Chapter 1

Assignments with solutions:
Number systems

1.2 Exercises NS-1

Topic of this homework:

Introduction to MATLAB/OCTAVE (see the Matlab or Octave tutorial for help).

Deliverable: Report with charts and answers to questions. Hint: Use \LaTeX.\footnote{http://www.overleaf.com}

Plotting complex quantities in Octave/Matlab

Problem # 1: Consider the functions \( f(s) = s^2 + 6s + 25 \) and \( g(s) = s^2 + 6s + 5 \).

\[\text{sol:} \text{The roots of } f(s) \text{ are } -3 \pm 4i \text{ (in Matlab: roots([1 6 25])). The roots of } g(s) \text{ are } -1 \text{ and } -5 \text{ (in Matlab: roots([1 6 5])). You will find the program that generates all these figures at } \text{http://jontalle.web.engr.illinois.edu/uploads/298.17/NS1.m}\]

\(\text{Q 1.1}\)

Find the zeros of functions \( f(s) \) and \( g(s) \) using the command \texttt{roots()}. \textbf{Sol:} The roots of \( f(s) \) are \(-3 \pm 4i \) (in Matlab: \texttt{roots([1 6 25]))}. The roots of \( g(s) \) are \(-1 \) and \(-5 \) (in Matlab: \texttt{roots([1 6 5]))}. You will find the program that generates all these figures at \texttt{http://jontalle.web.engr.illinois.edu/uploads/298.17/NS1.m}

\(\text{Q 1.2}: \text{Show the roots of } f(s) \text{ as red circles and of } g(s) \text{ as blue plus signs.}\)

The x-axis should display the real part of each root, and the y-axis should display the imaginary part. Use \texttt{hold on} and \texttt{grid on} when plotting the roots. \textbf{Sol:}
CHAPTER 1. ASSIGNMENTS WITH SOLUTIONS: NUMBER SYSTEMS

–Q 1.3 Give your figure the title ‘Complex Roots of f(s) and g(s)’. Label the x- and y-axis ‘Real Part’ and ‘Imaginary Part.’

Hint: use xlabel and ylabel. Type ylim([-10 10]) and xlim([-10 10]), to expand the axes.

Problem #2: Consider the function \( h(t) = e^{j2\pi ft} \) for \( f = 5 \) and \( t = [0:0.01:2] \).

–Q 2.1: Use subplot to show the real and imaginary parts of \( h(t) \).

Make two graphs in one figure. Label the x-axes ‘Time (s)’ and the y-axes ‘Real Part’ and ‘Imaginary Part’.

Sol: Breaking \( h(t) \) into real and imaginary parts gives \( e^{j2\pi 5t} = \cos(10\pi t) + j\sin(10\pi t) \).

–Q 2.2: Use subplot to plot the magnitude and phase parts of \( h(t) \).

Use the command angle or unwrap(angle()) to plot the phase. Label the x-axes ‘Time (s)’ and the y-axes ‘Magnitude’ and ‘Phase (radians)’. Sol:
Prime numbers, infinity, etc. in Octave/Matlab

**Problem # 3: Prime numbers, infinity, etc.**

---

**Q 3.1:** Use the Matlab function `factor` to find the prime factors of 123, 248, 1767, and 999,999.

**Sol:** Factors: 123 (3, 41), 248 (2,2,2,31), 1767 (3,19,31), 999999 (3,3,3,7,11,13,37)

---

**Q 3.2:** Use the Matlab function `isprime` to check if 2, 3 and 4 are prime numbers.

What does the function `isprime` return when a number is prime, or not prime? Why?

**Sol:** Function `isprime(2)` returns 1, `isprime(3)` returns 1, and `isprime(4)` returns 0. 1 means ‘yes’ and 0 means ‘no’

---

**Q 3.3:** Use the Matlab/Octave function `primes.m` to generate prime numbers between 1 and $10^6$.

Save them in a vector $x$. Plot this result using the command `hist(x)`. **Sol:**

![Histogram of prime numbers](image)

---

**Q 3.4:** Now try `[n,bincenters] = hist(x)`.

Use `length(n)` to find the number of bins. **Sol:** `length(n)` is 10

---

**Q 3.5:** Set the number of bins to 100 by using an extra input argument to the function `hist`.

Show the resulting figure and give it a title and axes labels. **Sol:**

![Histogram with 100 bins](image)

---

**Problem # 4: Inf, NaN and logarithms in Octave/Matlab**
– Q 4.1: Try 1/0 and 0/0 in the Octave/Matlab command window.
What are the results? What do these ‘numbers’ mean in Matlab? Sol: 1/0 returns Inf (infinity) and 0/0 returns NaN (‘not a number’).

– Q 4.2: Try log(0) in the command window.
In Matlab, the natural logarithm ln(·) is computed using the function log (log10, and log2 are computed using log10 and log2). Sol: log(0) is -Inf

– Q 4.3: Try log(-1) in the command window. Do you get what you expect for ln(-1)? Show how Matlab arrives at the answer by considering -1 = e\(^{i\pi}\).
Sol: log(-1) is 0 + i\(\pi\), because ln(-1) = ln(e\(^{i\pi}\)) = \(i\pi\)ln(e) = \(i\pi\)

– Q 4.4: Try log(exp(j*sqrt(pi))) (i.e., log e\(^{\sqrt{\pi}\}i\)) in the command window. What do you expect? Show how Octave/Matlab arrives at the answer by considering -1 = e\(^{i\pi}\).
Sol: log e\(^{\sqrt{\pi}\}i\) = \(\sqrt{\pi}\) because ln(·) is the inverse of e(·).

– Q 4.5: What does inverse mean in this context? What is the inverse of ln(f(x))?
Sol: The decibel is very important in engineering (and unused in mathematics). It is defined as the log of a power ratio. If a power ratio is 2, the dB value is 6 [dB]. A ratio of 10 is 20 [dB]. Thus the formula for the dB-ref is 10 log\(_{10}\) P/P ref. Thus the decibel is defined on the log (i.e., ratio) scale. Engineers quickly learn to “think” in dB units, because its so easy (once they learn to think in terms of ratios).

While the definition is in terms of power, it practice, is almost always used in terms of voltage, pressure, current, velocity, (force and flux), etc. Of course power is the product of force and flux, and the log of the power is the sum of the log of the force plus the log of the flux.

Problem # 5: Very large primes on Intel computers

– Q 5.1: Find the largest prime number that can be stored on an Intel 64 bit computer, which we call \(\pi_{\text{max}}\).
Hint: As explained in the Matlab/Octave command help flintmax, the largest positive integer is \(2^{53}\), however the largest integer that can be factored is \(2^{32} = \sqrt{2^{64}}\). Explain the logic of your answer.
Hint: help isprime(). Sol: Using Matlab/Octave, start with the largest integer \(2^{32}\) and check if its prime. Then work down by subtracting 1, and again check. Stop when you get to the first prime below the largest integer. The answer I get is: \(2^{32} - 5 = 4,294,967,291\), is the first prime below \(2^{32}\) prime.

Problem # 6: Suppose you are interested in primes that are greater than \(\pi_{\text{max}}\). How can you find them on an Intel computer (i.e., one using IEEE-floating point)?

– Q 6.1: Thus consider a sieve containing only odd numbers, starting from 3 (not 2).
Hint 1: Since every prime number greater than 2 is odd, there is no reason to check the even numbers. \(n_{\text{odd}} \in \mathbb{N}/2\) contain all the primes other than 2. Sol: At this time, I don’t see any way to do this, due to the Matlab limitation that it cannot factor numbers larger than \(2^{32}\).

Problem # 7: The following identity is interesting:
\[ 1 = 1^2 \]
\[ 1 + 3 = 2^2 \]
\[ 1 + 3 + 5 = 3^2 \]
\[ 1 + 3 + 5 + 7 = 4^2 \]
\[ 1 + 3 + 5 + 7 + 9 = 5^2 \]
\[
\ldots
\]
\[ \sum_{n=0}^{N-1} 2n + 1 = N^2. \]

–Q 7.1: Can you find a proof?²

Sol: Subtracting any line from the line following it, gives:

\[
(1 - 1) + 3 = 2^2 - 1^2
\]
\[ 5 = 3^2 - 2^2 \]
\[ 7 = 4^2 - 3^2 \]
\[ 9 = 5^2 - 4^2 \]
\[
\ldots
\]
\[ \sum_{n=0}^{N-1} 2n + 1 - \sum_{n=0}^{N-2} 2n + 1 = N^2 - (N - 1)^2 \]
\[ 2N - 1 = N^2 - (N^2 - 2N + 1) \]
\[ 2N - 1 = 2N - 1. \]

Thus the two sides are equal, as suggested by the above formula.

Can you find a simpler more constructive “proof?” Hint: assuming you know what integration by parts is, can you devise a concept called Summation by parts?

²This problem came from an exam problem for Math 213, Fall 2016.