MatLab Project 1 - For-loop and Approximation of $\pi$ (Due February 6)

MatLab (Matrix Laboratory) is an interactive software system for numerical computations and graphics. As the name suggests, MatLab is especially designed for matrix computations: solving systems of linear equations, computing eigenvalues and eigenvectors, factoring matrices, and so forth. In addition, it has a variety of graphical capabilities, and can be extended through programs written in its own programming language. Actually, MatLab has been called the 4th generation programming language.

MatLab is installed on PCs in Lab 220, Thompson Hall. To start the software, click the following sequence of icons: Start, Programs, TH 220 Programs, MatLab 6.5.1, MatLab icon. Three windows are displayed by MatLab. For this project, you mainly use the “Command Window” on the right. So, you may close the other two windows. Now let us learn some simple MatLab commands.

1. To Quit:
To quit from MatLab, type “quit” at the MatLab prompt, i.e.

```
>> quit (enter)
```

2. Using MatLab as a calculator
MatLab can be used as an expression evaluator. To do this you simply type a mathematical expression after the MatLab prompt >>. For example, we want to evaluate $\pi$:

```
>> pi (enter)
```

Now You see: ans = 3.1416. If you need to read more decimal digits, type format long and then press the return key. Now type pi again.

```
>> format long
```

```
>> pi
```

```
>> 3.14159265358979
```

Type format short to return back to 5 decimal digits. pi is a built-in MatLab variable. Try the following examples:

```
>> sin(pi/4)
```

```
>> 2^(log2(4))
```

```
>> sqrt(9)
```

3. Assigning Values to Variables
MatLab allows you create variables. For example,

```
>> x = 5;
```

```
>> y = pi/4;
```

```
>> z = y + sqrt(x)
```

```
z =3.02146614089724
```

From this point on, $x = 5$, $y = \frac{\pi}{4}$ and $z = \frac{\pi}{4} + \sqrt{5}$.

Use the who command to list the currently active variables.

```
>> who
```

```
Your variables are: ans x y z
```

4. Built-in Functions
MatLab has many built-in functions. For example, (try them to see what you get)

```
>> sin(pi/2)
```

```
>> cos(pi/3)
```

```
>> tan(pi/4)
```

```
>> exp(0.2)
```

```
>> atan(1)
```

```
>> log(exp(1))
```

Guess and try a function yourself. How does it work? After you use MatLab for a while, you can guess
many commands.

5. Operations:

Operations: +, -, *, /, ^. Try the following.

```matlab
gt2^(1/2)-sqrt(2)
gt3*exp(-1/2)*sin(pi/3)+2*log(3)
```

6. Programs: - for-loop

Suppose we need to compute the sum

\[ s = \sum_{i=1}^{n} \frac{(-1)^i}{i+1} \]

for a given positive integer \( n \).

```matlab
gt>>n=1000; s=0; for i=1:n, s=s+(-1)^i/(i+1); end
gt>>s
```

If we need to compute the sum with a different \( n \) now, we can use the up arrow key to bring up the previous command and then change the value \( n \).

**Your assignment (1):** Try the same computation with \( n = 10000 \), \( n = 100000 \), and \( n = 1000000 \). What are the values of \( s \)? Use **format long** and copy down **10 digits**.

Now let us use **MatLab** to approximate \( \pi \). It is known that the Taylor series for \( \tan^{-1}(x) \) is

\[
\arctan(x) = \tan^{-1}(x) = \sum_{i=1}^{\infty} (-1)^{i-1} \frac{1}{2i-1} x^{2i-1}, \quad -1 < x \leq 1.
\]

Since \( \arctan(1) = \frac{\pi}{4} \),

\[
\pi = 4 \arctan(1) = 4 \left( \sum_{i=1}^{\infty} (-1)^{i-1} \frac{1}{2i-1} (1)^{2i-1} \right) = 4 \left( \sum_{i=1}^{\infty} (-1)^{i-1} \frac{1}{2i-1} \right).
\]

Hence, \( \pi \) can be approximated by

\[
4 \left( \sum_{i=1}^{n} (-1)^{i-1} \frac{1}{2i-1} \right)
\]

for some large positive integer \( n \). Let \( n = 100 \).

```matlab
gt>>s=0; for i=1:100, s=s+(-1)^(i-1)/(2*i-1); end, s=4*s
    s = 3.13159290355855
```

Now we compute the relative error:

```matlab
gt>>abs(s-pi)/pi
    ans = 0.00318301929431
```

**Your assignment (2):** Approximate \( \pi \) with \( n = 1000 \), \( n = 10000 \) and \( n = 100000 \). Compute each relative error.