

MATH 3043, Numerical Analysis I
Fall 2020

Lab 0

This lab will get you started working in MATLAB.

This lab will not be graded but may be submitted for feedback to Problems 2-5. To receive feedback for your solutions on these problems, submit a single script file `Lab0Lastname.m` and the corresponding published file `Lab0Lastname.pdf` (for example, my submitted files would be `Lab0Zumbrum.m` and `Lab0Zumbrum.pdf`) on Canvas by **September 6 at 11:59 PM**. Each solution should

- be contained in a separate cell which includes the problem number,
 - run independent of other cells,
 - be adequately commented.
1. Complete Sections 1-10, 12, & 13 of the MATLAB Onramp course.
 2. Use `if`, `elseif`, and `else` statements to assign a letter grade for a particular percentage according to the grading scale for our course.
 3. Use a `for` loop to print a table of the first ten (non-zero, positive) integers and their square roots.
 4. Use a `for` loop to calculate the n^{th} partial sum of the series

$$\sum_{k=1}^{\infty} \left(\frac{2}{3}\right)^k$$

for $n = 5, 10, 25, 50, 100$.

5. Write a script that uses a `while` loop to repeatedly divide a number by 2 and outputs the result until the result is less than 1.

fprintf Syntax

- `fprintf` – displays text and formatted variable output in a single line
 - `fprintf('text %FW.PC text', variable)` – displays specified string including value stored in variable
 - `%` – marks place where value of `variable` is entered in text
 - `F` – flag: `-` (minus sign) left justifies number within field, `+` (plus sign) prints a sign character in front of the number, `0` (flag) adds zeros if the number is shorter than the field
 - `W` – field width: the number of total characters used in printing
 - `P` – field precision: the number of digits to the right of the decimal
 - `C` – conversion character: `d` decimal number, `e` exponential notation, `f` floating-point notation, `g` short e notation, `c` character, `s` string
 - `"` – adds a single quote in a string
 - escape characters – `'\n'` new line, `'\t'` horizontal tab
- Examples:
 - flag: `-` (minus sign)

```
>> % min. field width 10, text not left-justified
>> fprintf('%10s\n','abcdef')
      abcdef
>> % min. field width 10, text left-justified
>> fprintf('%-10s\n','abcdef')
abcdef
```
 - flag: `0`

```
>> % min. field width 5, fill in space before the input with zeros
>> fprintf('%05f\n',3)
00003
```
 - flag: `+` (plus sign)

```
>> fprintf('%f\n',100) % unsigned version of 100
100.000000
>> fprintf('%+f\n',100) % signed version of 100
+100.000000
```
 - Field Width

```
>> % min field width 10, 5 decimal places (notice unused spaces on the left)
>> fprintf('%10.5f\n',pi)
      3.14159

>> % no field width specified, 5 decimal places (notice no unused spaces)
>> fprintf('%5f\n',pi)
3.14159
```
 - Escape Characters: making a simple table

```
>> fprintf('%5s\t\t%5s\n%5d\t\t%5d\n', 'label 1', 'label 2', 1234,7)
label 1 label 2
1234      7
```

Conditional Statement Syntax

- **if** statement – performs an action if a condition is true

```
if condition
    action
end
```

- **if-else** statement – performs one action if a condition is true, another if the condition is false

```
if condition
    action1
else
    action2
end
```

- **elseif** clause – used with an **if** statement to choose between more than two actions

```
if condition1
    action1
elseif condition2
    action2
else
    action3
end
```

Loop Statement Syntax

- **for** – loop for which the number of iterations is known before the loop is executed

```
for loopvariable = range
    action
end
```

- **while** – loop for which the number of iterations is unknown before the loop is executed; iterations occur until a specified condition is satisfied; variables in the **condition** must have a value to test for the first iteration and at least one variable in the **condition** must be updated in the **action**

```
while condition
    action
end
```