1.2 Problems NS-2

Topic of this homework:

Prime numbers, greatest common divisors, the continued fraction algorithm

Continued fractions

Problem # *1: Here we explore the continued fraction algorithm (CFA)*
In its simplest form, the CFA starts with a real number, which we denote as $\alpha \in \mathbb{R}$. Let us work with an irrational real number, $\pi \in \mathbb{I}$, as an example because its CFA representation will be infinitely long. We can represent the CFA coefficients α as a vector of integers n_k, $k = 1, 2, \ldots, \infty$:

$$\begin{aligned} \alpha &= [n_1; n_2, n_3, n_4, \ldots] \\ &= n_1 + \frac{1}{n_2 + \frac{1}{n_3 + \frac{1}{n_4 + \cdots}}}. \end{aligned}$$

The CFA is recursive, with three repeated steps per iteration. For example $\alpha_1 = \pi \approx 3.14159\ldots$, $n_1 = 3$, $r_1 = \pi - 3$, and $\alpha_2 \equiv 1/r_1$.

$$\begin{aligned} \alpha_2 &= 1/0.1416 = 7.0625\ldots \\ \alpha_1 = n_1 + \frac{1}{\alpha_2} &= n_1 + \frac{1}{n_2 + \frac{1}{\alpha_3}} = \cdots . \end{aligned}$$

In terms of a Matlab/Octave script,

```
alpha0 = pi;
K=10;
n=zeros(1,K); alpha=zeros(1,K);
alpha(1)=alpha0;

for k=2:K  %k=1 to K
n(k)=round(alpha(k-1));
%n(k)=fix(alpha(k-1));
alpha(k)= 1/(alpha(k-1)-n(k));
%disp([fix(k), round(n(k)), alpha(k)]); pause(1)
end
disp([n; alpha]);
%Now compare this to matlab's rat() function
rat(alpha0,1e-20)
```

– 1.1: By hand find the first few values of n_k for $\alpha = e^\pi \approx 23.1407$.
Sol: The CFA for this is: $e^\pi = 23.1407\cdots = [23; 7, 9, 4, \cdots]$. ■

– 1.2: For the proceeding question, what is the error (remainder) when you truncate the continued fraction after $n_1, \ldots, n_3$? Give the absolute value of the error and the percentage error relative to the original α.
Sol: The remainder is $e^\pi - (23 + 1/(7 + (1/9))))$ which gives an error of $\epsilon = |e^\pi - (23 + 1/(7 + (1/9))))|/e^\pi = 2.92 \cdot 10^{-6} = 0.0003\%$ ■

– 1.3: Use the Matlab/Octave program provided to find the first 10 values of n_k for $\alpha = e^\pi$, and verify your result using the Matlab/Octave command `rat()`.
Sol: $e^\pi = 23.1407\cdots = [23; 7, 9, 4, -2, -591, -2, -10, 3, -2, \cdots]$. ■

– 1.4: Discuss the similarities and differences between the Euclidean algorithm and the CFA.
Sol:

1. Both are recursive, meaning that the steps are repeated one after another.
2. The EA starts from two numbers (a,b). The output of the gca(a,b) is the GCD. The CFA starts with a single number and the output is a sequence of integers. If the sequence terminates the number was rational. If the sequence does not terminate, the number is irrational.
3. The EA works with the difference between the minimum and maximum of the two numbers whereas the CFA works with the rounding function and the reciprocal of the error.
4. It would seem that the goals of the two algorithms, the starting point, and the results are totally different. Both are very useful and powerful. Both generalize to more difficult situations than working with simple numbers.

■

Greatest common divisors

Consider using the *Euclidean algorithm* to find the *greatest common divisor* (i.e., GCD; the largest common prime factor) of two numbers (Allen 2020, p. 42). This algorithm may be performed using one of two methods:

Method	Division	Subtraction
On each iteration...	$a_{i+1} = b_i$ $b_{i+1} = a_i - b_i \cdot \text{floor}(a_i/b_i)$	$a_{i+1} = \max(a_i, b_i) - \min(a_i, b_i)$ $b_{i+1} = \min(a_i, b_i)$
Start with $i = 1$ and terminate when:	$b = 0$ (GCD $= a$)	$b = 0$ (GCD $= a$)

The division method (Matlab's floor function) (Eq. 2.1, Sec. 2.1.2, Ch. 2) is preferred because the subtraction method may require a huge number of itterations steps.

Problem # *2: Understanding the Euclidean algorithm (GCD)*

– 2.1: Find the prime factors of $a = 85$ and $b = 15$.
Sol: From Octave's `factor()` we find $85 = 17 \cdot 5$, $15 = 3 \cdot 5$. ■

– 2.2: What is the greatest common prime factor of $a = 85$ and $b = 15$?
Sol: The largest common factor `gcd(85,15)` is 5. ■

– 2.3: By hand, perform the Euclidean algorithm for $a = 85$ and $b = 15$.
Sol: Division method:

$$a_1 = 15 \qquad b_1 = 85 - 15\lfloor \cancel{\tfrac{85}{15}}^{75} \rfloor = 10$$

$$a_2 = 10 \qquad b_2 = 15 - 10\lfloor \cancel{\tfrac{15}{10}}^{2} \rfloor = 5$$

$$a_3 = 5 \qquad b_3 = 10 - 5\lfloor \cancel{\tfrac{10}{5}}^{2} \rfloor = 0$$

$\therefore$ gcd$= 5$
Subtraction method:

$$\begin{aligned} a_1 &= 85 - 15 = 70 & b_1 &= 15 \\ a_2 &= 70 - 15 = 55 & b_2 &= 15 \\ a_3 &= 55 - 15 = 40 & b_3 &= 15 \\ a_4 &= 40 - 15 = 25 & b_4 &= 15 \\ a_5 &= 25 - 15 = 10 & b_5 &= 15 \\ &\text{swap} & & \\ a_6 &= 15 - 10 = 5 & b_6 &= 10 \\ a_7 &= 10 - 5 = 5 & b_7 &= 5 \\ &\text{terminate} & & \end{aligned}$$

$\therefore$ gcd$= 5$ ■

– 2.4: By hand, perform the Euclidean algorithm for $a = 75$ and $b = 25$. Is the result a prime number?
Sol: Division method:

$$a_1 = 25 \qquad b_1 = 75 - 25\lfloor \cancel{\tfrac{75}{25}}^{3} \rfloor = 0$$

Subtraction method:

$$\begin{aligned} a_1 &= 75 - 25 = 50 & b_1 &= 25 \\ a_2 &= 50 - 25 = 25 & b_2 &= 25 \end{aligned}$$

$\therefore$ gcd$= 25$
The result is $25 = 5^2$, the *square* of a prime number. ■

– 2.5: Consider the first step of the GCD division algorithm when $a < b$ (e.g., $a = 25$ and $b = 75$). What happens to a and b in the first step? Does it matter if you begin the algorithm with $a < b$ rather than $b < a$?

Sol: If $a < b$, the first step of the division algorithm swaps the terms ($a \to b$ and $b \to a$). ■

– 2.6: Describe in your own words how the GCD algorithm works. Try the algorithm using numbers that have already been divided into factors (e.g., $a = 5 \cdot 3$ and $b = 7 \cdot 3$).

Sol: Division method:

$$a_1 = 5 \cdot 3 \qquad b_1 = 7 \cdot 3 - 5 \cdot 3 \lfloor \frac{7 \cdot 3}{5 \cdot 3} \rfloor^{1} = 2 \cdot 3$$

$$a_2 = 2 \cdot 3 \qquad b_2 = 5 \cdot 3 - 2 \cdot 3 \lfloor \frac{5 \cdot 3}{2 \cdot 3} \rfloor^{2} = 1 \cdot 3$$

$$a_3 = 1 \cdot 3 \qquad b_3 = 2 \cdot 3 - 1 \cdot 3 \lfloor \frac{2 \cdot 3}{1 \cdot 3} \rfloor^{2} = 0$$

Subtraction method:

$$\begin{aligned} a_1 &= 7 \cdot 3 - 5 \cdot 3 = 2 \cdot 3 \\ a_2 &= 5 \cdot 3 - 2 \cdot 3 = 3 \cdot 3 \\ a_3 &= 3 \cdot 3 - 2 \cdot 3 = 1 \cdot 3 \\ a_4 &= 2 \cdot 3 - 1 \cdot 3 = 1 \cdot 3 \end{aligned} \qquad \begin{aligned} b_1 &= 5 \cdot 3 \\ b_2 &= 2 \cdot 3 \\ b_3 &= 2 \cdot 3 \\ b_4 &= 1 \cdot 3 \end{aligned}$$

The algorithm iteratively converges on the GCD by subtracting out multiples of the GCD until only the GCD is left. ■

– 2.7: Find the GCD of $2 \cdot \pi_{25}$ and $3 \cdot \pi_{25}$.

Sol: π_{25} ■ .

Problem # *3: Coprimes*

– 3.1: Define the term coprime.

Sol: when two integers have no common factors the are said to be *coprime* ■

– 3.2: How can the Euclidean algorithm be used to identify coprimes?

Sol: If `gcd(a,b)=1` they only have 1 as a common factor, thus they are coprime. ■

– 3.3: Give an important application of the Euclidean algorithm.

Sol: Given two integers $n, d \in \mathbb{Z}$, if we wish to reduce the fraction n/d, we must cancel the common factors. Example: If $n = 9, d = 6$ then $9/6 = (3 \cdot 3)/(2 \cdot 3) = 3/2$, where the GCD, 3, may be identified using the Euclidean algorithm. While this fraction may be easily simplified via inspection, the GCD algorithm could be very helpful for larger numbers n, d. ■

– 3.4: Write a Matlab function, `function x = my_gcd(a,b)`*, that uses the Euclidean algorithm to find the GCD of any two inputs* `a` *and* `b`*. Test your function on the (*`a`*,* `b`*) combinations from the previous problem. Include a printout (or hand-write) your algorithm to turn in.*

Hints and advice:

- Don't give your variables the same names as Matlab functions! Since `gcd` is an existing Matlab/Octave function, if you use it as a variable or function name, you won't be able to use `gcd` to check your `gcd()` function. Try `clear all` to recover from this problem.
- Try using a "while" loop for this exercise (see Matlab documentation for help).
- You may need to use some temporary variables for `a` and `b` in order to perform the algorithm.

Sol: Division method:

```
function x = my_gcd(a,b)
while b>0
atmp= a; btmp = b;
a = btmp; b = atmp-btmp*floor(atmp/btmp);
end
```

Subtraction method:

```
function x = my_gcd(a,b)
while a~=b
atmp= a; btmp = b;
a = max(atmp,btmp) - min(atmp,btmp); b = min(atmp,btmp);
end
```

■

Algebraic generalization of the GCD (Euclidean) algorithm

Problem # 4: *In this problem we are looking for integer solutions $(m,n) \in \mathbb{Z}$ to the equations $ma+nb = gcd(a,b)$ and $ma+nb = 0$ given positive integers $(a,b) \in \mathbb{Z}^+$.*
Note that this requires that either m or n be negative. These solutions may be found using the Euclidean algorithm only if (a,b) are coprime ($a \perp b$). Note that integer (whole number) polynomial relations such as these are known as *Diophantine equations*. Such equations (e.g., $ma+nb=0$) are linear Diophantine equations, possibly the simplest form of such relations.

Example: `gcd(2,3)= 1:` For $(a,b)=(2,3)$, the result is

$$\begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}0 & 1\\1 & -2\end{bmatrix}\begin{bmatrix}0 & 1\\1 & -1\end{bmatrix}\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}\begin{bmatrix}2\\3\end{bmatrix} = \underbrace{\begin{bmatrix}-1 & 1\\3 & -2\end{bmatrix}}_{m\quad n}\begin{bmatrix}2\\3\end{bmatrix}.$$

Thus from the above equation we find the solution (m,n) to the integer equation

$$2m+3n = \texttt{gcd}(2,3) = 1;$$

namely, $(m,n)=(-1,1)$ (i.e., $-2+3=1$). There is also a second solution $(3,-2)$ (i.e., $3\cdot 2-2\cdot 3=0$) that represents the terminating condition. Thus these two solutions are a pair and the solution exists only if (a,b) are coprime ($a \perp b$).
Subtraction method: This method is more complicated than the division algorithm because at each stage we must check whether $a<b$. Define

$$\begin{bmatrix}a_0\\b_0\end{bmatrix} = \begin{bmatrix}a\\b\end{bmatrix}, \qquad Q = \begin{bmatrix}1 & -1\\0 & 1\end{bmatrix}, \qquad S = \begin{bmatrix}0 & 1\\1 & 0\end{bmatrix},$$

where Q sets $a_{i+1} = a_i - b_i$ and $b_{i+1} = b_i$ assuming $a_i > b_i$, and S is a swap matrix that swaps a_i and b_i if $a_i < b_i$. Using these matrices, we implement the algorithm by assigning

$$\begin{bmatrix}a_{i+1}\\b_{i+1}\end{bmatrix} = Q\begin{bmatrix}a_i\\b_i\end{bmatrix} \text{ for } a_i > b_i, \qquad \begin{bmatrix}a_{i+1}\\b_{i+1}\end{bmatrix} = QS\begin{bmatrix}a_i\\b_i\end{bmatrix} \text{ for } a_i < b_i.$$

The result of this method is a cascade of Q and S matrices. For $(a,b)=(2,3)$, the result is

$$\begin{bmatrix}1\\1\end{bmatrix} = \underbrace{\begin{bmatrix}1 & -1\\0 & 1\end{bmatrix}}_{Q}\underbrace{\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}}_{S}\underbrace{\begin{bmatrix}1 & -1\\0 & 1\end{bmatrix}}_{Q}\underbrace{\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}}_{S}\begin{bmatrix}2\\3\end{bmatrix} = \underbrace{\begin{bmatrix}2 & -1\\-1 & 1\end{bmatrix}}_{m\quad n}\begin{bmatrix}2\\3\end{bmatrix}.$$

Thus we find two solutions (m,n) to the integer equation $2m+3n = \text{gcd}(2,3) = 1$.

– 4.1: By inspection, find at least one integer pair (m,n) that satisfies $12m+15n=3$.
Sol: By inspection, $(m,n)=(-1,1)$ is one solution. ■

– 4.2: Using matrix methods for the Euclidean algorithm, find integer pairs (m,n) that satisfy $12m+15n=3$ and $12m+15n=0$. Show your work!!!
Sol: Division method:

$$\begin{bmatrix}3\\0\end{bmatrix} = \begin{bmatrix}-1 & 1\\5 & -4\end{bmatrix}\begin{bmatrix}12\\15\end{bmatrix}$$

Subtraction method:

$$\begin{bmatrix}3\\3\end{bmatrix} = \begin{bmatrix}-1 & 1\\4 & -3\end{bmatrix}\begin{bmatrix}12\\15\end{bmatrix}$$

■

– 4.3: Does the equation $12m+15n=1$ have integer solutions for n and m? Why or why not?
Sol: No, because $\text{gcd}(12,15) = gcd(3*4, 3*5) = 3$, not 1. Thus there are no Diophantine solutions to this equation. ■

Problem # *5: Matrix approach:*
It can be difficult to keep track of the a's and b's when the algorithm has many steps. We need an alternative way to run the Euclidean algorithm using matrix algebra. Matrix methods provide a more transparent approach to the operations on (a, b). Thus the Euclidean algorithm can be classified in terms of standard matrix operations. Write out the indirect matrix approach discussed at the end of Sec. 2.4.3 (Eq. 2.4.3).
Sol: Division method:
Define

$$\begin{bmatrix} a \\ b \end{bmatrix}_0 = \begin{bmatrix} a_0 \\ b_0 \end{bmatrix}, \qquad \begin{bmatrix} a \\ b \end{bmatrix}_{i+1} = \begin{bmatrix} 0 & 1 \\ 1 & -\lfloor a/b \rfloor \end{bmatrix}_i \begin{bmatrix} a \\ b \end{bmatrix}_i$$

■

Prime numbers

Problem # *6: Every integer may be written as a product of primes.*

– 6.1: Write the numbers $1,000,000$, $1,000,004$, *and* $999,999$ *in the form* $N = \prod_k \pi_k^{\beta_k}$. Hint: Use Matlab/Octave to find the prime factors.
Sol: $1,000,000 = 2^6 \cdot 5^6$
$1,000,004 = 2^2 \cdot 53^2 \cdot 89$
$999,999 = 3^3 \cdot 7 \cdot 11 \cdot 13 \cdot 37$
■

– 6.2: Give a generalized formula for the natural logarithm of a number $\ln(N)$ *in terms of its primes* π_k *and their multiplicities* β_k. *Express your answer as a* sum *of terms.*
Sol: $\ln N = \sum_k \beta_k \ln(\pi_k)$ ■

Problem # *7: Using the computer*

– 7.1: Explain why the following brief Matlab/Octave program returns the prime numbers π_k *between 1 and 100.*

```
n=2:100;
k = isprime(n);
n(k)
```

Sol: The first line $n = 2 : 100$ defines the row vector $n = [2, 3, 4, \cdots, 100]$. The second line creates a row vector the same length as n, with entries of 1 if the element is prime and zero if the element is not prime. The third line $n(k)$ prints out $n()$ if $k = 1$, namely it is a list of all the primes from 2 to 100. Run this program without the ';' at the end of each line, and to see what it is doing. ■

– 7.2: How many primes are there between 2 *and* $N = 100$*?*
Sol: `length(n(k))` returns 25. Thus there are 25 primes less than 100 ($N/4$, on average). ■

Problem # *8: Prime numbers may be identified using a sieve.*

– 8.1: By hand, complete the sieve of Eratosthenes for $n = 1, \ldots, 49$. *Start by writing out a table of the integers 1-50, as 5 rows of 10 numbers. Starting with the first prime,* $p_k = 2$, $k = 1$, circle it *and* cross out *all multiples (e.g.,* $2\pi_k = r, 3\pi_k = 6, \cdots, 24 * \pi_2 = 48$.*). Then repeat for the second, third, and higher primes* π_2. *When done, only the circled primes should remain. Be sure you look up the definition of a prime.*