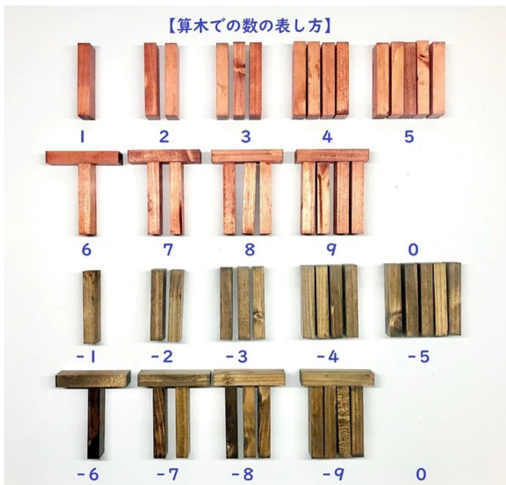
Wasan and Sangi - Japanese Mathematics to solve polynomials

What is *wasan*?

Wasan (和算 in Japanese), refers to the mathematics developed independently in Japan during the Edo Period. During this time, Japan was under *sakoku* (鎖国, closed country), meaning that the only countries available to trade with were the Dutch and the Chinese, and so most foreign influences on Japanese developments came from one of the two. Mathematics was no exception, many mathematical discoveries and ideas were imported from China into Japan. One of these was the idea to use sticks to represent numbers, as the table below:



Courtesy of 工房黒坂製作所

As you can see, the red sticks represent positive numbers and the black sticks represent negative numbers. These sticks in turn were put on a board to do arithmetic, and this was called *sangi* (算木, literal translation: "math sticks"). Mathematician Takakazu Seki expanded upon this concept to apply it to find the integer roots of polynomials with integer coefficients, publishing his results in his book, *Hatsubi-sanpo* (発微算法, published in 1674). The board that he used looked something like this:

万	千	百	十	一	
					商
					実
					方
					廉
					隅

The characters on the topmost row represent the places of each number (in increasing order from the left), 1, 10, 10^2, 10^3 and so on. The rightmost column represents, from the top: the root (商), the constant term (実), the coefficient of x (方), the coefficient of x^2 (廉), the coefficient of x^3 (隅). The ANS row (商) is where we expect the answer to be.

Let's go through, using an example, Seki's algorithm with a rewritten board with western mathematical notation.

10^4	10^3	10^2	10	1	
					ANS
					1
					x
					x^2
					x^3

How it works

We are trying to determine a root of a polynomial using his method, for example, $x^2 + 8x - 273$ whose roots are -21 and 13. Seki did not explain in detail *why* his method works, so I will include the general form of the quadratic and transform it alongside the *sangi* board using western mathematical notation to try and explain why his method works for quadratic polynomials with two-digit roots.

The general form of the quadratic is $Ax^2 + Bx + C$, and one of the roots is $10r_1 + r_2$, where A, B, C are integers and r_1, r_2 are integers from 0 to 9.

The coefficients of the terms are 1, 8, and -273, which are represented on the board as follows: 1 in the x^2 row, 8 in the x row, and -273 in the 1 row. Black font is used for negative numbers and red for positive numbers.

Initial board:

10^4	10^3	10^2	10	1	
					ANS
		2	7	3	1
				8	x
				1	x^2

After placing the numbers in the positions as above, the first step in Seki's algorithm is to estimate the number of digits the root has.

Pretending that we don't know, we estimate that the root has two digits, so Seki's algorithm tells us to move the numbers in the x and x^2 rows 1 and 2 squares left respectively so that the board now looks like step 1.

Step 1:

10^4	10^3	10^2	10	1	
					ANS
		2	7	3	1
			8		x
		1			x^2

Step 2:

10^4	10^3	10^2	10	1	
			1		ANS
		2	7	3	1
			8		x
		1			x^2

In western mathematics, this is equivalent to substituting x with $10x$ because Ax^2 becomes $A(10x)^2 = 100Ax$. Now the board shows the polynomial $100Ax^2 + 10Bx + C$.

Next, we are supposed to estimate what goes in the tens digit of the ANS row (step 2). We will estimate that the tens digit is 1. I will explain how to do this later after covering the whole process.

From here, we apply an algorithm called *renzoku-jouka* (連続乗加, meaning repeated multiplication and addition). No better name could have been given since the process is exactly that! We take the digit in the ANS row, multiply the bottom row (x^2 row) by the digit and add it to the row above, repeating until we hit the 1 row. In this case, the steps are as follows (zeros are added for visibility):

After multiplying the x^2 row by 1 and adding it to the x row:

10^4	10^3	10^2	10	1	
			1		ANS
		2	7	3	1
		1	8	0	x
		1	0	0	x^2

After multiplying the x row by 1 and adding it to the 1 row:

10^4	10^3	10^2	10	1	
			1		ANS
			9	3	1
		1	8	0	x
		1	0	0	x^2

The board after *renzoku-jouka* is as follows:

Step 3:

10^4	10^3	10^2	10	1	
			1		ANS
			9	3	1
		1	8		x
		1			x^2

The first part of step 3 is equivalent in form to $100Ax^2 + (10B + 100Ar_1)x + C$, and the second part is equivalent in form to

$$100Ax^2 + (10B + 100Ar_1)x + (C + r_1(10B + 100Ar_1))$$
$$= 100Ax^2 + (10B + 100Ar_1)x + (C + 10Br_1 + 100Ar_1^2)$$

For a quadratic equation, we do another *renzoku-jouka*, but this time we stop at the x row instead of the 1 row, giving us the following:

Step 4:

10^4	10^3	10^2	10	1	
			1		ANS
			9	3	1
		2	8		x
		1			x^2

In modern notation, this is equivalent in form to

$$100Ax^2 + (10B + 100Ar_1 + 100Ar_1)x + (C + 10Br_1 + 100Ar_1^2)$$
$$= 100Ax^2 + (10B + 200Ar_1)x + (C + 10Br_1 + 100Ar_1^2)$$

The next step is to reverse what we did in the first step, so we 'shift back' the coefficients of x and x^2 back to their original positions, giving us the diagram on the left.

Step 5:

10^4	10^3	10^2	10	1	
			1		ANS
			9	3	1
			2	8	x
				1	x^2

$$Ax^2 + (B + 20Ar_1)x + (C + 10Br_1 + 100Ar_1^2).$$

We then estimate that the one place of the answer is 3, so we put that in the ANS row and then do another *renzoku-jouka* (just once this time).

Step 6:

10^4	10^3	10^2	10	1	
			1	3	ANS
					1
			3	1	x
				1	x^2

In western notation, this is

$$Ax^2 + (B + 20Ar_1 + Ar_2)x + (C + 10Br_1 + 100Ar_1^2 + r_2(B + 20Ar_1 + Ar_2))$$
$$= Ax^2 + (B + 20Ar_1 + Ar_2)x + (C + 10Br_1 + 100Ar_1^2 + Br_2 + 20Ar_1r_2 + Ar_2^2)$$

Now, there are no numbers in the 1 row, which indicates that we have found a root to the quadratic polynomial, 13. The final goal of this algorithm is to have no numbers in the 1 row.

This relates to how we estimate what to put in each digit of the ANS row; by looking at the previous two rows in step 2, we can see that the 1 row has -273 and the x^2 row has 100 in it, thus when applying *renzoku-jouka*, the digit must be 1 because if it was greater than 2 then after two rounds of *renzoku-jouka* the 1 row would end up negative, which is against what the algorithm says.

In step 6, the 1 row is empty, but in western notation, it is equivalent in form to

$$(C + 10Br_1 + 100Ar_1^2 + Br_2 + 20Ar_1r_2 + Ar_2^2)$$

meaning that we can form the equation

$$C + 10Br_1 + 100Ar_1^2 + Br_2 + 20Ar_1r_2 + Ar_2^2 = 0$$

Rearranging in terms of r_1 gives us

$$100Ar_1^2 + (20Ar_2 + 10B)r_1 + (C + Br_2 + Ar_2^2) = 0$$

Using the quadratic formula and then expanding the brackets give us

$$\begin{aligned} r_1 &= \frac{-20Ar_2 - 10B \pm \sqrt{100B^2 - 400AC}}{200A} \\ &= \frac{-20Ar_2 - 10B \pm 10\sqrt{B^2 - 4AC}}{200A} \\ &= \frac{-2Ar_2 - B \pm \sqrt{B^2 - 4AC}}{20A} \\ &= \frac{-B \pm \sqrt{B^2 - 4AC}}{20A} - \frac{r_2}{10} \end{aligned}$$

Thus,

$$10r_1 + r_2 = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A}$$

which shows that Seki's method does indeed give us the correct root of the polynomial.

Generalizations

Takakazu Seki, in his book *Hatubi-sanpo*, gives us generalizations of this algorithm to apply to polynomials of any degree, where n is the number of digits of the root. The steps are similar to the ones we just followed, and they are as follows:

1. Estimate n
2. Apply a **shift** for n places
3. Apply ***renzoku-jouka*** n times
4. **Shift** back 1 place
5. Apply ***renzoku-jouka*** $n - 1$ times
6. Repeat steps 4 and 5 until the ANS row is zero

A shift is defined by moving every digit of the bottom-most row n places to the left, then moving the row above $n - 1$ places, then $n - 2$, and the topmost row (the 1 row) stays the same. The shift in step 5 is moving everything 1 box to the right.

Renzoku-jouka is taking one row, multiplying it by the digit we just guessed, and then adding it to the row above. We repeat this starting from the bottom row up until the 1 row, then return to the bottom and repeat the process until he hit the x row, etc, until we end up adding to the penultimate row if we are applying it n times. For $n - 1$ times, we stop at the x row instead of the 1 row, and so on.

Using these two algorithms in combination allows us to find any roots of polynomials, albeit with some guesswork to find the digits.

Formulae to find roots of polynomials in terms of radicals only exist up to quartics, which was proven by Abel and Galois, but the beauty of using *sangi* to solve polynomials is that this method can be used for polynomials of any order. In addition, this method can be better than the Newton-Raphson method because it does not suffer from issues such as stationary iteration points and each step ending up in a cycle. The main downside to the *sangi* method is that it needs a lot of estimation, including the number of digits of the solution and the actual digits themselves.

In conclusion, this few hundred-year-old method can be used to find the roots of, in theory, any polynomial with integer coefficients. With computers readily available to solve pretty much any equation to any precision, could we still learn something from this method?

References:

関孝和全集 - 平山諦、下平和夫、広瀬秀雄 (*Seki Takakazu zenshuu*, by Akira Hirayama, Kazuo Shimohira, and Hideo Hirose, 1974)

https://en.wikipedia.org/wiki/Abel–Ruffini_theorem

https://en.wikipedia.org/wiki/Newton%27s_method