

Using Numerical Methods to provide
Trigonometric Functions and Square Root

by

jameel sididiq

Preface

What is the background to this small e-book ?

Well, I sometimes use a Canon Starwriter 550C word-processor which also contains a spreadsheet option. The spreadsheet however is quite limited, with no trig or square root functions. The initial desire in going back to Numerical Methods that I had first been introduced to in school was to be able to add these functions to a spreadsheet if required. The extra challenge came from the fact that there was no conditional 'if' capability within the spreadsheet at all.

Although it should be obvious, **please note that unless otherwise stated when 'spreadsheet' is mentioned it refers specifically to my on limited spreadsheet on my word-processor and does not imply in any way that other spreadsheets suffer similar limitations.**

Can the methods be applied to other arenas? Well, of course! The first Java platform for Mobile Phone MIDP 1.0 was built on top of CLDC 1.0 where the Maths used is integer based. The first thing to do however, in such an environment, even before using these methods is to either import or build a floating point library. With devices such as mobile phones there may also be a considered trade-off between the number of steps used in a calculation and the overall time taken to return a result to the user.

For pcs, the techniques are not necessarily redundant as they could also be used to add functions that are not part of an original interpreter or compiler. In the past I have used a numerical method for square root to add such a function to a version of Logo did not have the function built in. Similarly, I've had a version of the Rexx Interpreter that contained no Trig function and could have used these methods to add these functions if required.

Any of the methods covered in this short paper will be far easier to implement within a proper programming environment than it was to implement within the very limited spreadsheet mentioned above.

This is an interactive e-book and so from within this PDF document you will have a chance to see the numerical methods in action.

I hope that this small e-book may prove to be of some interest.

Numerical Method for Sine and Cosine

The first thing to appreciate is that the numerical methods for sine and cosine depend on the angle being given in radians **and not degrees**.

The first job therefore is to convert the required angle. We are intending to use these methods only for the radian equivalent of angles between 0 and 180 degrees.

To convert an angle from degrees to radians you would use the following conversion formula :-

$$r = \frac{\pi\theta}{180}$$

where r is the angle in radians and theta is the angle in degrees.

Once the angle is in radians you can use the following series to evaluate the sine of the angle :-

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} + \dots$$

The number of terms used to determine your answer will depend on the level of accuracy you require or possibly the inherent accuracy limitations of the system you are using. The spreadsheet that I was using for example would not allow the display or entry (for pi say) of more than four decimal places.

On the next page is an interactive example that you use to see for yourself how this works

- Enter an angle in degrees between 0 and 90^0 below
- Click the convert button when done
- Right, your angle in radians is
- The 1st term of the series is
- The 2nd term of the series is
- The 3rd term of the series is
- The 4th term of the series is
- The four terms sum to
- The actual answer is

Click the CLEAR button to try this again

The series for cosine is slightly different and is shown below :-

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

Again, the number of terms used to determine your answer will depend on the level of accuracy you require or the inherent accuracy limitations of the system you are using.

Below, as before, on the next page you can see how this works for yourself.

- Enter an angle in degrees between 0 and 90^0 below
- Click the convert button when done
- Right, your angle in radians is
- The 1st term of the series is
- The 2nd term of the series is
- The 3rd term of the series is
- The 4th term of the series is
- The four terms sum to
- The actual answer is

Click the CLEAR button to try this again

Below is a screenshots of how I used the series formulae for sine and cosine to evaluate trigonometric values within my spreadsheet.

	A	B	C	D	E	F	G	H	I	J	K
1	Your sums in rows 2 to 4										
2	Example: $45*\sin(30)+60*\cos(45)$										
3	Answer: 64.932										
4											
5	Angle **	SIN	COS	TAN	**Enter in Degrees						
6	30	0.5	0.866	0.5774							
7	45	0.7072	0.7072	1							
8	60	0.866	0.5	1.732							
9	90	0.9998	-0.0009	-1110.89							
10	Trig Functions derived below:-			Sin sub-calculations			Cos sub-c			Cos sub-calculations	
11	0.5236	0.5	0.866	0.4997	0.0003	0	0.8629	0.0031	0	0	
12	0.7854	0.7072	0.7072	0.7047	0.0025	0	0.6916	0.0159	0.0003	0	
13	1.0472	0.866	0.5	0.8558	0.0105	0.0003	0.4517	0.0501	0.0018	0	
14	1.5708	0.9998	-0.0009	0.9248	0.0797	0.0047	-0.2337	0.2537	0.0209	0	

The limitation on formula length meant that I had to break up the series and evaluate it in parts and then sum the separate parts for the final answer, hence three columns for both sine and cosine evaluation. Although not coloured on the original spreadsheet the cells in yellow are where the angle values are entered and the cells in green indicate where all the number-crunching takes place. Column A in the green area is where the angles are converted to radians.

In case you were wondering the original spreadsheet was exported as a csv file, imported into excel, cells coloured for guidance and a screen-capture made and the final editing performed in Corel Paint.

In the next section we will look at evaluating arcsine (the inverse of sine) and arccosine (the inverse of cosine).

Numerical method for arcSine and arcCosine

For arcsine and arccosine we will use the Newton-Raphson iterative formula, which can be applied to a whole host of problems.

For applying this technique we need a formula that evaluates to zero.

If $\sin(x) = v$, where 'v' is a number that we actually have then

$$\sin(x) - v = 0$$

The value 'x' when solved will actually be $\arcsin(v)$ which is what we want.

This becomes the function that we want to find the root of.

In other words :-

$$f(x) = \sin(x) - v = 0$$

If 'x' is our guess (or seed) for the answer, then we use the following formula iteratively until we arrive at an answer to the required accuracy. Again the level of accuracy may naturally be inhibited by the system that we are using.

The iterative formula used for the Newton-Raphson iterative is :-

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

In this case

$$f(x_n) = \sin(x_n) - v$$

'v' the known number will be a constant

$f'(x)$ is the derivative of $f(x)$

$$\frac{d f(x_n)}{dx} = \cos(x_n)$$

The iterative solution for $\arcsin(x)$ becomes :-

$$x_{n+1} = x_n - \frac{(\sin(x_n) - v)}{\cos(x_n)}$$

You would of course be solving the solution in radians.

To convert the answer found in radians to degrees using the following conversion process :-

$$\theta = \frac{180}{\pi} r$$

Below we will see how this works, specifically for arcsine. The Javascript that sits behind this document will use the series formula to work out the relevant values for sine and cosine and we will go through a number of steps to illustrate how the method converges towards the answer.

Please turn over to the next page.

- Enter a value between 0 and 1 below
- Click the solve button when done
- Below are the steps to find the arcsine of the given value
- Starting with a seed of 45°
- This is the second approximation =
- This is the third approximation =
- This is the fourth approximation =
- This is the fifth approximation =
- And finally here is the sixth approximation =
- Click CANCEL to restart

The value for arcsine that has been derived in radians and then converted back to degrees is a remarkably good 60 degrees.

Arccosine is not so good with the accuracy only being to one decimal place. Within an interpretative language further steps could have been executed for higher accuracy. Here the spreadsheet is limited to 4 decimal place for entry and display although it is not clear to what number of places an number may be held internally.

I changed the seed value (in cell A6 on the spreadsheet) from 45 degrees to 30 degrees and got a better result, accurate in fact, for both sums.

Numerical method for Square Root

The numerical method for square root will again use the Newton-Raphson iterative method.

If 'x' is the square root of our known value 'v' then

$$x^2 - v = 0$$

Therefore

$$f(x) = x^2 - v$$

$$f'(x) = 2x$$

Therefore the iterative formula in this case will be :-

$$x_{n+1} = x_n - \frac{(x_n^2 - v)}{2x_n}$$

We will see on the next page how this works in practice.

- Enter a value, 'v', between 10 and 100 below
- Click the solve button when done
- Starting of halve of 'v',
- This is the second approximation =
- This is the third approximation =
- This is the fourth approximation =
- This is the fifth approximation =
- And finally here is the sixth approximation =
- Click CANCEL to restart

Finally, below is a copy of a spreadsheet in which this technique was used.

	A	B	C	D
1	Square Root			
2				
3	Number (n):	8.25		
4	Sq Root:	2.8723		
5				
6	Working:-			
7				
8	Equation to solve is			
9			x^2	
10		x	$- \text{given number}$	$= 0$
11				
12	start with $n/2$		$f(x)$	$f(x)$
13	4.125		8.7656	8.25
14	3.0625		1.1289	6.125
15	2.8782		0.034	5.7564
16	2.8723		0.0001	5.7446
17	2.8723		0.0001	5.7446
18	2.8723		0.0001	5.7446
19	2.8723		0.0001	5.7446
20	2.8723		0.0001	5.7446

The square root of 8.25 is 2.8723 to four decimal places, so not bad at all.

Conclusion

As I was going through I hinted at possible accuracy limitations and this does need to be kept in mind.

If you look at the first spreadsheet, as an example, you'll see that the sine of 90 degrees is 0.9998 and the cosine of 90 degrees is -0.0009, whereas we know that their respective values are zero and unity (ie. 0 and 1). If we took our answers to three decimal places, then we would be ok. In practice, in the case of my spreadsheets, it would probably be best to use two decimal places. Even then, when using the Newton-Raphson method, as in the example of arccosine in the second spreadsheet, depending on the trial number used we may have got less than the desired level of accuracy. For an iterative method like Newton-Raphson, if we are really worried about accuracy we can feed our answer back into the iterative process to either confirm or further improve the accuracy of the result we have.

We would also have to be conscious that the technique is best suited for short calculations where no more than one or two of the constituent parts of the formula to be calculated using a numerical method. If we were to attempt a lengthy calculation where many of the constituent terms or steps had to be evaluated using numerical methods then the cumulative effect of the usually small inaccuracies would combine cumulatively and possibly in such a way as to render our calculation meaningless.

So the technique is useful if applied judiciously and in the full knowledge of its limitations.

- The End -