*Updated on September 23, 2021*

# Sample Homework 4

(Hints for Homework 4)

1. (Tracing a satellite) A particle $A$ orbits around the origin clockwise along a circular path with radius $R$. While $A$ revolves around the origin, another particle $B$ orbits around $A$ counterclockwise, also along a circular path with radius $r < R$; think of $B$ as a satellite of $A$. Assume that both have the same period, that is, as $A$ revolves around the origin once, $B$ also revolves around $A$ once; for simplicity, you may set the period to be $2\pi$.

   (a) Analytically, determine the curve which is traced out by $B$ in one revolution.

   (b) Using the previous result, plot the trajectory of $C$ in one revolution for $r = 3$ and $R = 8$.

   (c) Play with the code by changing the period of the satellite, e.g., what if $B$ revolves around $A$ twice while $A$ revolves around the origin once?

2. (Spiral triangle) The following script[1] generates spirals using equilateral triangles as shown in the figure below.

```
m = 21; d_angle = 4.5; d_rot = 90;
th = linspace(0, 360, 4) + d_rot;
V = [cosd(th);
     sind(th)];
C = colormap(hsv(m));
s = sind(150 - abs(d_angle))/sind(30);
R = [cosd(d_angle) -sind(d_angle);
     sind(d_angle) cosd(d_angle)];
hold off
for i = 1:m
  if i > 1
      V = s*R*V;
  end
  plot(V(1,:), V(2,:), 'Color', C(i,:))
  hold on
end
set(gcf, 'Color', 'w')
axis equal, axis off
```

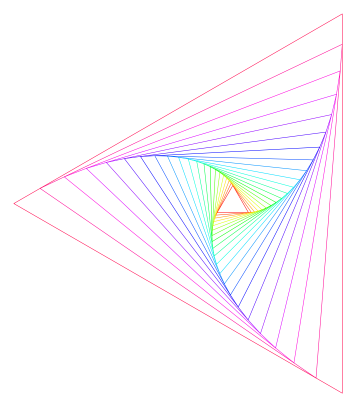


Figure 1: Spiral triangles with $m = 21$ and $\theta = 4.5°$.

   (a) Turn the script into a function named `spiralTriangle`. Your function must be written at the end of your homework live script (.mlx) file. Begin the function with the following header and comments.

[1] It is slightly modified from the code included in Lecture 9 slides. Note the introduction of a new variable `d_rot`, which is accountable for the rotation of the innermost triangle.

```
function V = spiralTriangle(m, d_angle, d_rot)
% SPIRALGON plots spiraling equilateral triangles
% input:   m = the number of triangles
%          d_angle = the degree angle between successive triangle
%          (can be positive or negative)
%          d_rot = the degree angle by which the innermost triangle
%                  is rotated
% output:  V = the vertices of the outermost triangle
....
```

(b) Run the statements below to generate some aesthetic shapes.

```
clf
subplot(2, 2, 1), spiralTriangle(21, 4.5, 0);
subplot(2, 2, 2), spiralTriangle(21, -4.5, 0);
subplot(2, 2, 3), spiralTriangle(41, 4.5, 0);
subplot(2, 2, 4), spiralTriangle(61, 3, 90);
```

**Note.** Copy the five lines, paste them inside a single code block, and run it. This code block must *precede* your function(s).

3. (Useful programs illustrating floating-point numbers) Download the following programs and play with them to understand how floating-point numbers work.

- `floatgui.m` (from NCM)
- `fp2ieee.m` and `ieee2fp.m` (from LM)

4. (Catastrophic cancellation; **LM** 9.3–11) The Taylor series expansion for the natural logarithm is

$$\log \xi = (\xi - 1) - \frac{1}{2}(\xi - 1)^2 + \frac{1}{3}(\xi - 1)^3 - \cdots,$$

which causes numerical problems when $\xi \approx 1$ since 1 must be subtracted from it before applying the Taylor series. So consider the function

$$f(x) = \begin{cases} \dfrac{\log(1+x)}{x} & \text{if } x > -1 \text{ and } x \neq 0 \\ 1 & \text{if } x = 0, \end{cases}$$

for $x \approx 0$. Numerically, 1 is added to $x$ before applying the logarithm and then the 1 is subtracted off before applying the Taylor series.

(a) Use the Taylor series expansion of $\log(1+x)$ to prove that $f$ is continuous at 0.

(b) Now approximate $f(x)$ numerically for $x = 10^{-k}$ for $k \in \mathbb{N}[1, 20]$ when the function is written using three slightly different expressions.

i. Calculate $f(x)$ as written.

ii. Calculate it as

$$f_1(x) = \frac{\log(1+x)}{(1+x) - 1}, \quad \text{for } x \neq 0.$$

(You and I know that analytically $f_1(x) \equiv f(x)$ for all $x$ – but MATLAB doesn't.)

iii. Define $f_2(x)$ which replaces $\log(1+x)$ by `log1p(x)`, so no adding or subtracting by 1 need to be done.

Then tabulate the results using `disp` or `fprintf`. The table should have four columns with the first being $x$, the second using $f(x)$, the third using $f_1(x)$, and the fourth using $f_2(x)$, with all shown to full accuracy. Do all these without using a loop.

(c) Comment on the results obtained in the previous part. Explain why certain methods work well while others do not.

5. (More on inversion of hyperbolic cosine) This is not a hint *per se*, but an explanation, for those curious, of why the proposed formula ($\star$) for $\operatorname{acosh}(x)$

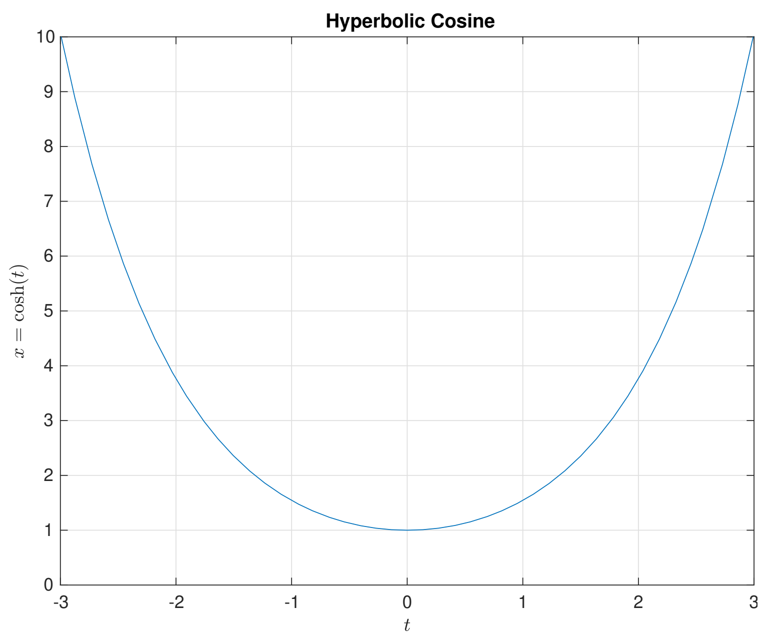
$$\log(x - \sqrt{x^2 - 1})$$

is different from what is presented in typical calculus textbooks or online sources[2].

One can readily confirm from the definition

$$\cosh(t) = \frac{e^t + e^{-t}}{2}$$

that the hyperbolic cosine function is an even function. Consequently, it is not *one-to-one* over its domain, $\mathbb{R}$, and so cannot be inverted entirely.




The usual workaround is to invert $\cosh(t)$ only on $[0, \infty)$ over which $\cosh(t)$ is one-to-one, and the resulting formula is conventionally regarded as *the* inverse hyperbolic cosine function. However, to handle cases where $t < 0$ such as ours (because we set `t=-4:-4:-16`), $\cosh(t)$ must be inverted over $(-\infty, 0]$ and the result is the formula given in the problem.

**Questions.**

[2]For example, see
https://en.wikipedia.org/wiki/Inverse_hyperbolic_functions#Inverse_hyperbolic_cosine

(a) Confirm that the inverse of the function $x = \cosh(t) = (e^t + e^{-t})/2$ on $(-\infty, 0]$ is

$$t = \log(x - \sqrt{x^2 - 1}), \quad x \in [1, \infty).$$

(b) Confirm that the previous result can be re-written as

$$t = -2\log\left(\sqrt{\frac{x+1}{2}} + \sqrt{\frac{x-1}{2}}\right).$$

(c) Would you need to worry about something like this in inverting hyperbolic sine function? Why or why not?