Software Engineering Process
Topic: Final Deliverables
Prof. P. Kamthan

Amit Sachdeva
40084627
Due Date: 19/July/2019

Gamma Function

Problem 1

1 Introduction

Gamma Function: It is commonly referred as factorial function for complex numbers. It is derived by Daniel Bernoulli. The gamma function $\gamma(z)$ is defined for all complex values of z larger than zero. Complex number can be consist of real and imaginary number, like $z = a + ib$ in which a and b can real numbers. A complex number is typically written in the form where sigma a is the real part and it is the imaginary part.

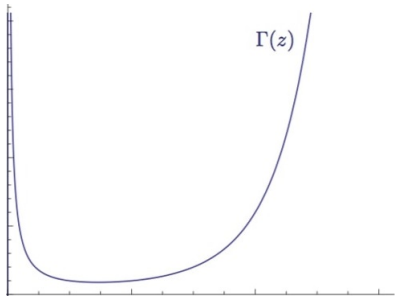

Figure 1: Gamma Function

2 Overall Description

This is a project based on gamma function in which we are making calculator for gamma value. User can insert any real value and expect real value except on boundary conditions.

3 Stakeholders

Users 1: This function is mostly used in physics calculations. So, most important stakeholders are scientists for their calculations.

Users 2: This function is also used in basic maths calculations or any analytically field.

4 Related to Function

4.1 Formulas

- **Formula1:** $\Gamma(x) = \int_0^\infty s^{x-1} e^{-s} ds \ \forall \ Re(x) > 0$
- **Formula2:** $\Gamma(1/2) = \sqrt{\pi}$
- **Formula3:** $n! = n * (n-1)!$

- **Formula4:** $\Gamma(x) = x\Gamma(x-1)$
- **Formula5:** $\Gamma(0) = undefined$

4.2 Popular Constant Values of Function

- **Constant 1:** $\Gamma(0) = undefined$
- **Constant 2:** $\Gamma(1) = 1$
- **Constant 3:** $\Gamma(2) = 1$
- **Constant 4:** $\Gamma(3) = 6$
- **Constant 5:** $\Gamma(3/2) = 0.886$
- **Constant 6:** $\Gamma(-3/2) = 2.36$
- **Constant 7:** $\Gamma(-1/2) = -3.54$

4.3 Domain of Function

$\forall$ Real numbers excluding negative values
$[0, \infty)$

4.4 Co domain of Function

- It ranges from $(0, \infty)$
- For positive integers, we returns integer value as normal factorial
- For other real numbers, we use integral function.

Problem 2

5 Requirements/Constraints of Function

5.1 Requirements

- **Req1**: For **Large input** in positive value, it will return infinity as **Const3**.
- **Req2**: For **negative input** $\forall x < 0$, **Function** will return **input error**, keeping in mind **Const1** and **Const2**
- **Req3**: For $x = 0$, **Function** will return **undefined**, keeping in mind **Const1**
- **Req4**: For $Re(x) > 0$, **Function** will return positive real value, keeping in mind **Const1**

5.2 Constraints

- **Constraint 1**: For Input, types must be Integer, Double, Float data types
- **Constraint 2**: We cannot **input** value of **non negative values**
- **Constraint 3**: We cannot input the value large positive number as it will return infinity as a programming language constraint

Problem 3

6 Algorithms

6.1 Pseudo Code 1

6.1.1 Algorithm

This algorithm is based on calculating based on using core integral using graph like dividing whole graph in small parts and calculating each part using formula of trapezium $(1/2 * (base1 + base2) * height)$ and combining it at the end.

function yAxisValue(Argument x, Argument s) {
    Calculate the value using $value = s^{x-1}e^{-s}$
    return value
end
}
function gammaFunction(Argument x) {
    if x $<$ 0
        then raise Input Error
    if x $>$ 170
        then return "Infinity"

    Initialize finalData with 0
    Set Interval for gap = 10^{-3}

    while loop i for range(0,Infinity)
        Add the finalData by using formula of trapezium using
        $1/2 * gap * (yAxisValue(i) + yAxisValue(i - gap))$
        increment i with gap value
    return finalData
end
}
{
In main function
    Take a input of x
    Call gammaFunction with input x-1 as a argument
end
}

6.1.2 Advantages

- Get More Precise Values for input as tested with existing results
- Using basic core approach of integration

6.1.3 Disadvantages

- We are iterating the loop at a large value so it takes time

6.2 Pseudo Code 2

6.2.1 Algorithm

This algorithm is based on calculating using pre-calculated values which act as a coefficients, iterating and calculating each values using input and coefficients

function gammaFunction(Argument x) {
    if x<0
        then raise Input Error
    if x>170
        then return "Infinity"

    declare array of finalArray precalculated values
    $temp = x + constant$
    Calculate temp value using log of temp value
    declare finalValue with value 1

    while loop i for range(0,length of finalArray)
        $finalValue+ = finalArray[index]/++temp$
    return finalValue
end
}
{
In main function
    Take a input of x
    Call gammaFunction with input x-1 as a argument
end
}

6.2.2 Advantages

- We are using constant pre-calculated value which increase the speed of algorithm
- This algorithm has less calculation which reduce complexity of code

6.2.3 Disadvantages

- We have generated constant value which not always give accurate value

7 References

- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4247832/
- https://medium.com/cantors-paradise/the-riemann-hypothesis-explained-fa01c1f75d3f