Homework Submission Instructions

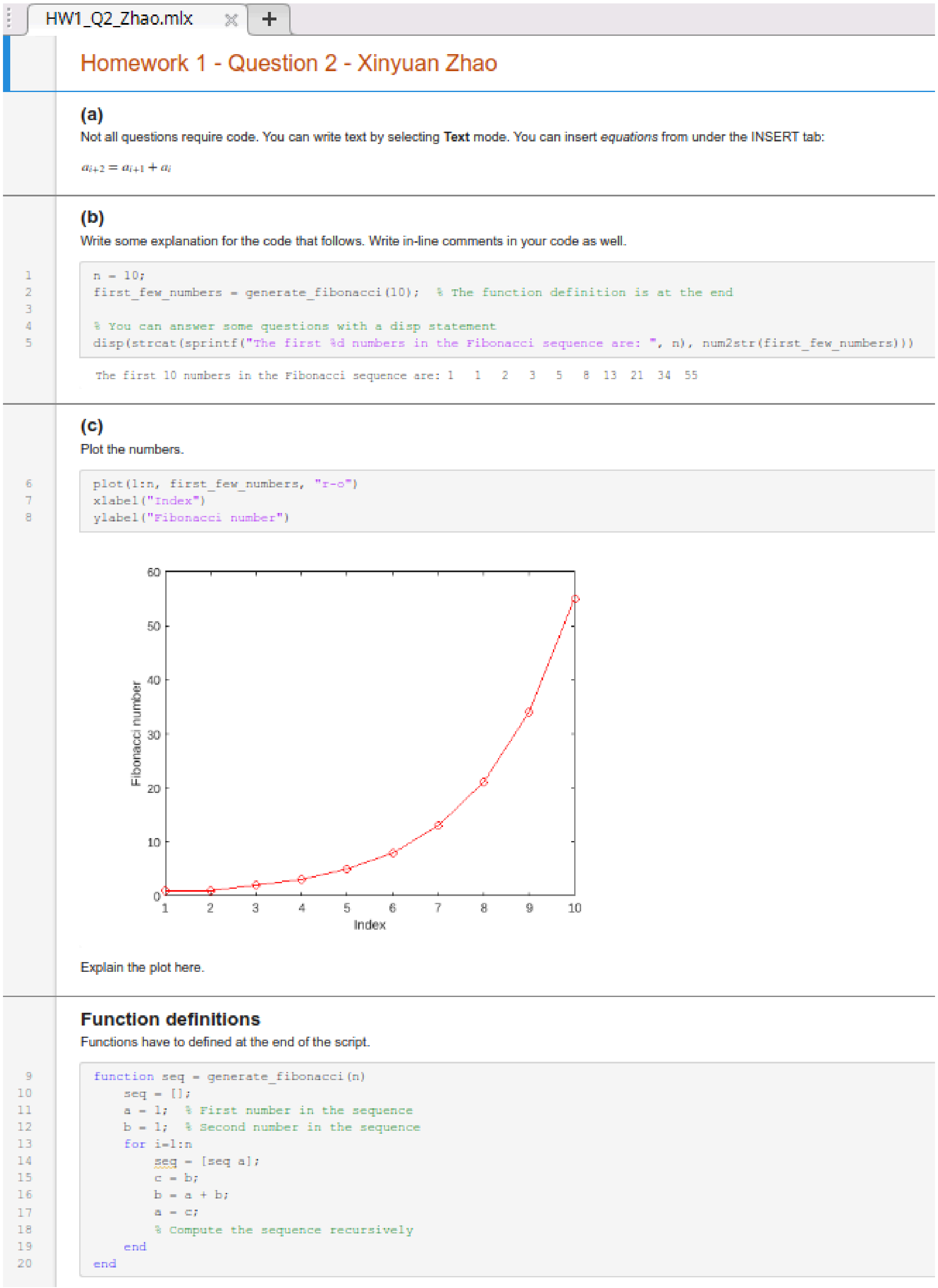
You can complete the homeworks with either MATLAB or Python. Use the interactive script format: live script (.mlx) for MATLAB and Jupyter notebook (.ipynb) for Python. You may create a separate interactive script for each question in the problem set, or have a single interactive script for all questions. If you are using a single interactive script, **make sure to clear all workspace variables in between questions** using *clear* in MATLAB or *%reset* in Python. Include all text, equations, code (including functions), output and figures in the interactive script file. Clearly label the start of each problem and sub-problem using text or markup. Do **not** use plain scripts (.m for MATLAB and .py for Python).

You will submit homeworks on **Gradescope** (linked through Brightspace). There will be an assignment for each homework, named “Homework #” (# denotes the homework number). Here you will submit the exported pdf from your script (export instructions for MATLAB and Python are below).
If you created separate scripts for each question, convert each interactive script file to a PDF, and join them as one PDF file for submission. If you don’t have a good PDF editor on your computer, you can join them with *free* browser tools like this one:
https://www.adobe.com/acrobat/online/merge-pdf.html
During the submission process, you should assign PDF pages to their corresponding questions. The website also gives you an option of submitting images: do not use it.

Handwritten answers: For questions that do not require coding, you may alternatively submit a handwritten answer. Make sure your handwriting is legible. Save your handwritten answer as a PDF and include them in the one joined PDF file that you submit to “Homework #”.

MATLAB live script

In a MATLAB live script, you can define functions at the end of the script. Do not use separate plain script files (.m) to define functions. To save the live script as a PDF file, click the “Export” button under the “LIVE EDITOR” tab and select “.pdf” format.



Homework 1 - Question 2 - Xinyuan Zhao

(a)

Not all questions require code. You can write text by selecting **Text** mode. You can insert *equations* from under the INSERT tab:

$a_{i+2} = a_{i+1} + a_i$

(b)

Write some explanation for the code that follows. Write in-line comments in your code as well.

```
n = 10;
first_few_numbers = generate_fibonacci(10);  % The function definition is at the end

% You can answer some questions with a disp statement
disp(strcat(sprintf("The first %d numbers in the Fibonacci sequence are: ", n), num2str(first_few_numbers)))
```

```
The first 10 numbers in the Fibonacci sequence are: 1   1   2   3   5   8  13  21  34  55
```

(c)

Plot the numbers.

```
plot(1:n, first_few_numbers, "r-o")
xlabel("Index")
ylabel("Fibonacci number")
```

Explain the plot here.

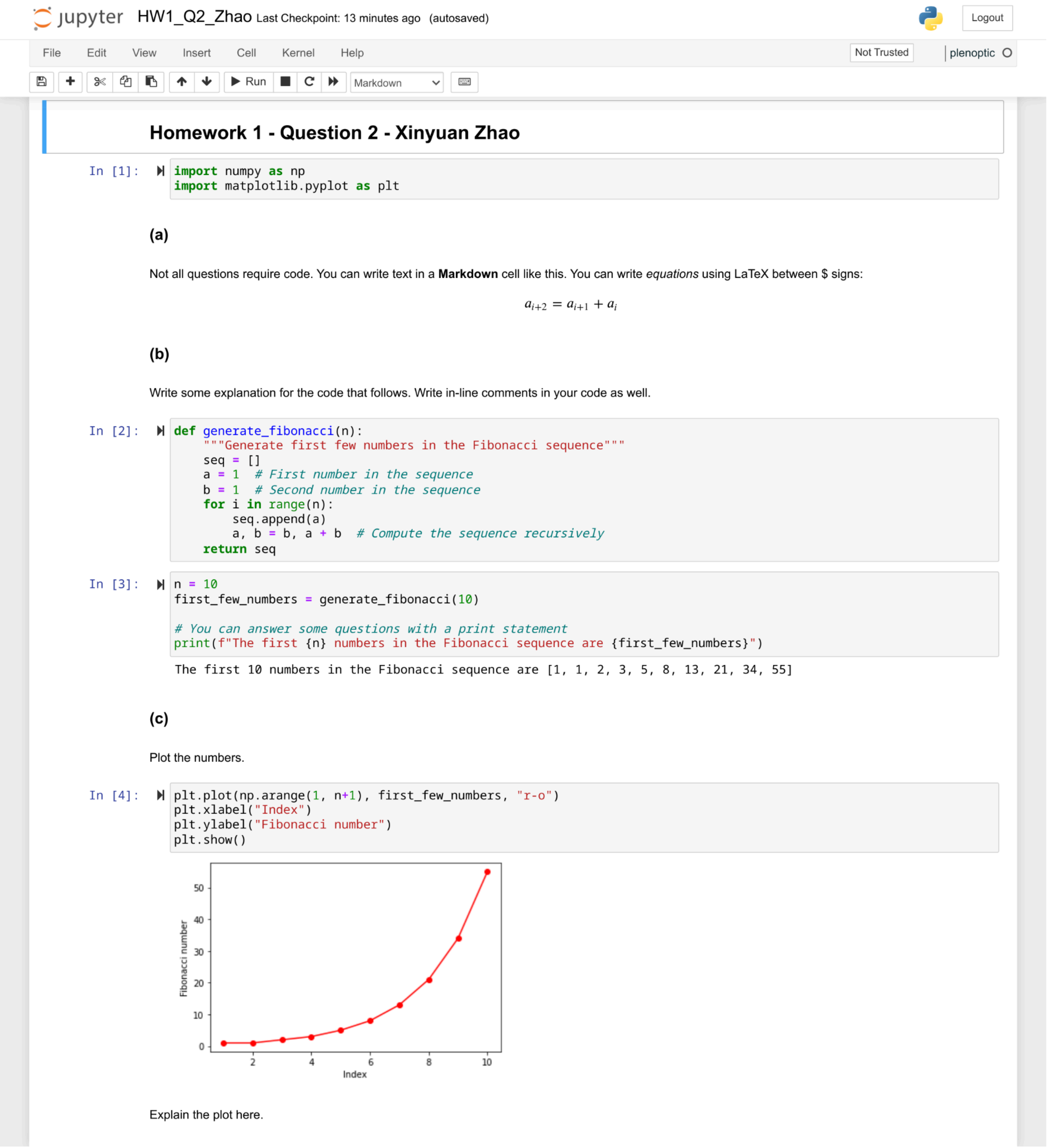
Function definitions

Functions have to defined at the end of the script.

```
function seq = generate_fibonacci(n)
    seq = [];
    a = 1;  % First number in the sequence
    b = 1;  % Second number in the sequence
    for i=1:n
        seq = [seq a];
        c = b;
        b = a + b;
        a = c;
        % Compute the sequence recursively
    end
end
```

Python Jupyter notebook

To save the notebook as a PDF file, the best way is to go to File -> Print Preview, and print the webpage to PDF using your browser (the .pdf export in Jupyter requires a copy of LaTeX which is a huge download so avoid this).



Homework 1 - Question 2 - Xinyuan Zhao

In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
```

(a)

Not all questions require code. You can write text in a **Markdown** cell like this. You can write *equations* using LaTeX between $ signs:

$$a_{i+2} = a_{i+1} + a_i$$

(b)

Write some explanation for the code that follows. Write in-line comments in your code as well.

In [2]:

```
def generate_fibonacci(n):
    """Generate first few numbers in the Fibonacci sequence"""
    seq = []
    a = 1  # First number in the sequence
    b = 1  # Second number in the sequence
    for i in range(n):
        seq.append(a)
        a, b = b, a + b  # Compute the sequence recursively
    return seq
```

In [3]:

```
n = 10
first_few_numbers = generate_fibonacci(10)

# You can answer some questions with a print statement
print(f"The first {n} numbers in the Fibonacci sequence are {first_few_numbers}")
```

```
The first 10 numbers in the Fibonacci sequence are [1, 1, 2, 3, 5, 8, 13, 21, 34, 55]
```

(c)

Plot the numbers.

In [4]:

```
plt.plot(np.arange(1, n+1), first_few_numbers, "r-o")
plt.xlabel("Index")
plt.ylabel("Fibonacci number")
plt.show()
```

Explain the plot here.