

Assignment 1

problems by Branko Ćurgus

posted on 20170720

Problem 1. Funny trigonometry and its beauty

■ Few *Mathematica* comments

There are several standard *Mathematica* functions that can be useful here. For example, **Floor**, **Sin**, **ArcSin**, **Round**, **Mod**, **Sign**, **Clip**, **Min**, **Max**, or you can use other functions. But, have in mind that functions in this problem must be defined for **all** real numbers. A particularly powerful function here is **Mod**. Study it in the help file and understand what this function does.

To get nice graphs I used the following options in **Plot** command:

```
PlotStyle → {Thickness[0.004]}, Axes → True,  
PlotRange → {{-4.5, 4.5}, {-1.5, 1.5}}, AspectRatio → Automatic,  
ImageSize → 450, Ticks → {Range[-6, 6], Range[-6, 6]}
```

In addition to the **Plot** command you should use **ParametricPlot**, **ParametricPlot3D**, and **Plot3D** where appropriate. Or, you can even use just **Graphics** when the previous functions do not give satisfactory results.

■ Definitions

Functions f and g are called a **funny cosine** and a **funny sine** if they satisfy the following properties:

Both functions f and g are defined for **all** real numbers.

Both functions f and g are **periodic** with period $2p$ where $p > 0$.

The function f is **even** and the function g **odd**.

The function f is a **shift** of g to the left by $\frac{p}{2}$, that is $f(x) = g\left(x + \frac{p}{2}\right)$.

The functions f and g satisfy $f(0) = 1$, $f'(0) = 0$ and $g(0) = 0$, $g'(0) = 1$ (in some examples we will allow that $f'(0)$ and/or $g'(0)$ to be undefined).

The most popular way of defining **Cos** and **Sin** is on the unit circle. That is, the plot of the equation

$$\{\text{Cos}[t], \text{Sin}[t]\}, 0 \leq t \leq 2\pi,$$

is the unit circle. For a pair of a **funny cosine** f and a **funny sine** g the set

$$\{f[t], g[t]\}, 0 \leq t \leq 2p,$$

will be called the **funny unit circle**.

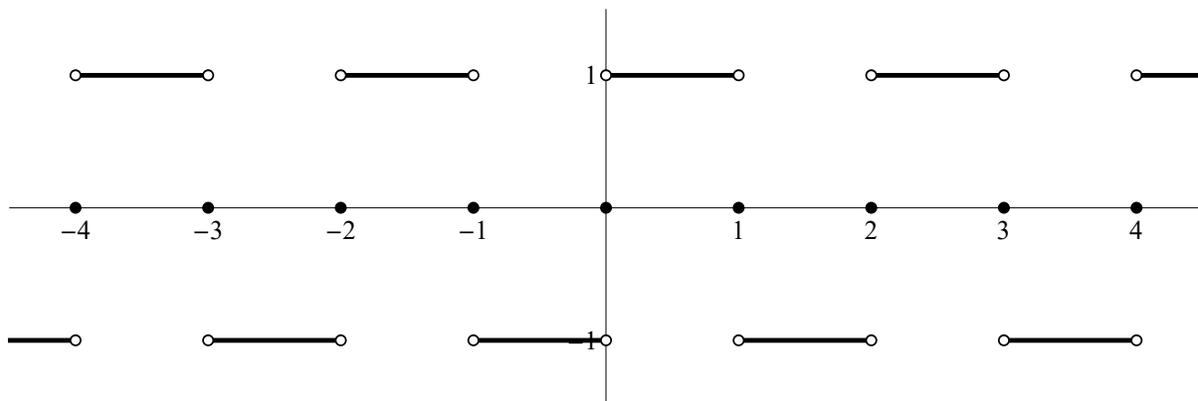
■ Specific examples

Your task in the specific examples below is the following: For each given graph discover a **formula** for the function whose graphs is given. It does not suffice to simply re-produce the graphs (although that is required). You must find an explicit **formula** in terms of \mathbf{x} , say **MyS** [\mathbf{x}], so that you can plug in *any* real number and find the value of the function; for example **MyS** [$10 * \mathbf{E} * \mathbf{Pi}$] should give a specific number. Also, you should make sure that your formula at appropriate points gives the **exact** values not some close approximations. For this you should use **Table** [], that is an easy way to test a lot of numbers with one command.

Please notice that the numbering of the Examples below starts from -1, not the usual starting number 1.

■ Example -1. A square cosine and a square sine

(a) Our first example is a discontinuous funny sine, call it **SquareSin** (briefly **SqSin**). The graph of this function is given below



Plot the functions (on separate plots, not together) **SqSin** and **SqCos** using the **Plot** command. This is an opportunity to learn about **Exclusions** option of the **Plot** command.

The default setting is **Exclusions** \rightarrow **Automatic**. This should work in most cases. Check what happens when you set **Exclusions** \rightarrow **None**. If the default setting does not produce desirable results you need to specify which points need to be excluded. For example, **Exclusions** \rightarrow **Range**[-5, 5].

Since the above function has **jump discontinuities** we need to add the points which specify the exact values of the function at the discontinuities. We do that with the option **Epilog** \rightarrow {} and the table of points. A trick of making a small circle is to overlay a larger black point with a smaller white point. For example,

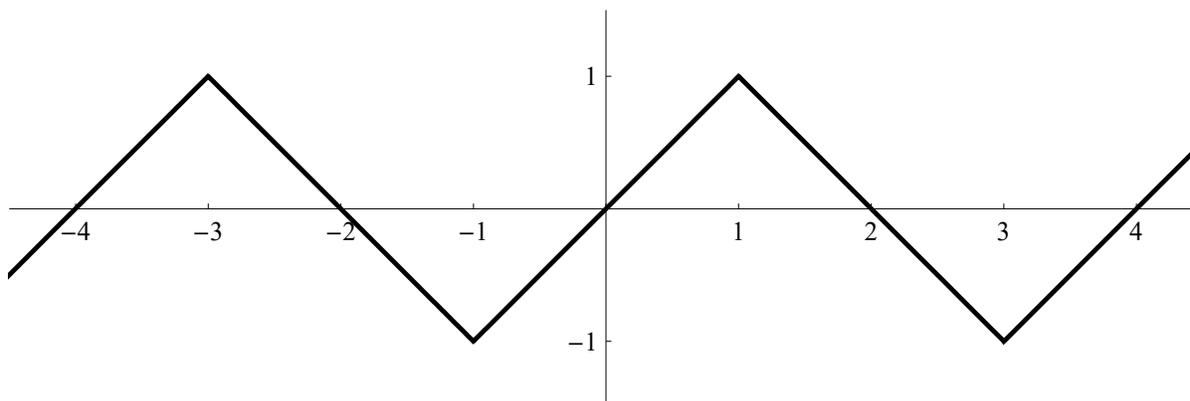
```
{PointSize[0.02], Black, Point[{0, 1}]},
  {PointSize[0.015], White, Point[{0, 1]}}
```

is one of the points on the graph above. You might need to experiment with the size of the points to get the best results. Since you have to make a lot of points like this, you should use **Table** [] to plot them efficiently.

(b) Plot the **exact** graph of the funny unit circle in this case. (Please do this question “by hand” first. Do not let yourself be deceived by *Mathematica*.)

■ Example 0. A line cosine and a line sine

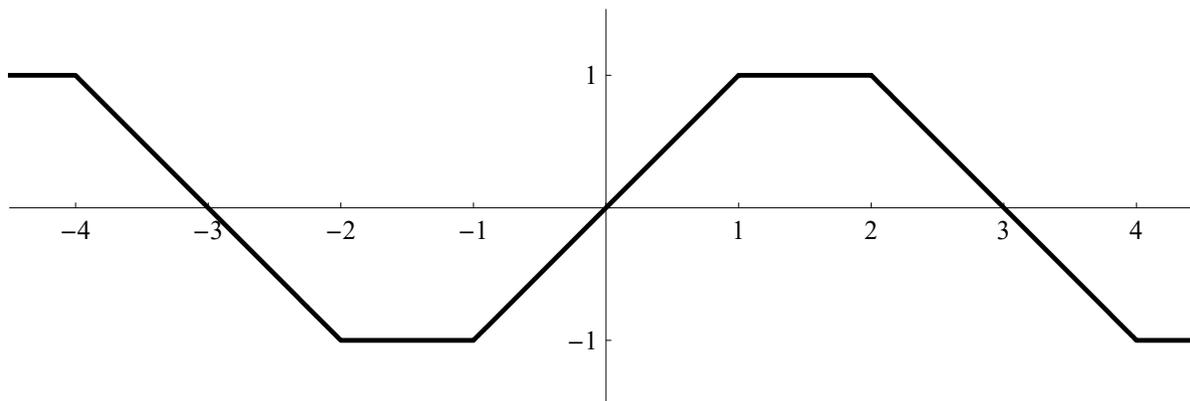
(a) Below I give a graph of a funny sine function. I call it **LineSin**, or briefly **LiSin**. Discover a formula which I used for **LiSin**. Provide a graph similar to one presented below. This function has the exact values -1, 0, 1 at the appropriate integers and it is a continuous and piecewise linear function. Give a formula for **LiCos**. On a separate Plot **LiCos**.



(b) Plot the graph of the funny unit circle in this case. Please no overlaps. On separate three graphs show “one quarter”, “one half” and “three quarters” of this funny unit circle.

■ Example 1. A trapezoidal cosine and a trapezoidal sine

(a) Below I give a graph of yet another funny sine like function. I call it **TrapezoidalSin**, or briefly **TrapSin**. Discover a formula which I used for **TrapSin**. Provide a graph similar to one presented below. Again, this function has the exact values -1, 0, 1 at the appropriate integers and it is a continuous and piecewise linear function. Find a formula and plot **TrapCos** on separate plot.

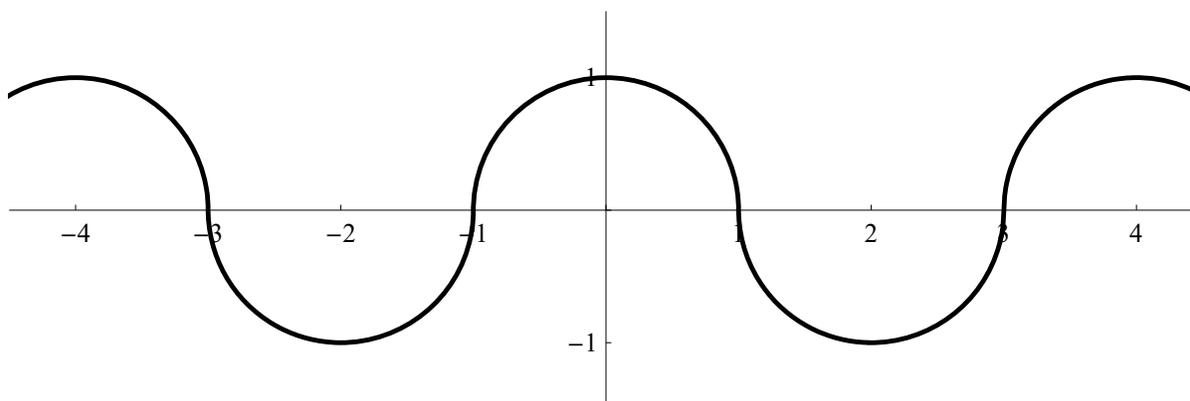


(b) Plot the exact graph of the funny unit circle in this case. Please no overlaps. On separate three graphs show “one quarter”, “one half” and “three quarters” of this funny unit circle.

■ Example 2. A round cosine and a round sine

(a) Here is yet another funny cosine function. I call it **RoundCos**, or briefly **RoCos**.

Discover a formula which I used for **RoCos**. This function has the exact values -1, 0, 1 at the appropriate integers. The round pieces on the graph below are halves of unit circles. Provide a graph similar to one presented below. Also, find a formula for **RoSin** and graph it on a separate graph.



(b) Plot the exact graph of the funny unit circle in this case. Please no overlaps. On separate three graphs show “one quarter”, “one half” and “three quarters” of this funny unit circle.

■ The beauty of funny trigonometry

Every student should do all four Examples -1, 0, 1 and 2 above. However, in this part of the problem each student will select one pair of funny trigonometric functions and explore the beauty of this pair further. Start with your W number. Denote the numerical part of your W number by **myWno**. For example, if your W number is W12345678, then **myWno = 12 345 678**.

First determine the remainder after **myWno** is divided by 3. Call this number **myn**. You can calculate **myn** using *Mathematica* **myn = Mod[myWno, 3]**. Thus **myn = 0** or **myn = 1** or **myn = 2**.

Second read my file `TheBeautyOfTrigonometry.nb`.

Finally, your task in this item.

■ **If `myn = 0` read this subsection.**

Your task in this item is to reproduce all the explorations done in my file `TheBeautyOfTrigonometry.nb` with cosine and sine replaced by the funny cosine and funny sine defined in **Example 0**. Pay attention to details in each command. There is a lot that you can copy from my file, but you have to make **same** changes to get pictures that look nice. In particular pay attention to domains for different variables. In my file those domains are related to `Pi`. In your file you should use the appropriate number. Your grade for this part will be determined on how well you adjusted my commands for the task at hand.

■ **If `myn = 1` read this subsection.**

Your task in this item is to reproduce all the explorations done in my file `TheBeautyOfTrigonometry.nb` with cosine and sine replaced by the funny cosine and funny sine defined in **Example 1**. Pay attention to details in each command. There is a lot that you can copy from my file, but you have to make **same** changes to get pictures that look nice. In particular pay attention to domains for different variables. In my file those domains are related to `Pi`. In your file you should use the appropriate number. Your grade for this part will be determined on how well you adjusted my commands for the task at hand.

■ **If `myn = 2` read this subsection.**

Your task in this item is to reproduce all the explorations done in my file `TheBeautyOfTrigonometry.nb` with cosine and sine replaced by the funny cosine and funny sine defined in **Example 2**. Pay attention to details in each command. There is a lot that you can copy from my file, but you have to make **same** changes to get pictures that look nice. In particular pay attention to domains for different variables. In my file those domains are related to `Pi`. In your file you should use the appropriate number. Your grade for this part will be determined on how well you adjusted my commands for the task at hand.

Problem 2. Funny factorials

Recall the recursive definition of the function `MyFactorial` in `Primer.nb`:

```
Clear[MyFactorial, n];
MyFactorial[1] = 1;
MyFactorial[n_Integer] /; (n > 0) :=
  MyFactorial[n] = MyFactorial[n - 1] * n

Table[MyFactorial[k], {k, 1, 23}]
```

The following definition is equivalent to the definition of **MyFactorial**.

```
Clear[MyF1, n];
MyF1[1] = 1;
MyF1[n_Integer] /; (n > 0) := MyF1[n] = n * MyF1[n - 1]

Table[MyF1[k], {k, 1, 23}]
```

■ The double factorial

(a) The goal here is to define the function **DoubleFactorial** (briefly **DF**). The domain of this function is the set of nonnegative integers. As with the standard factorial we have **DF[0] = 1**. For a positive even integer **n** the value **DF[n]** is the product of all positive even integers which are $\leq n$. For a positive odd integer **n** the value **DF[n]** is the product of all positive odd integers which are $\leq n$.

Give a recursive definition of **DF**. Try to make this function as similar to the above definition of **MyFactorial**. Do not use **If** or other branching functions.

■ (b) The "sum" factorial

(b-1) In the above definition of **MyFactorial** replace the multiplication sign with the addition sign. Call this new function **MySF**.

(b-2) Explore the function that you just created and discover a simple formula for **MySF[n]** in terms of **n**. This should be a closed form expression in **n**. Provide a table which illustrates that your discovery is correct.

■ (c) Two "subtraction" factorials

(c-1) In the above definitions of **MyFactorial** and **MyF1** replace the multiplication sign with the minus sign. Call these new functions **MyMF** and **MyMF1**.

(c-2) Explore the functions that you created in (c-1) and discover simple formulas for **MyMF[n]** and **MyMF1[n]** in terms of **n**. These should be closed form expressions in **n**. Provide tables which illustrate that your discoveries are correct.

■ (d) Two "mixed" factorials

(d-1) In the above definitions of **MyFactorial** and **MyF1** replace the multiplication sign by the minus sign at even steps and the plus sign at odd steps. Call these new functions **MyMPF** and **MyMPF1**.

(d-2) Explore the functions that you created in (d-1) and discover simple formulas for **MyMPF[n]** and **MyMPF1[n]** in terms of **n**. Ideally these should be closed form

expressions in n . Provide tables which illustrate that your discoveries are correct.

■ (e) Two "division" factorials

(e-1) In the above definitions of **MyFactorial** and **MyF1** replace the multiplication sign with the division sign. Call these new functions **MyDivF** and **MyDivF1**.

(e-2) For each function created in (e-1) discover a formula that connects that function to the function created in (a) and/or the standard factorial. Provide tables which illustrate that your discoveries are correct.

■ (f) Two other "mixed" factorials

(f-1) In the above definitions of **MyFactorial** and **MyF1** replace the multiplication sign with the division sign at even steps and by multiplication sign at odd steps. Call these new functions **MyDivMF** and **MyDivMF1**.

(f-2) For each function created in (f-1) discover a formula that produces the same values using the standard factorial, the double factorial or a similar factorial like function that you invent.

Problem 3. Ramanujan inspired identities

In the paper

Berndt, B.C.; Zhang, L.C.:

A new class of theta-function identities originating in Ramanujan's notebooks.

J. Number Theory 48 (1994), no. 2, 224–242,

authors, among many other things, discovered the following identity

$$\frac{(\text{Sin}[3\pi/7])^2}{(\text{Sin}[\pi/7])^4} + \frac{(\text{Sin}[\pi/7])^2}{(\text{Sin}[2\pi/7])^4} + \frac{(\text{Sin}[2\pi/7])^2}{(\text{Sin}[3\pi/7])^4} = 28$$

An interesting feature here is that a relatively complicated expression involving the sine function turns out to be an integer. Notice that in the fractional expressions involving π the denominator is always the same and the numerators are the numbers 1, 2, 3. That is, we consider the fractions

$\frac{1}{7}$, $\frac{2}{7}$, $\frac{3}{7}$ of π .

In this problem I ask you to explore possible new identities that are inspired by the above identity.

■ Part A

- 1. “Prove” the above identity using *Mathematica*.
- 2. Explore whether similar identities hold for other permutations of the numbers 1, 2, 3 in the sine functions in the **numerators**. The permutation in the original expression is {3, 1, 2}. Explore the other five permutations {1, 2, 3}, {2, 3, 1}, {2, 1, 3}, {1, 3, 2}, {3, 2, 1}. For which of these five permutations you get a simple expression for the sum? Do you see anything in common for those permutations that give simple sums? Give complete and clear answers.
- 3. (a) State the analogous expressions when 7 is replaced by 5 and explore them. Provide a clear summary of your explorations.
(b) State the analogous expressions when 7 is replaced by 9 and explore them. Provide a clear summary of your explorations.
(Please think carefully and critically what “analogous” means here.)
- 4. Explore the corresponding questions for the cosine function. Summarize your findings clearly.

■ Part B

■ 1.

The sum associated with the permutation {1,2,3} studied in Part A-2 reduces to the sum of the reciprocals of the squares of the sine function. Namely, we have the identity

$$\frac{(\text{Sin}[\pi/7])^2}{(\text{Sin}[\pi/7])^4} + \frac{(\text{Sin}[2\pi/7])^2}{(\text{Sin}[2\pi/7])^4} + \frac{(\text{Sin}[3\pi/7])^2}{(\text{Sin}[3\pi/7])^4} =$$

$$\frac{1}{(\text{Sin}[\pi/7])^2} + \frac{1}{(\text{Sin}[2\pi/7])^2} + \frac{1}{(\text{Sin}[3\pi/7])^2}$$

Since

$$\frac{1}{\text{Sin}[x]} = \text{Csc}[x]$$

we have

$$\frac{(\text{Sin}[\pi/7])^2}{(\text{Sin}[\pi/7])^4} + \frac{(\text{Sin}[2\pi/7])^2}{(\text{Sin}[2\pi/7])^4} + \frac{(\text{Sin}[3\pi/7])^2}{(\text{Sin}[3\pi/7])^4} =$$

$$(\text{Csc}[\pi/7])^2 + (\text{Csc}[2\pi/7])^2 + (\text{Csc}[3\pi/7])^2$$

Therefore it is natural to explore simplifications of the sums of squares of the cosecant function evaluated at fractions of the form $\frac{k\pi}{n}$ where n is a fixed integer and k is such that

$$\frac{k\pi}{n} \in \left(0, \frac{\pi}{2}\right).$$

Explore simplifications of the sums of squares of the cosecant function evaluated at the fractions of the form $\frac{k\pi}{n}$ where n is a fixed integer greater than 3 and k a positive integer such that $\frac{k\pi}{n} \in \left(0, \frac{\pi}{2}\right)$. For example, for $n = 3$ we have

$$(\text{Csc}[1 \pi / 3])^2$$

$$\frac{4}{3}$$

for $n = 4$ we have

$$(\text{Csc}[1 \pi / 4])^2$$

$$2$$

for $n = 5$ we have

$$\text{FullSimplify}[(\text{Csc}[1 \pi / 5])^2 + (\text{Csc}[2 \pi / 5])^2]$$

$$4$$

Explore other n -s; $n = 6, 7, 8, 9, 10, \dots$ and so on. Explore as many n -s as *Mathematica* would do. Try to find a functional relationship between the value of n and the simplified value of the sum. Formulate your conjecture clearly and illustrate that it might be true by *Mathematica* calculations. Considering even and odd values of n separately can be helpful here.

- **2.** Repeat Part **B-1** for the function secant.
- **3.** Do you notice any similarities between what you found in Parts **B-1** and **B-2**? Clearly formulate your findings and try to explain mathematically.

Problem 4. Three points in search of a parabola

In this problem we define:

Let a, b, c be arbitrary real numbers. In this problem, a *parabola* is a set of points in the plane defined by $\{t, at^2 + bt + c\}$ where $t \in \mathbb{R}$ is an arbitrary real number.

■ The first part

(1a) Given three points $\{1, 1\}$, $\{2, 3\}$ and $\{3, 7\}$ we want to find the parabola passing through these three points. To this end, write down a system of three equations with three unknowns (call the unknowns \mathbf{a} , \mathbf{b} , \mathbf{c}).

(1b) Use `Solve[]` to solve the system of equations obtained in (a).

(1c) Define a function, say `p1[t]`, using your parabola formula and evaluate `p1` at $\mathbf{t} = 1, 2, 3$, to check that your parabola does pass through the right points.

(1d) Check visually that the parabola passes through the points. Use `Plot[]` and its

Options[] to do this. In particular you should use the option **Prolog** to include the graphics primitive **Point** into **Plot[]**.

- To make the points easier to see add this, for example,

```
{PointSize[.02], RGBColor[0, 1, 0], Point[{1, 1}]}
```

Here **PointSize** makes the points larger (you can experiment with the number in square brackets to see how it works--the number specifies the ratio of width of point to width of the entire plot) and **RGBColor[1,0,0]** makes them green. (**RGBColor[1,0,0]** would make red dots and **RGBColor[0,0,1]** blue dots. Making more than one entry nonzero mixes the colors. For example **RGBColor[0,1,1]** is cyan.)

■ The second part

(2a) The next step is to automate the process from the first part: Define a function whose inputs are three points and the output is the formula for the parabola through these three points. For example, name the points **{Ax, Ay}**, **{Bx, By}** and **{Cx, Cy}**. Then the formula for the parabola should look like

```
p3p[{Ax_, Ay_}, {Bx_, By_}, {Cx_, Cy_}, t_] := ....
```

(2b) For certain points **{Ax, Ay}**, **{Bx, By}**, **{Cx, Cy}** the parabola in (2a) does not exist. In other words the function **p3p**, is not defined for such points. Find an algebraic condition on **{Ax, Ay}**, **{Bx, By}** and **{Cx, Cy}** under which the parabola in (2a) does not exist.

(2c) The final step is to automate the process from (b). Use **Manipulate[]** to demonstrate that your command produces parabolas for different input points.

(2d) Refine your command in (2c) to include the case discovered in (2b). In your **Manipulate[]** command you should incorporate **If[]** statement which should prevent *Mathematica* of trying to plot a parabola when parabola does not exist.

(2e) Use **Manipulate** to show that your construction works in all cases.