendental function $y = ab^x$ .

**Herve Ngomseu Fotsing (40067741)**

*SEON 6011 – Software Engineering Processes*

Project Report

Concordia University

August 05<sup>th</sup>, 2022

**Repository address:** [CLICK HERE!!](#).



*Keywords: agile methodologies and DevOps; software engineering; versioning and testing;  
transcendental function.*

# Contents

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>Introduction</b>                                     | <b>3</b> |
| 1.1      | <i>Description of the function</i> . . . . .            | 3        |
| 1.2      | Context of use model . . . . .                          | 4        |
| <b>2</b> | <b>Requirement Extraction</b>                           | <b>5</b> |
| 2.1      | Requirement elicitation . . . . .                       | 5        |
| 2.2      | Requirements specification . . . . .                    | 6        |
| 2.2.1    | Functional requirements . . . . .                       | 6        |
| 2.2.2    | Non-functional requirements . . . . .                   | 6        |
| 2.3      | Requirements verification and validation . . . . .      | 6        |
| 2.4      | Requirements management . . . . .                       | 6        |
| <b>3</b> | <b>Source code review and description of test cases</b> | <b>6</b> |
| <b>4</b> | <b>Conclusion</b>                                       | <b>7</b> |

# 1 Introduction

Software engineering best practices such as agile methodologies and DevOps are increasingly incorporated in the development and implementations of software in organisations. In scientific software engineering, for instance, scientific calculators have to support functionalities including evaluating transcendental functions. In this document we present an implementation of the function  $ab^x$ . In this section we describe the function and construct a context of use for the model, section 2 expresses the requirements of the function and assumptions made. In section 3, we describe the algorithm and its subordinate functions, and reasons behind the choices. We conclude the study in section 4.

## 1.1 Description of the function

In the function  $ab^x$  we have three parameters namely: the numbers  $a$  and  $b$ , and the variable  $x$  representing the power or exponent of the base number  $b$ . Hence, to evaluate this function we need to multiple two numbers  $a$  and  $b$  raised to the power of  $x$ .  $a$  and  $b$  are real constants with range  $(-\infty, +\infty)$  while  $x$  is a variable with the same range.

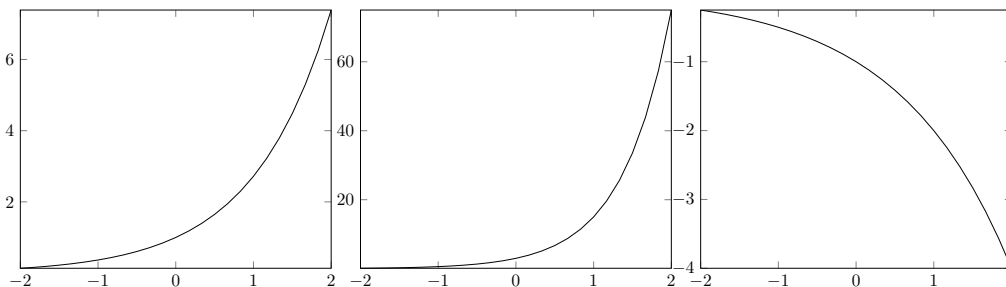


Figure 1: Illustration of examples  $e^x$ ,  $3(5^x)$ , and  $-2^x$  respectively with domain  $[-2, 2]$ .

In the function  $y = ab^x$ ,  $b^x$  represents an exponential function  $a$  is regarded as a scaling factor. The function contains a single term and also can not be simplified or reduced to an alternate form. Some examples are show in figure 1 and 2 below.

The range of  $y$  varies depending on various parameter as follows.

- When  $a < 0$ ,  $y \in (-\infty, 0)$ .
- When  $a > 0$ ,  $y \in (0, +\infty)$ .

- When  $a = 0$ ,  $y = 0$ , if  $b = 1$  then  $y = a$ .
- When  $b = e$ ,  $a = 1$ ,  $y = e^x$  (The exponential function.  $y \in (0, +\infty)$ ).

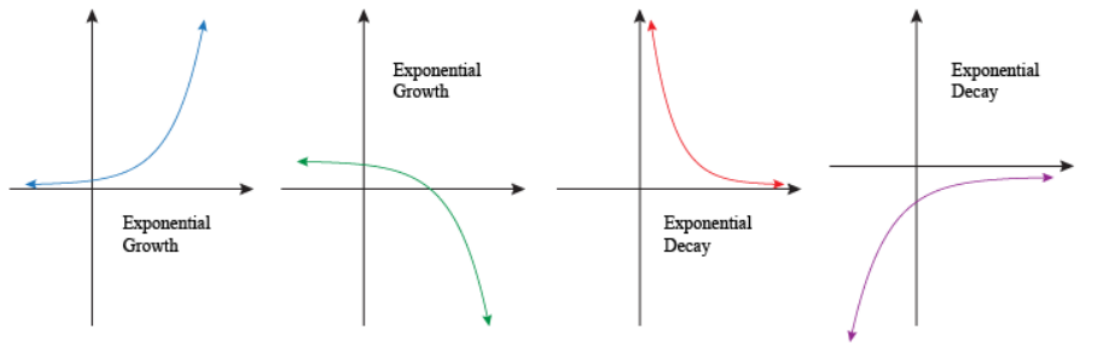


Figure 2: Typical variations of  $y = ab^x$  for standard trends.

## 1.2 Context of use model

The calculator application as described by the function can accommodate two types of user namely. Scientists (human users) and external programs through Application Program Interface (API) calls on subordinate functions.

The main function takes as input valid double data types for the three parameters  $a$ ,  $b$  and  $x$ . These values are entered by the user through the user interface and validated by the program and then passed into the main method. This method in turn calls subordinate function to obtain the relevant results. The calculator then return the final results of calculation to the user interface. This could be a value or simply a message indicating a calculation error. This process is illustrated in figure 3 below.

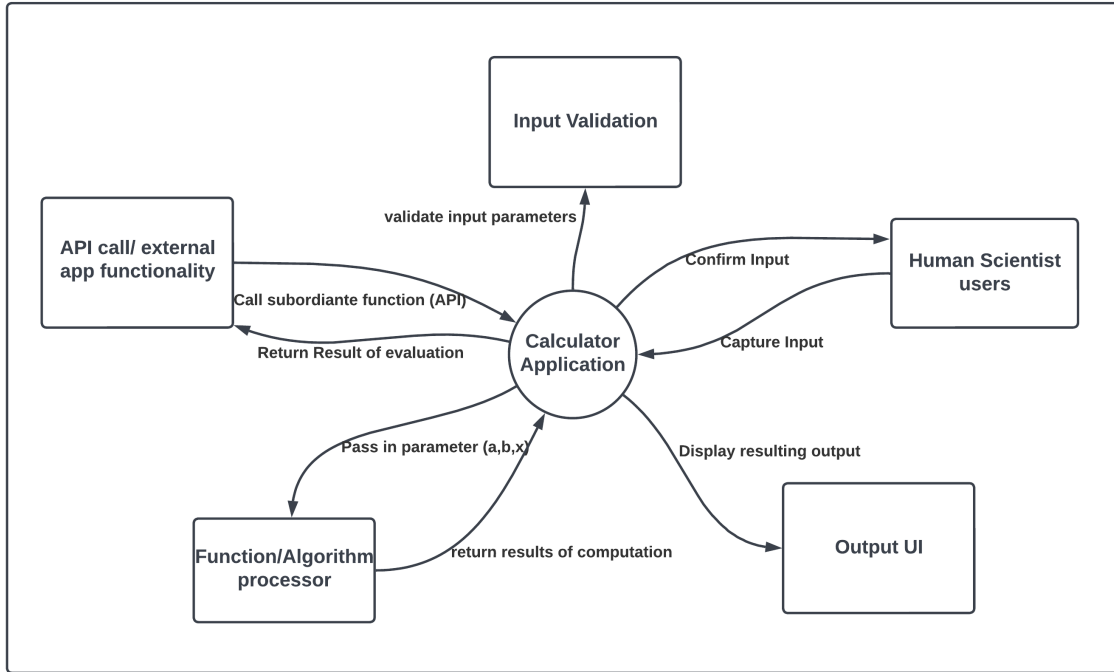


Figure 3: Context of use model for  $y = ab^x$

## 2 Requirement Extraction

In order to obtain a artefact through the software development process, it is necessary to formulate requirement of the system. These artefacts are usually the by-product produced during the development process (DevOps) include relevant designs diagrams and models use as described in the document. We shall consider four main activities in the requirement extraction process as follows.

### 2.1 Requirement elicitation

At this stage we communicated intensively with stakeholders. In this case this was done by reading project description, clarifying requirements with the lecturer and discussing the assignment with teaching assistant and other fellow student to have more insights on the problem. The various approaches to the implementation of subordinate function were investigated and constraint were defined. The context diagram was developed and the necessary literature was conducted for various implementation. The users include human

users and APIs. Mathematical concepts and implications were considered and there was revision of boundaries and constraints of the function  $y = ab^x$ . A draft of the UI design was discussed too with stakeholders At this stage there is a clear understanding of the problem.

## **2.2 Requirements specification**

Here we elaborate requirements formally based on the previous stage of the process. All requirements including functional and non-functional requirements, and constraints are specifies here. As studied in agile methodologies, these activities are carried out incrementally and iteratively as we develop the program by revising requirements. Git and Github was used as the version control system throughout the development

### **2.2.1 Functional requirements**

### **2.2.2 Non-functional requirements**

- The program must be user friendly and well documented using software engineering best practices.
- The solution implementation must be scalable for various mathematical calculations and maintainable over time.
- The program must be portable. The algorithms and source code for the program and document (Latex) must run on any integrated development environment for the specified language (Java).
- The algorithms for subordinate functions must be designed to attain high performance in terms of both space and time complexities.

## **2.3 Requirements verification and validation**

## **2.4 Requirements management**

# **3 Source code review and description of test cases**

First-level headings should be in bold.

---

**Algorithm 1** An algorithm with caption

---

**Require:**  $n \geq 0$ **Ensure:**  $y = x^n$  $y \leftarrow 1$  $X \leftarrow x$  $N \leftarrow n$ **while**  $N \neq 0$  **do**    **if**  $N$  is even **then**         $X \leftarrow X \times X$          $N \leftarrow \frac{N}{2}$ 

▷ This is a comment

**else if**  $N$  is odd **then**         $y \leftarrow y \times X$          $N \leftarrow N - 1$     **end if****end while**

---

## 4 Conclusion

In conclusion, we described, implemented and documented the scientific calculation of the function  $y = ab^x$  using software engineering best practices. The program was built conceptually and tested in Java programming language, we designed a also designed a context of use that fits with the implementation.

## References