

HISTORY OF MATHEMATICS | MATLAB PROJECT

Counting the Unknown

Reconstructing numbers from remainders in the Sunzi Suanjing

Essential question: How can the remainders of an unknown number be used to reconstruct that number, and what does a MATLAB implementation reveal about the historical procedure?

Learning goals

By completing this project, you will:

1. interpret and solve a remainder problem from the *Sunzi Suanjing*;
2. explain why the historical coefficients 70, 21, and 15 reconstruct the desired number;
3. implement and compare exhaustive-search and direct-reconstruction algorithms in MATLAB;
4. generalize the method to pairwise relatively prime moduli and investigate non-coprime cases; and
5. distinguish a historical computational procedure from its modern symbolic formulation.

Part I. The problem of the unknown collection

Historical problem. Suppose there is an unknown number of objects. When counted in threes, 2 are left over; when counted in fives, 3 are left over; and when counted in sevens, 2 are left over. How many objects are there?

1. **Solve by hand.** Find the least positive number satisfying all three conditions. Record a systematic method, not only the answer.

2. **Verify.** Show explicitly that your proposed number leaves the required remainder when divided by 3, 5, and 7.

3. **Continue the pattern.** Find the next three positive numbers satisfying the same conditions. Describe the pattern and explain why it occurs.

Part II. Investigating the Sunzi procedure

For remainders a , b , and c upon division by 3, 5, and 7, respectively, the historical procedure calculates

$$70a + 21b + 15c,$$

followed by subtraction of multiples of 105.

4. **Apply the rule.** Use $a = 2$, $b = 3$, and $c = 2$. Show how the procedure produces the least positive solution.

5. **Analyze the coefficients.** Complete the table without using a computer.

Coefficient	Remainder mod 3	Remainder mod 5	Remainder mod 7
70			
21			
15			

6. **Explain.** What special role does each coefficient play? Explain how the three coefficients work together to reconstruct the desired remainders.

Part III. Exhaustive search in MATLAB

Open `Counting_the_Unknown_Student.m`. Complete the section labeled **Method 1**. Your program should test values from 0 through one less than the product of the moduli and stop at the first value satisfying every condition.

7. **Predict.** Before running the code, predict the output for remainders [2 3 2] and moduli [3 5 7].
8. **Test and document.** Run the code. Record the output and explain how the logical test determines whether a candidate satisfies the entire system.

Part IV. The historical algorithm in MATLAB

Complete `Method 2` in the student script. Store the historical coefficients in a vector, calculate the weighted sum, and reduce the result modulo 105.

9. **Test cases.** Test at least three different remainder vectors for the fixed moduli [3 5 7]. Verify every result using `mod(x,moduli)`.

Remainders	Computed solution	Verification

Part V. Generalizing the reconstruction

Open `chineseRemainder.m`. Complete or study the generalized function as directed by your instructor. The function calculates the least nonnegative solution when the moduli are pairwise relatively prime.

10. **Trace the algorithm.** For moduli [3 5 7], identify N , each partial product N_i , and each modular inverse u_i . Explain how these reproduce the coefficients 70, 21, and 15.

11. **New system.** Use the function to solve $x \equiv 1 \pmod{4}$, $x \equiv 3 \pmod{5}$, and $x \equiv 4 \pmod{7}$. Verify the result.

Part VI. Testing the assumptions

Investigate the following systems using search rather than the generalized function:

$$\text{System A: } x \equiv 1 \pmod{4}, \quad x \equiv 2 \pmod{6},$$

$$\text{System B: } x \equiv 1 \pmod{4}, \quad x \equiv 3 \pmod{6}.$$

12. **Compare.** Determine whether each system has a solution. For any solvable system, identify its repeating period. Explain what this reveals about the assumption that the moduli are pairwise relatively prime.

Part VII. Synthesis and reflection

13. **Compare algorithms.** Compare exhaustive search with direct reconstruction. Which is easier to understand initially? Which is more efficient for large moduli? Support your conclusion with evidence.

14. **Historical reflection.** The *Sunzi Suanjing* presents a numerical problem and computational procedure rather than a theorem in modern symbolic notation. What mathematical knowledge is contained in the procedure? What becomes clearer when it is translated into modern notation and MATLAB, and what might that translation obscure?
