PROBLEM 1. PROJECT DESCRIPTION

The arccosine of x is defined as the inverse function of cosine of x where x lies in the range of

$-1 \leq x \leq 1$

Domain and range:

The domain of the arccosine function is from 1 to +1 inclusive and the range is from 0 to radians inclusive (or from 0 to 180). The arccosine functions can be extended to the complex numbers, in which case the domain is all complex numbers.

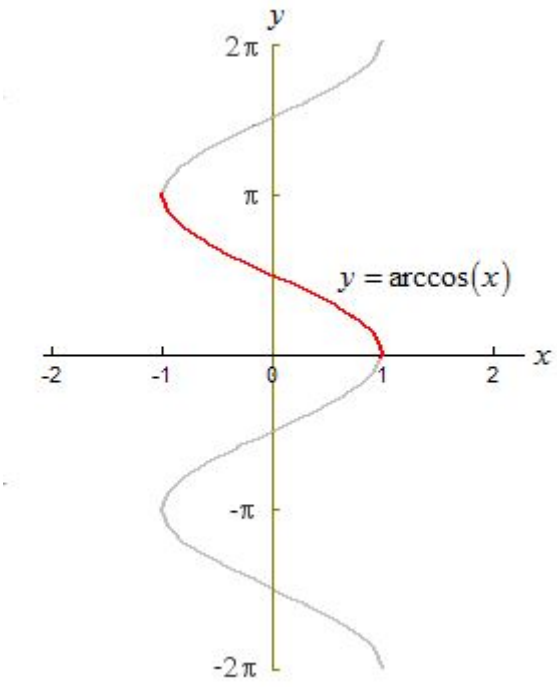

Figure 1: y = arccos(x)

Characteristics of the function y=arccos(x) that make it unique from another inverse

trigonometric functions:

1. Domain is in the range of [-1,1].

2. Range is part of [0, π].

3. It is neither even or odd function.

4. It is a decreasing function.

PROBLEM 2: REQUIREMENTS

1) The scope of arc cosine function

   a) Evaluates numerically
   b) Evaluates high precision numbers upon 6 digits after the decimal point
   c) Evaluates complex numbers
   d) Defined for all real values from the interval [-1, +1]
   e) Cannot deal with values at infinity
   f) Can deal with all the real values from the interval [0, π].
   g) Values of the arccosine function are fixed at few fixed points
   h) Value of the arccosine function at 1 is zero

2) Identification of Arccosine Function Requirements

   a) The result should always be in radians
   b) For real numbers ranging between -1 to +1, the results should always be in the range of [0, π].
   c) For certain special arguments, Arccos should automatically evaluate to the exact values
   d) The function should be suitable for numerical manipulation
   e) The function should be evaluated by arbitrary numerical precision
   f) The function should return an error message upon infinite value as input

3) Version Number

   All the Function requirements have to be implemented in the first version of the build delivery.

4) Owner

Kiranmayie

5) Stakeholder Priority

The requirements listed from 'a' to 'c' and 'f' are of high priority and the rest of the requirements are of low priority.

6) Risk

While implementing the requirements listed from 'a' to 'c', there are few possible issues. The major issues are the inverse of arc cosine function gives the output as the complex number. Generically $f^{-1}(f(z)) \notin z$. And near the branch cuts meaning from [∞ , -1] to [1, ∞], machine-precision inputs can give numerically wrong answers.

7) Rationale

The requirements identified in the Section 2 satisfy all the characteristics of the arc cosine function.

8) Difficulty

The requirements listed from 'a' to 'c' are difficult to implement as it involves invoking other functionalities such as PI and Strict Math.

9) Assumptions and Dependency

To reduce the difficulty and risks involved in this projects, we have assumed that the user doesn't give any inputs ranging outside the interval of arc cosine function.

PROBLEM 3: PSEUDOCODE

```
public double acos (double x)

{
//It returns inverse cosine value in degrees
// the returned angle is in the range 0.0 through pi
        // IF the argument i.e., x is NaN or its absolute value is greater than     1, THEN
```

```
            //it is applicable only for values between -1 to 1
            //if the value is other than this it gives error
      // RETURN input error "NaN"
  //ELSE
      // RETURN value of arc cosine of x
  //END IF
}
```

Advantages of the above Pseudocode format

Programming languages are difficult to read for most people, but pseudocode allows nonprogrammers, such as business analysts, to review the steps to confirm the proposed code matches the coding specifications. Some programmers write pseudocode in a separate document, while others write directly in the programming language using comments before the actual code. This provides a handy reference during coding.

Disadvantages of Pseudocode format

While pseudocode is easy to read, it does not provide as good a map for the programmer as a flowchart does. It does not include the full logic of the proposed code. Since it is basic by nature, pseudocode sometimes causes nonprogrammers to misunderstand the complexity of a coding project. Pseudocode is by nature unstructured, so the reader may not be able to see the logic in a step.

[References]

1) https://en.wikipedia.org/wiki/Inverse_trigonometric_functions

2) https://reference.wolfram.com/language/ref/ArcCos.html