Math 105AL – Lab 1

Haoyu Zhang, 25327328

1. Objective

In this project, I will experiment with finite-precision arithmetic using chopping and rounding to study how numerical errors propagate through floating-point operations. Specifically, I will simulate 5-digit algorithm to compute $x \oplus y$, $x \ominus u$, and $(x \ominus u) \oslash w$ (Floating-Point Arithmetic with k-Digit Chopping and Rounding)

2. Procedure

I have implemented two MATLAB functions, “fl_chop.m” and “fl_round.m”, which simulate five-digit floating-point arithmetic by chopping and rounding real numbers to **k** significant digits.

Then, I created a “main.m” file to compute the following expressions: 1. $x \oplus y = x + y$; 2. $x \ominus u = x - u$; 3. $(x \ominus u) \oslash w = (x - u)/w$; For each those expressions, I performed the operations twice: once **using five-digit chopping** and once **using five-digit rounding**.

In each operations, I applied the corresponding function, “fl_chop” and “fl_round”, to simulate the true floating-point model in order to make sure the rounding and chopping occurs after every step.

Lastly, I calculated the actual, absolute, and relative errors between the floating-points and true values. As a result of this, I created the tables for chopping and rounding methods.

3. Results

Chop Results:

Operation	Result	ActualValue	ActualError	AbsoluteError	RelativeError
(a) x ⊕ y	1.0476	1.04761904761905	-1.90476190475852e-05	1.90476190475852e-05	1.81818181817859e-05
(b) x ⊖ u	7.0714e-05	7.07142857142573e-05	-2.85714257306879e-10	2.85714257306879e-10	4.04040363868476e-06
(c) (x ⊖ u) ⊘ w	6.3642	6.36429207857524	-9.20785752356323e-05	9.20785752356323e-05	1.44679995981966e-05

Round Results:

Operation	Result	ActualValue	ActualError	AbsoluteError	RelativeError
(a) x ⊕ y	1.0476	1.04761904761905	-1.90476190475852e-05	1.90476190475852e-05	1.81818181817859e-05
(b) x ⊖ u	7.0714e-05	7.07142857142573e-05	-2.85714257306879e-10	2.85714257306879e-10	4.04040363868476e-06
(c) (x ⊖ u) ⊘ w	6.3643	6.36429207857524	7.9214247641346e-06	7.9214247641346e-06	1.24466706843976e-06

The results show that both chopping and rounding introduce small differences from the true arithmetic results. However, the rounding method provides slightly smaller absolute and relative errors. Among the three operations, I think (x⊖u)⊘w showed the largest error since it subtracts two nearly equal numbers (x and u) caused significant cancellation, and dividing by a small w made the errors.

4. Conclusions

I observe that rounding yields more accurate results than chopping for all three operations. The operation (x⊖u)⊘w exhibits the highest relative error, illustrating the failure. This lab demonstrates how finite-precision algorithm can distort the results in operations. It also highlights the importance of rounding in operations.

5. Challenges and Bugs

1. I forgot to apply “fl_chop” or. “fl_round” after the initial steps, and then it indicates to the errors and failure to further coding and operations.
2. I accidentally used the 5 founding as a code line and it caused a MATLAB syntax error.
3. I did not verify the “chat_r” for rounding was defined before the resuls table. Then the MATLAB gives me the “warning about unrecognized variable error”.