Homework 02 Hints

Problem 2

Make sure you begin by remembering/writing down the conversion formulas:

$$\begin{aligned}\rho &= \text{some function of } x, y, z\\ \theta &= \text{another function of } x, y, z\\ \phi &= \text{yet another function of } x, y, z\end{aligned}$$

A key point of this problem is how you manage to calculate θ in $[0, 2\pi)$. If you simply use either `theta=atan(y/x)` or `theta=atan2(y,x)`, your script will fail to convert properly points such as

$$(x, y, z) = (1, -1, 1) \quad \longrightarrow \quad (\rho, \phi, \theta) = (1.7320\ldots, 0.9553\ldots, 5.4977\ldots).$$

since `atan` outputs angles in $[-\pi/2, \pi/2]$ while `atan2` produces angles in $[-\pi, \pi]$. For more hints/references:

- Exercise 3 of Lecture 3
- In the Command Window, type `doc atan` and `doc atan2`.

Problem 3

- (d) To iterate over all prime numbers not exceeding n, you may use

  ```
  for j = primes(n)
  ...
  end
  ```

  For example,

  ```
  for j = primes(20)
  ...
  end
  ```

  runs the statement inside the loop body repeated as `j` takes on values $2, 3, 5, 7, 11, 13, 17, 19$.

- (f) Skip this
- (j) Recall that we can design an infinite loop by

```
while true
...
end
```

We can install a *brake* inside the loop using an `if`-statement and the `break` command. For example,

```
% break out when a randomly generate integer n is 50.
step = 0;
while true
    step = step + 1;
    n = randi(100);   % random integer btw 1 and 100
    if n == 50
        break
    end
end
```

When MATLAB see `break`, it immediately comes out of the loop and carries out subsequent statements, if any. There is another command that is useful in controlling the behavior of loops.

- `break`: break out of a loop
- `continue`: pass control to the next iteration

Problem 5

The following series is famously known to be convergent.

$$\sum_{k=1}^{\infty} \frac{1}{k^2} = \frac{\pi^2}{6}.$$

In other words, the sequence of partial sums $\{s_n\}$ is convergent with $\lim_{n\to\infty} s_n = \pi^2/6$ where

$$s_n = \sum_{k=1}^{n} \frac{1}{k^2}.$$

(The setting looks very similar to the problem now.)
Let's see how many terms are needed so that the partial sum is close enough to the true value. More concretely, let's find the smallest integer n so that $e_n := |s_n - \pi^2/6| \leq 10^{-2}$.

- To calculate a partial sum s_n, use a for-loop:

```
% assume n is given
s_n = 0;
for k = 1:n
    s_n = s_n + 1/k^2;
end
```

This will be a central building block in our program.

- To design a loop which continues to repeat while $e_n > 10^{-2}$:

```
s = pi^2/6;                              % exact sum
errBound = 1e-2;                         % error bound
err_n = 1;                               % so that loop below can be initiated
n = 1;
while err_n > errBound                   % the loop will close when err_n <= errBound
    s_n = 0;
    for k=1:n
        s_n = s_n + 1/k^2;
    end
    err_n = abs(s_n - s);
    n = n + 1;
end
```

The two snippets above should you a good idea of what to do.