

Lab 10: Convergence of Iterative Algorithms and Equations

The Taylor Series for $\cos(x)$ is given by:

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \dots$$

- Fill in the Table 1 using the Taylor series estimate for $\cos(x)$ assuming $x = \pi$. Do these calculations by hand.

Table 1: Test Values

Initial	1 st Term	Estimate = 1
k = 1	Add Term 2	Estimate = Estimate \square $x^2/2!$ =
k = 2	Add Term 3	Estimate = Estimate + $x^4/4!$ =
k = 3	Add Term 4	Estimate = Estimate \square $x^6/6!$ =
k = 4	Add Term 5	Estimate = Estimate + $x^8/8!$ =

- Download the template script file and re-name it `Lab10_YourLastName`. Write a script file that prompts the user for the angle, x , in radians and the number of terms, N , to be used for the Taylor series estimate. Your script should compute an estimate for $\cos(x)$ and output the estimate with 5 places behind the decimal point.
- Test your script using the values you calculated in step 1. Correct your script if necessary.
- Now use your script to fill in Table 2.

Table 2: Estimates of $\cos(x)$ Using Taylor Series Algorithm

Angle, x , in radians	Number of Terms	Estimate Using Taylor Series	Actual Value
$\pi/2$	1		0
$\pi/2$	5		0
$\pi/2$	10		0
$5*\pi/4$	5		\square 0.70711
$5*\pi/4$	10		\square 0.70711
$5*\pi$	10		\square 1
$5*\pi$	20		\square 1
$5*\pi$	30		\square 1
\square 11.5 π	30		0
\square 11.5 π	40		0
\square 11.5 π	50		0
\square 11.5 π	53		0
\square 11.5 π	54		0
\square 11.5 π	55		0

$\pm 11.5\pi$	60		0
$\pm 11.5\pi$	100		0

Comment: For the last angle, once we get out to 100 terms, the values we need to compute become so large, we actually exceed the range of a 64 bit double. We will use the fact that cosine is a periodic function to fix this problem.

Periodicity of Trig Functions:

You can see that as the angle, x , moves away from 0 radians, we need more and more terms to get a decent estimate of $\cos(x)$. In fact, if x is too large, we actually exceed the range of a 64 bit double (the factorial gets huge!) and end up with NAN. We can fix this issue by making use of the periodicity of the cosine function. Since the cosine function is a periodic waveform with a period of 2π , we know the following is true:

$$\cos(x \pm N \cdot (2\pi)) = \cos(x)$$

So, we can take any angle, x , and reduce it to an angle in the range of $-\pi$ to π by adding or subtracting some integer multiple of 2π .

For example:

$$\begin{aligned} \cos(10\pi) &= \cos(0) \\ \cos(9.5\pi) &= \cos(1.5\pi) \\ \cos(-11.5\pi) &= \cos(-1.5\pi) \end{aligned}$$

Subtracting multiples of 2π is equivalent to dividing by 2π and keeping the remainder as the equivalent (smaller) angle. To see this complete the division problems below (by hand). The first one is done for you.

	$2\pi \overline{) \begin{array}{r} 5 \\ 10\pi \\ \underline{-10\pi} \\ 0 \end{array}}$	$2\pi \overline{) \begin{array}{r} 9.5\pi \\ \hline \end{array}}$
Remainder	0	_____

MATLAB has a function called *rem* that divides a number by another number and produces the remainder. Use the *rem* function to fill in Table 3.

Table 3: Examples of rem function

MATLAB COMMAND	Result	Result with π factored out
>> rem(10*pi, 2*pi)		
>> rem(9.5*pi, 2*pi)		

```
>> rem(-11.5*pi, 2*pi)
```

Comment: MATLAB has another function called *mod* that is very similar to *rem*. The only difference is *mod* produces the absolute value of the remainder. We will use *rem* because we want to preserve the sign of the remainder to get an equivalent angle.

5. Modify your script file to apply the *rem* function to the angle, *x*, that the user enters then calculate the Taylor series estimate for cosine based on the adjusted (or equivalent) angle.
6. Now fill in Table 6 using your modified program to verify that your program now works well for any angle, *x*.

Table 6: Estimates of cos(x) using equivalent angle

Angle, x, in radians	Number of Terms	Estimate Using Taylor Series	Actual Value
-11.5*pi	5		0
-11.5*pi	10		0
-11.5*pi	15		0

Convergence:

How do we know when we have used enough terms without knowing the actual value of cos(x)?

When we wrote the algorithm for estimating the square root of a number, we had a good measure of how close our estimate was using $error_sq = abs(Number - Estimate^2)$. We don't have anything like this that we can use for cos(x). This is the case for many iterative algorithms and iterative equations.

We will then test for convergence by comparing consecutive estimates. When the estimate doesn't change much from the previous estimate, we will stop our loop and output the estimate.

7. Modify your script file as follows:
 - Replace the for loop with a while loop. Your while loop should continue as long as $abs(Estimate - PreviousEstimate)$ exceeds 0.00001.
 - Eliminate your input statement that prompts the user for the number of terms.
 - Add an fprintf statement that displays how many terms were required for convergence.
8. Run your script to complete Table 7.

Table 7: Estimate of cos(x) and Number of Terms to Converge

Angle, x, in radians	Actual Value, cos(x)	Estimate, cos(x)	Number of Terms Required for Convergence
pi/2	0		
5*pi/4	-0.70711		
5*pi	1		
-11.5*pi	0		