

Answer to Problem 3

Arnab Roy

August 13, 2021

1 Algorithms

1.1 Pseudocode

1.1.1 The algorithm for $\sin(x)/\cos(x)$

Algorithm 1 $\text{tangent}(\text{angle})$ using $\sin(\text{angle})/\cos(\text{angle})$

```
1: result = calculateSin(angle)/calculateCos(angle)
2: if result is equal to  $\infty$  then
3:   throw error
4: end if
5: Round result to 7 digits
6: return result
```

Algorithm 2 calculateSin(angle)

```
1: convert angle to 360 degrees
2: if angle is less than 0 then
3:   store the sign of angle
4:   angle = absolute value of angle
5: end if
6: determine the quadrant of the angle
7: signInQuadrant = calculate the sign of the angle in determined quadrant
8: shouldSubtract = true
9: angleInRadian = angle * 3.141592654 / 180
10: squareOfRadianAngle = angleInRadian * angleInRadian
11: numerator = angleInRadian * angleInRadian * angleInRadian
12: digitToFactorial = 3
13: result = angleInRadian
14: for  $i = 1 \dots 9$  do
15:   nTerm = numerator / factorials[digitToFactorial]
16:   if shouldSubtract then
17:     nTerm *= -1
18:     shouldSubtract = false
19:   else
20:     shouldSubtract = true
21:   end if
22:   result += nTerm
23:   numerator = numerator * squareOfRadianAngle
24:   digitToFactorial += 2
25: end for
26: result = roundTo7Digits(result)
27: return result == 0 ? 0 : result * sign * signInQuadrant
```

Algorithm 3 calculateCos(angle)

```
1: convert angle to 360 degrees
2: if angle is less than 0 then
3:   angle = absolute value of angle
4: end if
5: determine the quadrant of the angle
6: signInQuadrant = calculate the sign of the angle in determined quadrant
7: shouldSubtract = true
8: angleInRadian = angle * 3.141592654 / 180
9: squareOfRadianAngle = angleInRadian * angleInRadian
10: numerator = squareOfRadianAngle
11: digitToFactorial = 2
12: result = 1
13: for  $i = 1 \dots 9$  do
14:   nTerm = numerator / factorials[digitToFactorial]
15:   if shouldSubtract then
16:     nTerm *= -1
17:     shouldSubtract = false
18:
19:   else
20:     shouldSubtract = true
21:   end if
22:   result += nTerm
23:   numerator = numerator * squareOfRadianAngle
24:   digitToFactorial += 2
25: end for
26: result = roundTo7Digits(result)
27: return result == 0 ? 0 : result * signInQuadrant
```

1.1.2 The algorithm for the Taylor series of tangent function

Algorithm 4 tangent(angle) using Taylor series

```
1: nominator[13] = [1, 1, 2, 17, 62, 1382, 21844, 929569, 6404582, 443861162, 18888466084,
   113927491862, 58870668456604]
2: denominator[13] = [1, 3, 15, 315, 2835, 155925, 6081075, 638512875, 10854718875,
   1856156927625, 194896477400625, 49308808782358125, 3698160658676859375]
3: result = 0
4: squareOfAngle = angle * angle
5: for test = 0 ... 12 do
6:   result += angle * nominator[test] / denominator[test]
7:   angle *= squareOfAngle
8: end for
9: return result
```

1.2 Description of the implemented algorithm

We have implemented the tangent function using the sin/cos formula. For this, we need the Taylor series for sine and cosine. At first, the angle in degrees provided by the user is converted within 90 degrees and the quadrant is determined. Then it is converted to radians. The algorithm considers upto 9 terms and achieves an accuracy upto 6 decimal digits.

1.2.1 Time complexity

$O(N)$

1.2.2 Space complexity

$O(T)$, where T is the number of factorials generated.

1.3 The advantages and disadvantages

We have implemented the tangent(angle) using the sin(angle)/cos(angle) algorithm. It has several advantages of the Taylor series of the tangent function. To understand this, we will first discuss about the disadvantage of the Taylor series of tangent.

1.3.1 Disadvantage of Taylor series of tangent

- The algorithm is not accurate for smaller terms. A large number of terms (approximately 30) needs to be taken to get an accurate result.
- The denominator of the 13th term of the series is 3698160658676859375. It is evident that the denominators of the larger terms will exceed the capacity of the primitive data types in Java.
- Keeping the Taylor series small results in output whose fractional parts are largely deviated from the accurate result.

1.3.2 Advantage of $\sin(x)/\cos(x)$

- Needs only 9 terms for the Taylor series of sine and cosine to make the output accurate upto 6 fractional parts
- Smaller number of terms mean faster execution