One thing I realized while plotting a straight line in p-adic space is that we only considered integer x. Thus, $ax + b$ is an integer and therefore has a normally defined mod p^n (which is just the integer modular arithmetic we are familiar with).

However, if we consider fractions between integers, everything becomes jumpy since we have the unusual modular operation on fractions. For example, for

$$y = 2x + 3$$

in 3-adic space with $n = 3$ precision:

If we plug in $x = \frac{1}{4}$, we have

$$2x + 3 = \tfrac{7}{2},$$

then

$$\tfrac{7}{2} \bmod p^3 = 17$$

(can be verified with GP).

Now if we tweak x just a little bit, for example

$$x' = \tfrac{1}{4} + \tfrac{1}{200} = \tfrac{51}{200},$$

then

$$ax + b = \tfrac{351}{100},$$

and

$$\tfrac{351}{100} \bmod p^3 = 0.$$

So, the line I generated contains only points for integer x. For the fractions between integers, the values could jump to 100 and then suddenly drop to 0, etc. Therefore, the plot may not reflect any property of the p-adic straight line except for the integer part. I'm not sure.

Below is a graph of the p-adic line

$$y = 4x + 3$$

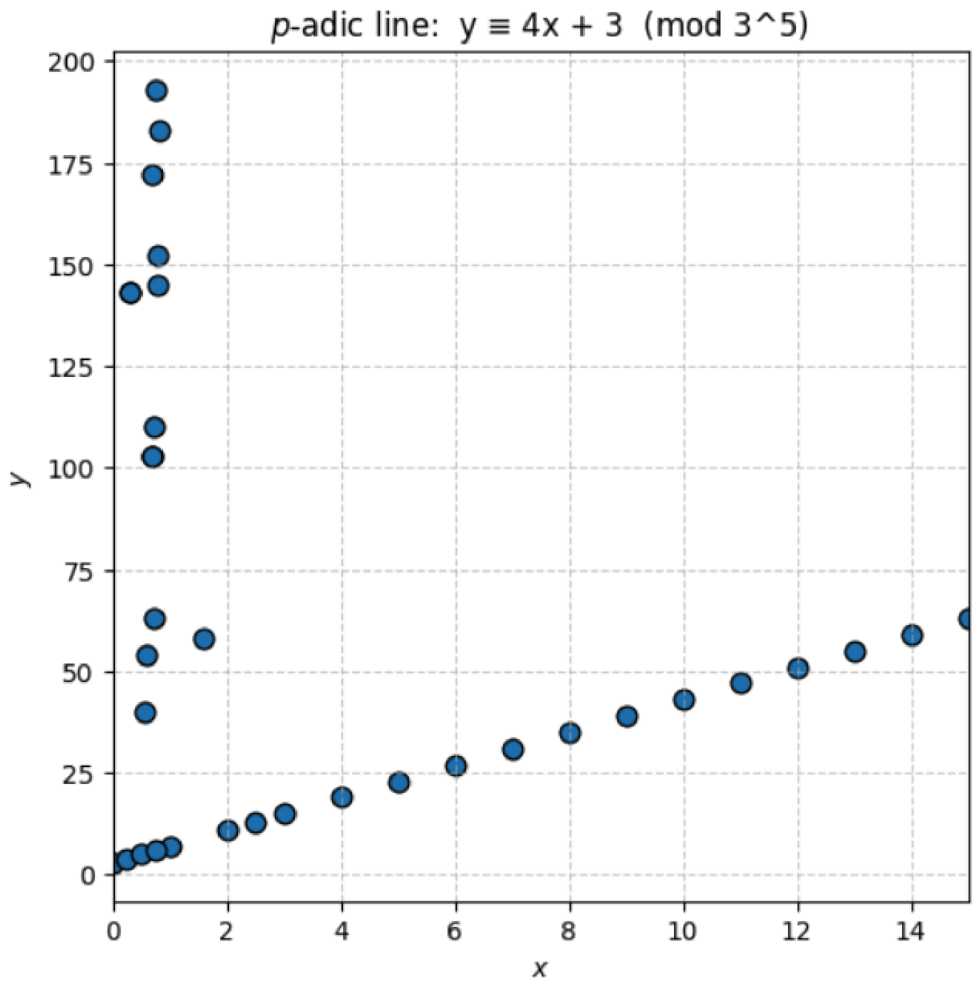

Figure 1: The p-adic line $y = 4x + 3$ in 3-adic space with precision 5, showing randomness when fractions are included.

in 3-adic space with precision 5. When I added some fractions between 0 and 1, everything became random.

Another problem is when I tried to plot the exponential function, sine function, and logarithm function. I used the power series definition, which inevitably results in calculating modular operations on fractions, which again leads back to the jumpy issue I discussed above. I'm not sure if I did it completely wrong, but the graphs for all three functions all seem random, and when I try to change precision n, no matter how much I increase n, the plots do not converge to a single stable plot.

For example, here's a plot of the exponential function that calculated up to term 100 for power series reaching precision $n = 10, 100, 300$ respectively. There's no pattern when I increase the precision, so I'm not sure if I wrote the correct program for calculating exp, sin, and log.

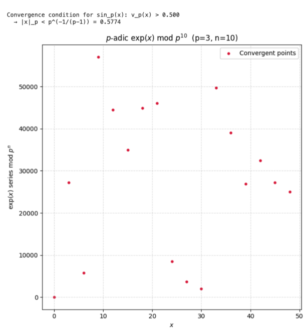

Figure 2: Exponential Function with precision 10.

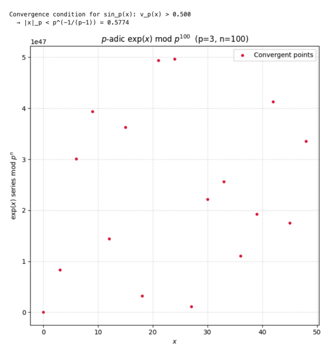

Figure 3: Exponential Function with precision 100.

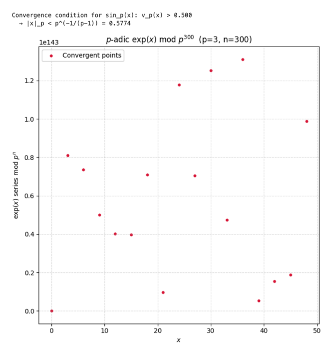

Figure 4: Exponential Function with precision 300.

My fundamental function plotting program is as follow, take exponential as example:

Listing 1: Python code for first n terms of exponential power series

```
def exp_calculation(x, p, terms):
    sequence = []
    for i in range(0,terms+1):
        numerator =int((x ** i))
        denominator = int(math.factorial(i))
        sequence.append(Fraction(numerator, denominator)
            )
    return sum(sequence)
```

Listing 2: Python code for calculating reminder of the power series

```
def p_adic_remainder(r, s, p, n):
    digits, shift = p_expansion(r, s, p, n)
    dim = len(digits)
    if shift < 0:
        return 'Modular not Defined'
```

```
    elif shift >= 0:
        coefficients = []
        for i in range(0, dim+1):
            coefficients.append((p**i))
        remainder = sum(a * c for a, c in zip(digits,
           coefficients))
        return remainder
```

I calculated the reminder of a fraction divided by p to the power of n by first expanding the number in p adic base and then take the first n term of the expansion (there expansion program is guaranteed to be correct as I checked its correwctness using GP)

Listing 3: Python code for calculating reminder of the power series

```
        if convergent:
            frac = func(x, p, terms)
            a, b = frac.numerator, frac.denominator
            if math.gcd(b, p) != 1:
                y_output[i] = None  # noninvertible
                   denominator
            else:
                y_output[i] = p_adic_remainder(a,b,p,n)
        else:
            y_output[i] = None
```

Before this code, I checked for convergent condition of exponential function, and calculated the reminder of the sum of power series using the p-adic reminder function in listing 2