Homework 1 (Solution)

Math 3607

Tae Eun Kim

Table of Contents


```
format long g
```

Problem 1 (2.1--6)

(a) One can use the fact that $\pi\,\mathrm{rad} = 180^\circ$ for conversion. For example, define conversion factors

```
deg_to_rad = pi/180;
rad_to_deg = 180/pi;
```

and then use them as

```
27.9 * deg_to_rad
```

```
ans =
        0.486946861306418
```

```
13 * rad_to_deg
```

```
ans =
          744.84513367007
```

Tip. MATLAB provides `deg2rad` and `rad2deg` functions.

```
deg2rad(27.9)
```

```
ans =
        0.486946861306418
```

```
rad2deg(13)
```

```
ans =
          744.84513367007
```

(b) Using `sind` function,

```
sind(27)
```

```
ans =
```

```
        0.453990499739547
```

(c) Since the input is in radians, use `tan`:

```
tan(pi/5)
```

```
ans =
        0.726542528005361
```

(d)

```
1 / ( sind(20) + cosd(20) )
```

```
ans =
        0.780206008709879
```

(e) Note that the outputs of $\arcsin(1/3)$ and $\arctan(1/3)$ represent angles measured radians. If required, one can put them in degrees as shown above.

Radian outputs are computed by

```
asin(1/3)
```

```
ans =
        0.339836909454122
```

```
atan(1/3)
```

```
ans =
        0.321750554396642
```

while degree outs are obtained by `-d` variants of the corresponding functions

```
asind(1/3)
```

```
ans =
         19.4712206344907
```

```
atand(1/3)
```

```
ans =
          18.434948822922
```

One can quickly confirm that it is indeed correct by

```
asin(1/3) - deg2rad( asind(1/3) )
```

```
ans =
     0
```

```
atan(1/3) - deg2rad( atand(1/3) )
```

```
ans =
     0
```

(f) By the Pythagorean theorem, we know that the length of the base of a right triangle is given by

$$(\text{base}) = \sqrt{(\text{hyp})^2 - (\text{hgt})^2}$$

given the lengths of hypothenuse and height. In this problem, they are 145 and 144, respectively, and so the length of the base is

```
sqrt( 145^2 - 144^2 )
```

```
ans =
    17
```

Problem 2 (2.1--8)

(a) I will just type up my solutions instead of hand-writing.

(i) Taking the natural log of both sides and simplifying, we obtain $x = 3\ln(51)/\ln(5)$. Keep in mind that the natural logarithmic function is calculated by `log` in MATLAB. No function is named as `ln`.

```
x = 3 * log(51) / log(5)
```

```
x =
          7.32894186662572
```

(ii) Let $\theta = x + 2$ and $\theta_0 = \arcsin(-0.99) \in [-\pi/2, \pi/2]$.

```
theta0 = asin(-0.99)
```

```
theta0 =
         -1.42925685347047
```

It is clear that θ_0 is *a* solution of $\sin(\theta) = -0.99$. Using the symmetry of the sine curve about $\theta = \pi/2$, we deduce that $\theta_1 = \pi/2 + (\pi/2 - \theta_0) = \pi - \theta_0$ is another solution.

```
theta1 = pi - theta0
```

```
theta1 =
          4.57084950706026
```

θ_0 and θ_1 are all the roots within the $2\pi\text{-periodic}$ interval $[-\pi/2, 3\pi/2]$. Confirm:

```
sin(theta0)
```

```
ans =
                     -0.99
```

```
sin(theta1)
```

```
ans =
                       -0.99
```

Finally, using the 2π-periodicity of the sine function, we conclude that all solutions are written as

$$\theta = \theta_0 + 2\pi m \text{ or } \theta = \theta_1 + 2\pi n \text{ where } m, n \text{ are integers;}$$

or

$$\theta = \theta_0 + 2m\pi \text{ or } \theta = -\theta_0 + (2m+1)\pi \text{ where } m, n \text{ are integers.}$$

(iii) The (real) root of the given cubic equation is given by $x_0 = \sqrt[3]{\pi^2 - 5}$.

```
x0 = nthroot(pi^2 - 5, 3)  % or  x = (pi^2 - 5)^(1/3)
```

```
x0 =
          1.69497993102987
```

Advanced Note. The equation has two additional roots $z = x_0 e^{\pm 2\pi i/3}$ which are complex.

```
% complex roots
z1 = x0 * exp(2i*pi/3)
```

```
z1 =
         -0.847489965514937 +      1.46789567917667i
```

```
z2 = x0 * exp(-2i*pi/3)
```

```
z2 =
         -0.847489965514937 -      1.46789567917667i
```

```
% confirm that they solve the equation
z1^3 + 5 - pi^2
```

```
ans =
                            0 -    3.5527136788005e-15i
```

```
z2^3 + 5 - pi^2
```

```
ans =
                            0 +    3.5527136788005e-15i
```

We were asked to solve only for real x, so the calculation of `x0` is sufficient, but I still decided to write this down for any interested or advanced audience.

(iv) High school algebra exercise yields $x = (1/\sin(20^\circ) - 1)^2$:

```
( 1/sind(20) - 1 )^2
```

```
ans =
```

```
        3.70102337008686
```

(vi) Let $t = x^2$. Then the given equation become $t^2 + t - 5 = 0$ whose roots are $t = (-1 \pm \sqrt{21})/2$. Since $t = x^2$, we obtain that

$$x = \pm\sqrt{(-1+\sqrt{21})/2} \text{ (real) and } x = \pm i\sqrt{(1+\sqrt{21})/2} \text{ (purely imaginary).}$$

Since we are only interested in real roots, we take the first two.

```
t0 = (-1 + sqrt(21))/2;
x1 = sqrt(t0)
```

```
x1 =
        1.33839002068826
```

```
x2 = -x1
```

```
x2 =
       -1.33839002068826
```

```
%% complex roots
% t1 = (-1 - sqrt(21))/2;
% x3 = sqrt(t1)
% x4 = -x3
```

Problem 3 (2.1--14)

The pendulum of a clock is supposed to make a single swing in exactly 1 second, *i.e.*, its period T is 1 second.

```
T = 1;
```

In a day, the pendulum makes 86,400 swings, because there are that many seconds in 24 hours:

$$24\,\text{hr} \times \frac{60\,\text{min}}{1\,\text{hr}} \times \frac{60\,\text{sec}}{1\,\text{min}} = 86,400\,\text{sec}.$$

So a clock running 5 minutes fast per day swings 86,400 times in 24 hr - 5 min = 86,100 seconds, which in turn implies that its period T' is smaller than 1

$$T' = \frac{86,100}{86,400}.$$

```
Tprime = 86100/86400;
```

Denoting the length of the correct pendulum by L and the incorrect one by L', we can express their periods using Equation (15.25) as

$$T = 2\pi\sqrt{\frac{L}{g}}\left(1 + \frac{1}{16}\theta_0^2 + \cdots\right) \text{ and } T' = 2\pi\sqrt{\frac{L'}{g}}\left(1 + \frac{1}{16}\theta_0^2 + \cdots\right).$$

Note that the maximum angle of oscillation is θ_0 for both cases. The problem requires that only the length of the pendulum can be changed.

Taking the ratio of the two equations and cancelling terms, we find that

$$\frac{T}{T'} = \sqrt{\frac{L}{L'}}, \text{ which implies that } L = L'\left(\frac{T}{T'}\right)^2.$$

That is, the length of the pendulum arm needs to by lengthened by $(T/T')^2$:

```
(T/Tprime)^2
```

```
ans =
         1.00698078160473
```

Problem 4. (Temperature Conversion)

Let F and C denote temperatures in (degree) Fahrenheit and Celsius, respectively. It is known that

$$C = \frac{5}{9}(F - 32).$$

(The factor of $5/9 = 100/180$ is due to the fact that the temperature between the freezing point and the boiling point of water is divded into 180 degrees in Farrenheit and 100 degrees in Celsius; the subtraction of 32 is to match the representations for the freezing point -- $32°F = 0°C$. The two are linearly related.)

I will present the script as a non-executable code block.

```
F = input('Degrees in Fahrenheit: ');
C = (5/9)*(F-32);
fprintf('Fahrenheit: %6.2f\n', F)
fprintf('Celsius:    %6.2f\n', C)
```

Problem 5. (Oblate Spheroid)

I will present the requested script as a single code block, so that the document is self-contained. Instead of using `input` function, I will directly provide the geo-data of the Earth.

```
% r1 = input('Enter equatorial radius (r1): ');
% r2 = input('Enter polar radius (r2 < r1): ');
r1 = 6378.137;
r2 = 6356.752;
gamma = acos(r2/r1);
A_exact = 2*pi*( r1^2 + ...
```

```
    r2^2/sin(gamma) * log( cos(gamma)/(1-sin(gamma)) ));
A_approx = 4*pi*( (r1+r2)/2 )^2;
disp('The surface area of the given spheroid:')
```

```
The surface area of the given spheroid:
```

```
disp(['Exact:  ', num2str(A_exact)])
```

```
Exact:  510065604.9442
```

```
disp(['Approx: ', num2str(A_approx)])
```

```
Approx: 509495321.6397
```

Remarks.

1. Note that the line where `A_exact` is calculated is continued to the next line by using `...`. This is useful when you want to type in a lengthy code over multiple lines for enhanced readability.
2. When `disp` or `fprintf` is called within a Live Script, it prints out outputs and disrupts the code block. I personally find this annoying and would prefer all outputs to be printed after the end of the gray box. A workaround, for now, is to write an external script and call it in here as I demonstrated in class. Soon, we will learn about MATLAB *functions* and they can help us fix this nuiance to some degree.