Chapter 1

Number systems

1.1 Problems NS-1

Topic of this homework:
Introduction to Matlab/Octave (see the Matlab or Octave tutorial for help)
Deliverables: Report with charts and answers to questions.

Plotting complex quantities in Octave/Matlab

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

- 1.1: Find the zeros of functions \( f(s) \) and \( g(s) \) using the command \texttt{roots()}.

Ans:

- 1.2: Show the roots of \( f(s) \) as red circles and of \( g(s) \) 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.

Ans:

- 1.3 Give your figure the title “Complex Roots of \( f(s) \) and \( g(s) \).” Label the \( x \)- and \( y \)-axes “Real Part” and “Imaginary Part.” Hint: Use \texttt{xlabel}, \texttt{ylabel}, \texttt{ylim([-10 10])}, and \texttt{xlim([-10 10])} to expand the axes.
Problem #2: Consider the function $h(t) = e^{2\pi ft}$ for $f = 5$ and $t=[0:0.01:2]$.

- 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.”

  Ans:

- 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).”

  Ans:

Prime numbers, infinity, etc. in Octave/Matlab

Problem #3: Prime numbers, infinity, etc.

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

  Ans:

- 3.2: Use the Matlab/Octave 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?

  Ans:

- 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)`.

  Ans:
– 3.4: Now try \( [n, \text{bincenters}] = \text{hist}(x) \).
Use \text{length}(n) to find the number of bins. \textbf{Ans:}

– 3.5: Set the number of bins to 100 by using an extra input argument to the function \text{hist}.
Show the resulting figure and give it a title and axes labels. \textbf{Ans:}

\textbf{Problem #4: Inf, NaN and logarithms in Octave/Matlab}

– 4.1: Try \(1/0\) and \(0/0\) in the Octave/Matlab command window.
What are the results? What do these ‘numbers’ mean in Octave/Matlab? \textbf{Ans:}

– 4.2: Try \(\log(0)\), \(\log_{10}(0)\) and \(\log_{2}(0)\) in the command window.
In Matlab/Octave, the natural logarithm \(\ln(\cdot)\) is computed using the function \(\log\). Functions \(\log_{10}\) and \(\log_{2}\) are computed using \(\log_{10}\) and \(\log_{2}\). \textbf{Ans:}

– 4.3: Try \(\log(1)\) in the command window. What you expect for \(\log_{10}(1)\) and \(\log_{2}(1)\)?
\textbf{Ans:}

– 4.4: Try \(\log(-1)\) in the command window. What do you expect for \(\log_{10}(-1)\) and \(\log_{2}(-1)\)?
\textbf{Ans:}
– 4.5: Show how Matlab/Octave arrives at the above answer because \(-1 = e^{i\pi}\).  
\textbf{Ans:} 

– 4.6: Try \(\log(\exp(j\sqrt{\pi}))\) (i.e., \(\log e^{j\sqrt{\pi}}\)) in the command window. What do you expect?  
\textbf{Ans:} 

– 4.7: What does inverse mean in this context? What is the inverse of \(\ln f(x)\)?  
\textbf{Ans:} 

– 4.8: What is a decibel? (Look up decibels on the internet.)  
\textbf{Ans:} 

\textbf{Problem # 5: Very large primes on Intel computers} 

– 5.1: Find the largest prime number that can be stored on an Intel 64 bit computer, which we call \(\pi_{\text{max}}\).  
\textbf{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.  
\textbf{Ans:}
1.1. PROBLEMS NS-1

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)?

– 6.1: Extending the number of primes you may considered.
   Hint 1: Use `uint64(myprimes)` to extend the numbers unsigned 64 bit integers (we don’t need negative primes). Hint 2: Since every prime number greater than 2 is odd, there is no reason to check the even numbers. Starting from 3 (not 2). \( n_{\text{odd}} \in \mathbb{N}/2 \) contain all the primes other than 2. **Ans:**

Problem #7: The following identity is interesting:

\[
\begin{align*}
1 &= 1^2 \\
1 + 3 &= 2^2 \\
1 + 3 + 5 &= 3^2 \\
1 + 3 + 5 + 7 &= 4^2 \\
1 + 3 + 5 + 7 + 9 &= 5^2 \\
&\quad \\vdots \\
\sum_{n=0}^{N-1} 2n + 1 &= N^2.
\end{align*}
\]

– 7.1: Can you find a proof?\(^1\)

**Ans:**

\(^1\)This problem came from an exam problem for Math 213, Fall 2016.