Topics in Elementary Number Theory

Introduction

Number theory is the study of the set of natural numbers $\mathbb{N} = \{1, 2, 3, ...\}$. Some typical examples of number theoretic questions are:

- Can the sum of two squares be a square?
- Is the equation $x^n + y^n = z^n$ solvable for $n > 2$?
- Are there infinitely many primes?
- Is a given integer prime or composite?
- How to efficiently factorize a composite number?

The first two questions are about existence of positive integer solutions of the equations $x^n + y^n = z^n, n \geq 2$. You certainly know that for $n = 2$ the equation is solvable. Solutions to this equation are called *Pythagorean Triples*, *PTs*. A PT with *mutually prime* x and y (that is, x, y do not have common divisors other than 1) is called *Primitive Pythagorean Triple*, *PPT*. You will see two versions of a characterization of PPTs and use one of them in Lab 1 to construct and visualize a large number of PPTs. *Fermat's Last Theorem* states that there is no three positive integers a, b, and c satisfying the equation $a^n + b^n = c^n$ with $n > 2$. Fermat did not provide the proof, and the statement remained just a conjecture for more than three centuries. In 1994 a British mathematician Andrew Wiles presented the first successful proof − after 358 years of effort by mathematicians! The equation $x^n + y^n = z^n$ is a particular example of so-called *Diophantine equations*. Diophantine equation in two variables has the form $P(x, y) = 0$, where $P(x, y)$ is a polynomial with integer coefficients; solutions are sought in the set $\mathbb{Z}$ of integers. A universal algorithm for deciding if a particular Diophantine equation is solvable was posed by D. Hilbert as the 10^{th} of his celebrated 23 mathematical problems for 20th century. In 1970, a novel result in mathematical logic, known as Matiyasevich's theorem, settled the problem negatively: such a general algorithm does not exist. In this module you will work only with linear Diophantine equations (LDEs) in two variables. These equations are always solvable under a simple assumption. A method for solving LDE in two unknowns is described in this lecture. This simple method proved to be useful in an applied industrial problem, and in this lecture you will solve this problem in Lab 2 with CAS assistance.

The third question was answered positively by Euclid. Euclid's proof of the theorem on infinitude of primes is simple and beautiful. Take a look at https://en.wikipedia.org/wiki/Euclid%27s_theorem and enjoy.

The last two questions are about prime characterization and prime factorization. The questions have been stated by mathematicians, in particular C. F. Gauss, as the central problems in the number theory. At the present, they are not just academic questions. Primality and prime factorization became very important in applications, mostly in cryptography. An efficient algorithm for testing primality called AKS (using the first letters of the three co-authors) was found in 2002. In this lecture, we will introduce only two classic "toy" factorization algorithms to give you a flavor of the factoring problem. For more, see http://www.cs.columbia.edu/~rjaiswal/factoring-survey.pdf. The first three pages in this article are on the history of the factoring problem. See also https://en.wikipedia.org/wiki/RSA_Factoring_Challenge about challenge put forward by RSA Laboratories to encourage research into computational number

theory. The RSA Laboratories offered cash prizes for successful factorization of certain composite numbers. Many algorithms have been devised to make factorization an ever faster problem. It is a major open problem in the number theory to decide how fast integer factorization algorithms can be.

There are other open problems in the number theory stated as conjectures and verified for a huge number of cases using modern computers but not proven rigorously. Here is a classic example of such *conjecture*.

Goldbach's Conjecture. Every even number $n \geq 4$ is the sum of two primes.

For example, $6 = 3 + 3, 10 = 3 + 7, 100 = 3 + 97$. The conjecture has been tested up to the number $4 \cdot 10^{14}$ (2017) but remains unproven.

Exercise 1. Make a CAS function $my\_goldbach(n)$ that takes an even number $n > 2$ and prints representation(s) of the number as the sum of two primes.

The ternary Goldbach conjecture, or the three-primes problem, asserts that every odd integer n greater than five is the sum of three primes. In 1937, Russian mathematician I.M. Vinogradov proved that the conjecture is true for all n above a large constant C. The bound was improved several times, and in 2013 the conjecture was proven for $n \geq C = 10^{30}$ (!) by a Peruvian mathematician H.A. Helfgott, see https://arxiv.org/abs/1312.7748. For $n < 10^{30}$, it was verified by "brute force" computer calculations that the statement is true. Problem solved.

Number theory was considered to be one of the "purest" branches of mathematics, but when it comes to computer security, it has turned out to become one of the most useful. For example, algorithms based on number theory help to protect your credit card number when you shop online or, generally, protect sensitive information transmitted through communication channels. Moreover, according to Donald Knuth, who has been called the "father of the analysis of algorithms", in computer science, "...virtually every theorem in elementary number theory arises in a natural, motivated way in connection with the problem of making computers do high-speed numerical calculations". For more about number theory see https://en.wikipedia.org/wiki/Number_theory.

All coefficients and variables in this lecture are assumed to be integers or rational numbers unless stated otherwise.

Some algorithms of the elementary number theory

Primes, primality tests, and prime factorization

A *prime number* is a positive integer with no positive integer divisors other than one and itself. A set of natural numbers is *coprime* if the only common divider of the members of the set is 1. The *Fundamental Theorem of Arithmetic*, also called the unique factorization theorem, states that every integer greater than one either is a prime number itself or can be represented as the product of prime numbers. This representation is unique up to the order of factors. The theorem is a typical existence statement; it does not provide a way to constructively factor composite numbers or determine if a number is prime or composite.

The *Sieve of Eratosthenes* algorithm finds all primes less than some number n. This ancient algorithm can be used as a primality test for integers that are not too large. Here are the steps

of the algorithm.

Step 1. Make the list of all natural numbers from ***2*** to ***n***.

Step 2. Set $p = 2$.

Step 3. Cross out all numbers divisible by ***p*** except for ***p*** itself.

Step 4. Find the first number l on the list (not crossed) that is greater than p. If there is no such number, stop. Otherwise, set $p = l$, which is the next prime, and repeat from Step 3.

Once you crossed off all multiples of all primes less or equal $floor(\sqrt{n})$, than anything not crossed out must be prime.

Clearly, if n is composite, its smallest divider is less than or equal to $\sqrt{n}$. **(Why?)** Therefore, if on a step of crossing multiples of some number p, the input number n gets crossed, then n is not a prime, and p is its smallest divisor. The algorithm is slow and not efficient for large n.

Example 1. Make a list of all primes less than 50 and implement the Eratosthenes algorithm using CAS interactively.

Solution

The algorithm was implemented step by step using CAS assistance. We consecutively eliminated multiples of primes up to and including the number $floor(\sqrt{50}) = 7$. Specific commands for these steps are different for different CASs. In writing a code for a particular CAS, we have chosen to cross all numbers multiple to primes, including the primes themselves, and collected the excluded primes on a separate list. At the end, we merged the list of excluded primes and the primes left on the original list.

Here are the outputs after excluding multiples of 2, 3, 5, and 7:

[3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49]

[5, 7, 11, 13, 17, 19, 23, 25, 29, 31, 35, 37, 41, 43, 47, 49]

[7, 11, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 49]

[11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]

The last list merged with primes **[2, 3, 5, 7]** gives the **answer**:

[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47].

Exercise 2.

(a) Make a CAS function $find\_primes(n)$ that takes a natural number *n* and returns the list of all primes less than or equal to *n*. Test your function on the problem in Example 1.

(b) Make a CAS function $list\_of\_primes(k, n)$ that makes a list of n prime numbers greater than a positive integer k. Make a list of 30 prime numbers greater than 100. You may use appropriate CAS command for finding a prime next to a specific integer.

Two simple primality tests

The simplest primality test is the *trial division* also called the *brute force* method. In this method, given an input number n, all integers less than or equal to $\sqrt{n}$ are tested using the trial division to see if they actually divide the given number. The algorithm is slow and not practical for large n.

Another simple primality test, *Fermat's factorization method*, is based on the following fact:

Any odd integer N can be represented as the difference of squares of integers.

Proof

Let $N = c \cdot d$ be an odd number. (If N is prime, either d or c equals one.) You can write

$$N = \frac{(c+d)^2 - (c-d)^2}{4} = \left(\frac{c+d}{2}\right)^2 - \left(\frac{c-d}{2}\right)^2.$$

Set $a = \frac{c+d}{2}, b = \frac{c-d}{2}$. Since both c and d are odd (**why?**), $c + d$ and $c - d$ are even. Therefore a and b are integers. □

The proof implies that $N = (a+b) \cdot (a-b)$. The basic Fermat's algorithm consecutively tries integers $a \geq \mathbf{ceil}(\sqrt{N})$ until the difference $a^2 - N$ becomes the perfect square. Then b is set to $\sqrt{a^2 - N}$ and the factors c, d are defined as $c = a + b, d = a - b$.

To speed up the basic method, various improvements of the Fermat algorithm have been developed, see https://en.wikipedia.org/wiki/Fermat%27s_factorization_method.

Example 2. Factor $N = 6767$ using the Fermat factorization algorithm.

Solution

Step 1. $a = \mathbf{ceil}(\sqrt{6767}) = 83;\ b = \sqrt{83^2 - 6767} \approx 11.18;$

Step 2. $a = 84; b = \sqrt{84^2 - 6767} = 17.$

Answer: Factors of $N = 6767$ are: $a + b = 101,\ a - b = 67$. Check: $101 \cdot 67 = 6767$.

Exercise 3. Make a CAS function $my\_fermat(n)$ that takes an odd number n, implements the Fermat's of factorization algorithm, and returns two factors of n. Run your code on example of your choice and check the result with CAS factoring command.

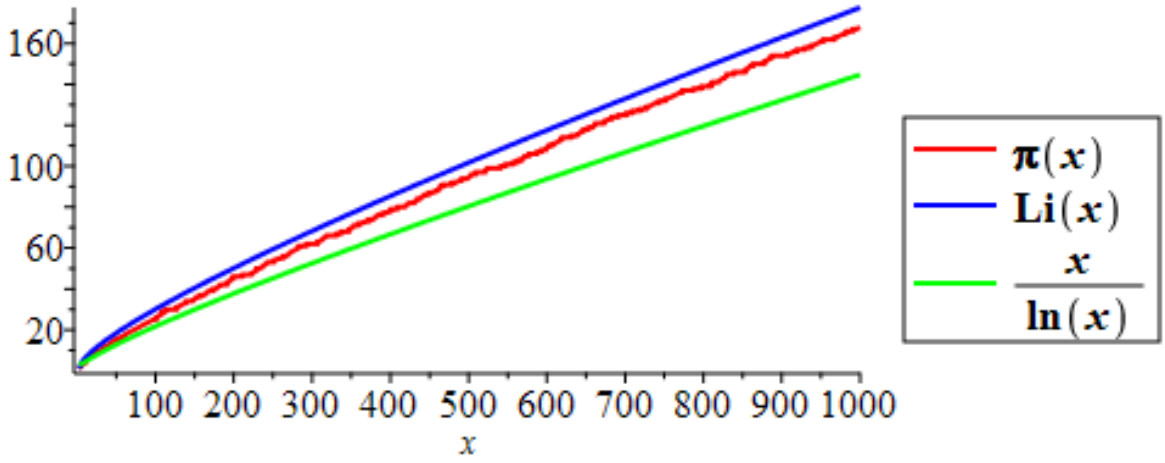
Number of primes and the Riemann Hypothesis.

The *prime counting function*, $\pi(x)$, counting number of primes less than or equal to a given positive number *x*, can be estimated *asymptotically* by the function $\frac{x}{\ln(x)}$, that is, the limit of the ratio of these two functions is one, $\lim_{x \to \infty} \frac{\ln(x) \cdot \pi(x)}{x} = 1.$

This is the statement of the *Prime Number Theorem*. It is known that the famous *Riemann hypothesis* is equivalent to the conjecture that the function $\mathbf{Li}(x) = \int_2^x \frac{1}{\ln(t)} dt$ is also an *asymptotic estimate* of $\pi(x)$, that is, $\lim_{x \to \infty} \frac{\pi(x)}{\mathbf{Li}(x)} = 1$.

Exercise 4. Most CASs include commands for evaluation of the prime counting function and

the function $\mathbf{Li}(x)$. (For the latter, you can also use appropriate CAS integration command.) Make a function $counting\_primes(N)$. The function should return a figure with plots of $\pi(x)$, $\mathbf{Li}(x)$, and $\frac{x}{\ln(x)}$ in the range $[3, N]$. Choose a large integer N and use your function to make a figure similar to Fig. 2.1.



The prime counting function and its two asymptotic approximations.

Euclid's algorithm revisited

The algorithm returns the greatest common divisor $gcd(m, n)$ of the integers m and n. Without loss of generality, assume that $m, n > 0$ and $m > n$.

Example 2. Find $gcd(203, 91)$.

Manual solution

Using the division algorithm, you obtain the equation

$$203 = 91 \cdot 2 + 21.$$

Clearly, any common divisor of *203* and *91* is also a common divisor of 91 and 21, i.e.,

$$gcd(203, 91) = gcd(91, 21).$$

Applying the division algorithm to this new pair of numbers, you obtain

$$91 = 21 \cdot 4 + 7.$$

Finally,

$$21 = 7 \cdot 3 + 0.$$

Answer: $gcd(203, 91) = 7$.

Symbolic description of the Euclid's algorithm

Step 1: $m = n \cdot q_0 + r_0, 0 \le r_0 < n$.

Step 2: $r_0 = r_1 \cdot q_2 + r_2, 0 \le r_2 < r_1$.

...

Step k : $r_{k-2} = r_{k-1} \cdot q_k + r_k, 0 \le r_k < r_{k-1}$. ·(*)

...

Step xx : $r_{l-2} = r_{l-1} \cdot q_l + 0 \Rightarrow r_l = gcd(m, n)$.

Remark for coding. Notice that on each step you need to implement a computation according to (*) to replace the current pair of numbers $\left(r_{k-2}, r_{k-1}\right)$ with the new one, $\left(r_{k-1}, r_k\right)$, where r_k is the remainder of the *integer division* of r_{k-2} by r_{k-1}.

Exercise 5.
Find gcd(5989,4187) using the Euclid algorithm and CAS assistance interactively.

Representation of numbers in different bases

The decimal notation of natural numbers that we are usually using in everyday life is a special case of notation in *the base m*, or base-*m* notation. The long version of the representation in the base-10 of a number with *k* digits has the form

$$n = n_{k-1}10^{k-1} + n_{k-2}10^{k-2} + \ldots + n_1 \cdot 10 + n_0 \quad (**)$$

A brief version of this form is $\left(n_{k-1}n_{k-2}\ldots n_1 n_0\right)_{10}$.

Representation in the base *m* has the form (**) with 10 replaced by *m*.

Converting integers from decimal to binary and vice versa

To convert a binary number to decimal use the formula (**) with the base two. Since the only binary digits are 0 and 1 (called *bits*), you just add powers of two that correspond to 1's in the binary representation.

Example 3. Convert $(10110011)_2$ to the base 10.

Solution

$$1 \cdot 2^7 + 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 179.$$

Answer: $(10110011)_2 = (179)_{10}$.

To convert an integer in the decimal representation to binary, start with the integer in question and implement the integer division by two keeping notice of the quotients and the remainders. Continue the division operations until you get a quotient of zero. Then just write out the remainders in the reverse order.

Example 4. Convert 14 to the base 2.

Solution

$$14 = 7 \cdot 2 + 0$$
$$7 = 3 \cdot 2 + 1$$
$$3 = 1 \cdot 2 + 1$$
$$1 = 0 \cdot 2 + 1$$

Answer: $(14)_{10} = (1110)_2$.

Exercise 6. (a) Convert $(317)_{10}$ to the base 2; (b) Convert $(100110101)_2$ to the base 10. Check your manual solutions with appropriate CAS commands.

(b) (optional) Write a CAS function $int2binary(n)$. The function converts a positive decimal integer n into its binary representation. Test your function on an example of your choice and check the result using appropriate CAS command.

Pythagorean triples

Let a triple (x, y, z) of positive integers be a PPT, that is, $x^2 + y^2 = z^2$ and $gcd(x, y) = 1$.

Examples of PPTs are $(3, 4, 5), (5, 12, 13), (7, 24, 25), (11, 60, 61)$. For any PPT one of the "legs" x and y is odd and the other even.

Theorem: PPT Characterization, version 1. Every PPT with x odd and y even satisfies the system of equations $x = s \cdot t, y = \frac{(s^2 - t^2)}{2}, z = \frac{(s^2 + t^2)}{2}$, where $s > t \geq 1$ are chosen to be any odd mutually prime integers.

For example, choosing $s = 3, t = 1$, we obtain $(3, 4, 5)$. The PPT defined by $s = 7, t = 5$ is $(35, 12, 37)$.

Exercise. Find all PPTs with $x = 45$.

Exercise. Prove that for any PPT the even number y is actually divisible by 4.

Exercise 7. Make a CAS function $my\_PPT(s, t)$ that takes two odd mutually prime integers s, t with $s > t \geq 1$ and returns corresponding PPT. Include a verification of mutual primality. Verification of the inequality condition $s > t \geq 1$ is not required in your code.

Slightly different form of the characterization of PPTs can be obtained using a *rational parametrization* of the unit circle centered at the origin.

To derive a rational parametrization of the unit circle, notice that a triple (a, b, c) of positive integers is Pythagorean iff $\left(\frac{a}{c}\right)^2 + \left(\frac{b}{c}\right)^2 = 1$, that is, the point (x, y) with rational coordinates $x = \frac{a}{c}, y = \frac{b}{c}$ lies on the unit circle. You know that the unit circle can be defined parametrically as $x = \cos(s), y = \sin(s), s \in [0, 2\pi)$. Using the trigonometric identities

$\cos(s) = \cos^2\left(\frac{s}{2}\right) - \sin^2\left(\frac{s}{2}\right), \sin(s) = 2\sin\left(\frac{s}{2}\right) \cdot \cos\left(\frac{s}{2}\right)$, and

$\sin^2\left(\frac{s}{2}\right) + \cos^2\left(\frac{t}{2}\right) = 1$, you can derive the formulas

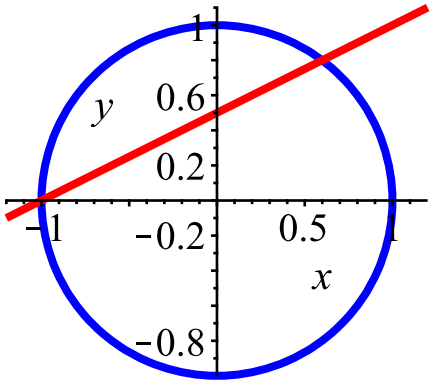
$$\cos(s) = \frac{\left(1 - \tan^2\left(\frac{s}{2}\right)\right)}{1 + \tan^2\left(\frac{s}{2}\right)}, \sin(s) = \frac{2\tan\left(\frac{s}{2}\right)}{1 + \tan^2\left(\frac{s}{2}\right)}.$$

Exercise. Derive the last two formulas.

Introduce notation $t = \tan\left(\frac{s}{2}\right)$ and you arrive to a rational parametrization of the circle

$$x(t) = \frac{(1 - t^2)}{1 + t^2}, y(t) = \frac{2t}{1 + t^2}, -\infty < t < \infty. \qquad (***)$$

The parametrization does not include the point $(-1, 0)$. **(Why?)**
There is another way to derive this rational parametrization of the unit circle based on geometric considerations instead of trigonometry. Consider the unit circle crossed by a non-vertical line passing through the point $(-1, 0)$.



Exercise. (a) Show that the red line $y = t \cdot (x + 1)$ crosses the unit circle at the point $(x(t), y(t))$ with coordinates defined by equations (***).
(b) Use CAS assistance to interactively execute three operations:

- substitute a rational slope $t = \frac{m}{n}$ into equations (***) and simplify the result
- set a to the numerator of $x(t)$, b to the numerator of $y(t)$, c to the denominator of either of the simplified equations
- check that a, b, c is a PT using appropriate logic command of CAS

The exercise suggests a slightly different form of the characterization theorem for PPT in terms of integers *m*, *n*, one of which is odd and the other even. This form is convenient for systematic procedure of constructing large number of PPTs in Lab 1.

Theorem: PPT Characterization, version 2. Every PPT (a, b, c) satisfies the system of equations $a = n^2 - m^2, b = 2m \cdot n, c = m^2 + n^2$, where m, n are mutually prime natural numbers, one of which is odd and the other even, and $m < n$.

Remark. If m and n are odd, then a, b, c defined by the theorem are even numbers, and we obtain a PT but not PPT. For instance, for $n = 3$ and $m = 1$, we have
$a = 8, b = 6, c = 10$.

Exercise.
(a) Find all PPTs with $b = 48$.
(b) Does a PPT with $b = 30$ exist? If yes, find the PPT. If no, explain why.

Exercise 8. Make a CAS function $one\_PPT(t)$ that takes a rational number $t = m/n$ and returns the corresponding PPT or prints "Both *m* and *n* is odd."

Linear Diophantine equation in two variables

Solution to a *linear Diophantine equation*

$a \cdot x + b \cdot y = c,\ a,\ b,\ c \in \mathbb{Z},$ ***(LDE)***

is any pair of integers $x,\ y \in \mathbb{Z}$ that satisfy the equation.

Theorem: The Existence Criterion. The equation ***(LDE)*** has a solution $x, y \in \mathbb{Z}$ iff c is divisible by the $gcd(a,\ b)$.

In symbolic notation this condition is written as $gcd(a,\ b) \mid c$.

To prove that existence of solution implies divisibility of c by the $gcd(a,\ b)$ is trivial.

Proof of the existence of solution to the equation ***(LDE)*** provided c is divisible by $gcd(a,\ b)$ is more challenging. **(Try it!)**

General solution to (LDE)

Let x_0, y_0 be a particular solution to ***(LDE)***, that is, $a\,x_0 + b\,y_0 = c$. Clearly, the modified equation $a\left(x_0 + b\,k\right) + b \cdot \left(y_0 - a\,k\right) = c$ is true for all $k \in \mathbb{Z}$. Thus $x = x_0 + b\,k,\ y = y_0 - a\,k, k \in \mathbb{Z}$ is the *general solution* to the ***(LDE)***.

Finding nonnegative solutions

In applications, the unknowns typically stand for real life quantities only nonnegative solutions are sought. Nonnegative solutions exist if the system of inequalities $x_0 + b\,k \geq 0, y_0 - a\,k \geq 0$ is solvable for $k \in \mathbb{Z}$.

Remark. When (LDE) is solvable, the following facts simplify the solution process.

1. If $a,\ b$ are not mutually prime, you can divide the (LDE) by $gcd(a,\ b)$ to obtain the equivalent reduced equation. So, you can always assume that $a,\ b$ are mutually prime.

2. If $x_0,\ y_0$ solves the equation $a \cdot x + b \cdot y = 1$, then $c \cdot x_0,\ c \cdot y_0$ is a solution to the (LDE). That is why a CAS command that solves an LDE with two unknowns typically takes only a and b and returns a solution to the LDE with rhs $c = 1$.

There is a simple algorithm for solving LDE with two unknowns when the equation is solvable. The example below shows the steps of the algorithm.

Example 5. Find all solutions to the LDE $18\,x + 7\,y = 5$.

Solution

1) Since $18 = 7 \cdot 2 + 4$, we can rewrite the given LDE as a "smaller" one in two new variables: $(7 \cdot 2 + 4) \cdot x + 7\,y = 5 \Rightarrow 7 \cdot (2\,x + y) + 4\,x = 5$, or $7\,u_1 + 4\,u_2 = 5$ with $u_1 = 2\,x + y,\ u_2 = x$.

Repeat this step until the coefficient of the second new variable equals one:

2) Since $7 = 4 \cdot 1 + 3$, we can rewrite the LDE $7\,u_1 + 4\,u_2 = 5$ as $4 \cdot \left(u_1 + u_2\right) + 3\,u_1 = 5$, or $4\,u_3 + 3\,u_4 = 5$ with $u_3 = u_1 + u_2,\ u_4 = u_1$.

3) Rewrite the LDE $4\,u_3 + 3\,u_4 = 5$ as $3\left(u_3 + u_4\right) + u_3 = 5.$

A particular solution to the last LDE is $u_3 = 5,\ u_3 + u_4 = 0 \Rightarrow u_4 = -5$.

Now move backward to find the original unknowns:

$-5 = u_4 = u_1 = 2x + y;\ 5 = u_3 = u_1 + u_2 = 3x + y.$

Solving the system of two linear equations, $2x + y = -5,\ 3x + y = 5,$ you find a particular solution to the original LDE:

$x = 10,\ y = -25.$

Check: $18 \cdot 10 + 7 \cdot (-25) = 5$. It checks!

Exercise 9.

(a) Solve the LDE in Example 5 in two steps. First, following the example, solve the LDE $18x + 7y = 1$, then construct the general solution to the original LDE $18x + 7y = 5$.

(b) Show that the LDE in Example 5 has no nonnegative solutions.

Exercise 10. You have two containers of volume 17 and 55 oz of salt. How would you measure 1 oz of salt? Assume that you also have sufficient salt supply and a large container for this task to pour salt into this container and out of it using your two containers. **Hint:** Set up the LDE for the problem, solve it, and then use the solution and **common sense** to describe the procedure of measuring 1oz of salt.

Lab 1: Plotting legs (a,b) of Primitive Pythagorean Triples

Problem formulation

Write a CAS function $my\_ppts(N)$ that executes the following.

1. Constructs a list L of pairs $[m, n]$ from fractions on N subdiagonals of the table of rational numbers shown below. Here m is the numerator and n the denominator of a fraction.

2. Makes a list S of legs $[a, b]$ of PPTs obtained using the second version of the theorem on characterization of PPT.

3. Plots the points (a, b) from the list S.

Table of fractions from the interval $(0, 1)$

Consider the following two-way infinite table of rational numbers:

$$\begin{array}{cccccc} \frac{1}{2} & \frac{2}{3} & \frac{3}{4} & \frac{4}{5} & \dots \\ \frac{1}{3} & \frac{2}{4} & \frac{3}{5} & \dots & \dots \\ \frac{1}{4} & \frac{2}{5} & \dots & \dots & \dots \\ \frac{1}{5} & \dots & \dots & \dots & \dots \end{array}$$

You will use the table as the source of "building material" for constructing legs a, b of PPT.

Clearly, some fractions in the table are repeated, like $\frac{2}{3}$ and $\frac{6}{9}$, but this can be easily fixed later (see Part 2 in the plan below).

Exercise. What is the length of a list of fractions from N subdiagonals?

Suggested plan
Your code will be more readable if you split the problem into subproblems.

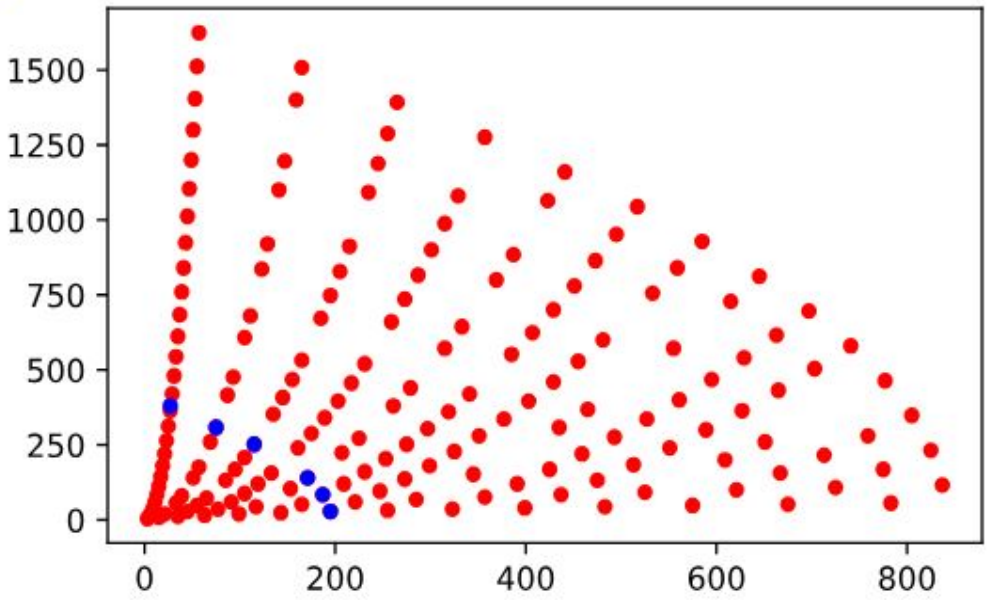
Part 1. Make a CAS function $mn\_pairs(N)$ that takes a natural number N and returns the list L of pairs $[m, n]$ for fractions taken from the first N consecutive subdiagonals going from southwest to northeast in the table described above.

$L = [[1, 2], [1, 3], [2, 3], [1, 4], [2, 4], [3, 4], ...]$.

Part 2. Make a CAS function $ppt\_legs(L)$ that takes a list L of pairs of natural numbers $[m, n], m < n$, constructs the list of corresponding PTs, and returns

- the list S of PPTs without duplicates
- the length of the list of PTs and the length of the list S of PPTs.

Part 3 (main). Make a CAS function $plot\_ppt(N)$ that uses the functions you made in Part 1 and Part 2 and returns the plot of points on the list S. Run this function with $N = 25$. Your figure will look similar to Fig. 2.2. (Six points are made blue just for the next part of the project.)



Legs of PPT's with "odd" legs on the horizontal axis and "even" legs on the vertical axis.

Part 4. You will see some patterns in your figure. It looks like the blue points are located along some arc and there is a missing point in the sequence of six blue points in Fig. 2.2. The points have coordinates

$[195, 28], [187, 84], [171, 140],$ ***missing point***, $[115, 252], [75, 308], [27, 364]$.

Determine the patterns in the sequences of *x*- and *y*-coordinate of these points and find the coordinates of the missing point. Explain why the point is missing.

Part 5. Write a brief summary. Include the numbers of elements on the lists *L, M,* and *S*. If you see some patterns in your plot, briefly describe them. If you did some experimentation, tell us about it.

Remark. An alternative way of visualizing Pythagorean triples is described in the article by Robert Saunders and Trevor Randall "The Family Tree of the Pythagorean Triplets Revisited", The Mathematical Gazette, Vol. 78, No. 482 (Jul., 1994), pp. 190-193.

Lab 2 (optional): Industrial application of a linear Diophantine equation in three variables

This is an educational version of a real industrial problem solved by WCSU undergraduate student Josh Torres in 2015 for Connecticut manufacturing company SWI, Ltd.

Problem formulation

An assembly is made of segments of three types: type 1 of length l_1, type 2 of length l_2, and type 3 of length l_3, where l_j are positive integers. The total length of the assembly M is significantly larger than any of l_j, j=1, 2, 3. Due to different electronic "stuffing" of the segments, the prices of the segments are not proportional to their lengths. The price for one segment is c_1 for type 1, c_2 for type 2, and c_3 for type 3. Make a function that takes integer M, lengths l_j, j=1, 2, 3, and prices c_j, j=1, 2, 3, and returns the number of segments of each type needed to construct the assembly of total length M for the minimal cost. (The cost of labor is not included.)

Mathematical model

Let $x_j \in \mathbb{N}^0$ be the number of segments of type j, $j = 1, 2, 3,$ used in assembly. The problem can be stated as the *integer linear programming problem*:

Minimize $f(x_1, x_2, x_3) = c_1x_1 + c_2x_2 + c_3x_3,\ x_i \in \mathbb{N}^0,\ j = 1..3,$ **subject to the constraint** $l_1x_1 + l_2x_2 + l_3x_3 = M,\ l_j,\ M \in \mathbb{N}.$

In words: In a set of all nonnegative integer solutions of the LDE $l_1x_1 + l_2x_2 + l_3x_3 = M$ find solution(s) that minimizes the objective function f.

Assumptions

- $l_j, j = 1..3,$ are pairwise coprime
- without loss of generality, assume that l_3 denotes the largest length

Suggested plan

Part 1. Transform the LDE constraint with three variables into a family of LDEs with two variables, x_1 and x_2

$$\mathcal{F} = \{l_1x_1 + l_2x_2 = M - l_3x_3, x_3 = 0..max\_x_3\}.$$

The family is parametrized by the admissible values of x_3. These are the values for which the right hand side of the equation $l_1x_1 + l_2x_2 = M - l_3x_3$ is nonnegative. Find the largest admissible value max_x_3.

Part 2. Find all nonnegative solutions of the family $\mathcal{F}$.

Part 3. Calculate the costs for all solutions found on Step 2 and choose solution(s) with minimal cost.

Directions

Write the following CAS functions

1) $isolve2(a, b, c)$ that returns a particular solution to LDE $a \cdot x + b \cdot y = c$. You may use appropriate CAS command.

2) $\boldsymbol{two\_lists}\left(\boldsymbol{M}, \boldsymbol{l_1}, \boldsymbol{l_2}, \boldsymbol{l_3}\right)$ that returns two lists of the same length:

- list $\boldsymbol{L}$ of all particular solutions $\left[\boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}\right]$ with $\left[\boldsymbol{x_1}, \boldsymbol{x_2}\right]$ being solution to 2D LDE in the set $\mathcal{F}$ corresponding to the value x_3.
- list $\boldsymbol{K}$ of ranges for parameter k such that $\boldsymbol{x_1} + \boldsymbol{l_2} \cdot \boldsymbol{k} \geq \boldsymbol{0}$ and $\boldsymbol{x_2} - \boldsymbol{l_1} \cdot \boldsymbol{k} \geq \boldsymbol{0}$ for each element in $\boldsymbol{L}$.

3) $\boldsymbol{sol\_list}(\boldsymbol{L}, \boldsymbol{K})$ that returns a list $\boldsymbol{S}$ of all nonnegative solutions to the LDE constraint.

4) $\boldsymbol{opti\_cost}\left(\boldsymbol{S}, \boldsymbol{c_1}, \boldsymbol{c_2}, \boldsymbol{c_3}\right)$ that takes a list of nonnegative solutions, makes a list of corresponding costs and chooses solution(s) with minimal cost.

Test your code for the problem with parameters

$$\boldsymbol{M = 45,\ l_1 = 3,\ l_2 = 7,\ l_3 = 16,\ c_1 = \$3.75,\ c_2 = \$4.99,\ c_3 = \$5.50.}$$

Answer to the test problem: The optimal solution is $\left[\boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}\right] = [\boldsymbol{1, 6, 0}]$ with cost ***$33.69***.

5) Solve the problem with the same parameters as in the test problem but with total length $\boldsymbol{M = 114}$. Write your answer in the context of the industrial problem.

Glossary

- *prime number*
- *mutually prime (coprime) pair of integers*
- *coprime set of integers; pairwise coprime set of integers*
- *Pythagorean triple; primitive Pythagorean triple*
- *Diophantine equations*
- *binary representation of integers*