Concordia University

SOEN 6011 - Software Engineering Process

# ETERNITY: FUNCTIONS
## arccos(x)

Deliverable 1

**Surya Prakash Govindaraju**
Student ID : 40085527
Team D

https://github.com/Surya64/SOEN-6011

# Contents

# 1 Introduction

## 1.1 Description

The arccos function is the inverse of the cosine function. It returns the angle whose cosine is a given number. The angle returned by this function is measured in radians or in degrees.The inverse is also called as acos or $cos^{-1}$.

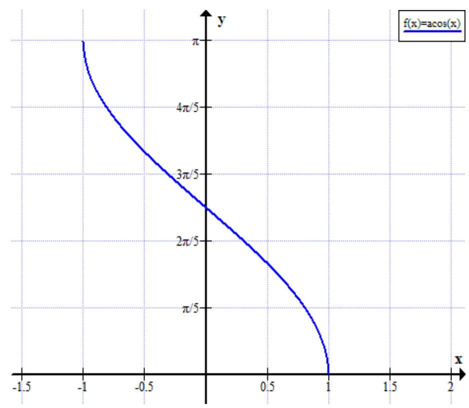


Figure 1: Graph of inverse cosine function (Source: Google Images)

arccos function is useful when trying to determine the remaining two angles of a right triangle when the length of sides is known. It is calculated as below

$$\theta = arccos(\frac{adjacent}{hypotenuse})$$

Derivative of arccos is given by,

$$\frac{d}{dx} arccos(x) = \frac{-1}{\sqrt{1-x^2}}$$

## 1.2 Domain

$$-1 \leq x \leq 1$$

## 1.3 Co-Domain

$$0 \leq y \leq \pi$$

or

$$0^\circ \leq y \leq 180^\circ$$

## 1.4 Characteristic

- Function is neither even nor odd
- Function is decreasing
- Ranges of the inverse function are subsets of the domain
- Function is defined for complex arguments
- Function is multi-valued, unless its principal value is defined

# 2 Functional Requirement

## 2.1 Assumptions

- ID: A1
  - Description: Input for the function arccos(x) is real number
- ID: A2
  - Description: Output of the function arccos(x) is in radians°
- ID: A3
  - Description: For values out of range, the result is undefined and throws error
- ID: A4
  - Description: The results obtained from approximation formulas will have some relative error.

## 2.2 Requirements

- ID: R1
  - Version: 1.0
  - Type: Functional Requirement
  - Priority: 1
  - Risk: High
  - Description: For any other input out of domain, the system should throw error
  - Rationale: Output is not defined for values out of domain range
- ID: R2
  - Version: 1.0
  - Type: Functional Requirement
  - Priority: 1

  - Risk: High
  - Description: There is no duplicate output value for an input within domain range
  - Rationale: one to one mapping between the domain and co domain

- ID: R3
  - Version: 1.0
  - Type: Functional Requirement
  - Priority: 1
  - Risk: High
  - Description: For valid input, the output is displayed in degree or radians
  - Rationale: Output is defined with in co domain range.

- ID: R4
  - Version: 1.0
  - Type: Functional Requirement
  - Priority: 2
  - Risk: medium
  - Description: User provides only one input value for the function
  - Rationale: x is the input for arccos

- ID: R5
  - Version: 1.0
  - Type: Non- Functional Requirement
  - Priority: 3
  - Risk: Low
  - Description: The calculation of the function should be completed within a desired time window
  - Rationale: To have a better performance

# 3 Algorithm

**Algorithm 1** Approximation algorithm for StrictMath

**if** $x$ is negative **then**
    $x = -x$
**if** $x$ is equal to 1 **then**
    **return** $negative\ ?PI : 0$
**if** $x$ is less than 0.5 **then**
    **if** $x$ less than $\frac{1}{K}$ **then**
        **return** $\frac{PI}{2}$
    z $= x * x$
    p $= z * (P0 + z * (P1 + z * (P2 + z * (P3 + z * (P4 + z * P5)))))$
    $q = 1 + z * (Q1 + z * (Q2 + z * (Q3 + z * Q4)))$
    $r = x - (PI_L/2 - x * (p/q))$
    **return** $negative?\frac{PI}{2+r} : \frac{PI}{2-r}$
**if** $x$ is negative **then**
    z $= (1 + x) * 0.5$
    p $= z * (P0 + z * (P1 + z * (P2 + z * (P3 + z * (P4 + z * P5)))))$
    $q = 1 + z * (Q1 + z * (Q2 + z * (Q3 + z * Q4)))$
    s $= sqrt(z)$
    w $= \frac{p}{q} * s - \frac{PI_L}{2}$
    **return** $PI - 2 * (s + w)$
    $z = (1 - x) * 0.5$
    $s = sqrt(z)$
    $df = (float)s$
    $c = (z - df * df)/(s + df)$
    p $= z * (P0 + z * (P1 + z * (P2 + z * (P3 + z * (P4 + z * P5)))))$
    $q = 1 + z * (Q1 + z * (Q2 + z * (Q3 + z * Q4)))$
    $w = p/q * s + c$
    **return** $2 * (df + w)$

**Algorithm 2** polynomial approximation algorithm using arctan

1: **if** $x$ is negative **then**
2: $\quad arctan(-x) = -arctan(x)$
3: **if** $x$ is greater than 1 **then**
4: $\quad$ **return** $arctan(x) = \frac{\pi}{2} - arctan(\frac{1}{x})$ ▷ To replace value bigger than 1
5: **if** $x > 2 - \sqrt{3}$ **then**
6: $\quad$ **return** $arctan(x) = \frac{\pi}{6} + arctan(\frac{\sqrt{3}x-1}{\sqrt{3}+x})$ ▷ To replace value bigger than $2 - \sqrt{3}$
7: $arctan(x) = x - \frac{x^3}{3} + \frac{x^5}{5}$
8: Substitute the $arctan(x)$ value in step 6
9: Results is in radians, convert by multiplying by $\frac{180}{\pi}$

## 3.1 Description

For calculating the inverse cosine of a value, above are the two approximation algorithms selected. Algorithm1 focus on the approximation that is derived from the Strict Math which basically focuses on constant variables used in the approximation. The algorithm also uses the square root function and the value of pi is declared as a constant variable. The results obtained are in radians and is a approximation value for arccos trigonometric function.

Algorithm2 is a polynomial approximation algorithm which is used to calculate the arccos function using the arctan function. From the Taylor's series of arctan(x), the value of arctan is calculated and obtained the result. The results are in radians and converted to degree by multiplying with $\frac{180}{\pi}$.

## 3.2 Advantages and Disadvantages

### Advantages

- Algorithm1 is provides the more accurate results for the value of x with in the domain
- Algorithm1 uses no other trigonometric function which will have no dependencies

- Performance is faster in algorithm1 than algorithm2 as estimated from the Big O Notation

**Disadvantages**

- For algorithm2, the higher order derivative of inverse trigonometric function becomes extremely complicated
- Algorithm2 approximation is only for small arguments and provides no accurate results for large arguments

# Bibliography


[1] Mathonweb,
http://mathonweb.com/help_ebook/html/algorithms.htm#arcsin

[2] Wolfram,
https://reference.wolfram.com/language/ref/ArcCos.html

[3] Wikipedia,
https://en.wikipedia.org/wiki/Taylor_series

[4] RapidTables,
https://www.rapidtables.com/math/trigonometry/arccos.html