COMPENG 3SK3 – Project 1

Instructor: Dr. Xiaolin Wu

Glen Tsui – tsuig – 400201284

1. **Use pseudo code to describe all necessary details of your parallel algorithm, in particular the assignment of the series terms to the 4 processors and the sequence of carrying out all intermedium additions**

```
Main Script Module() {
    Initialize number display formatting in MATLAB and the total number of terms
    to compute.
    Assign the four parallel processors to call quarterSum(start,finish) to compute
    sums of each quarter section in the series of terms.
    Sum all the computations from each processor to an accumulated total.
    Print the computed sum, the MATLAB true value, and calculated true error.
}

quarterSum(start,finish){
    Initialize output variable sum.
    For loop from start to finish of the quarter section {
        Calculate the sign, whether to add or subtract the term.
        Multiply the sign by the term and add it to sum total.
    }
}
```

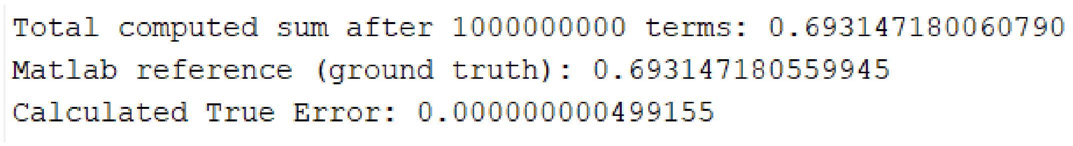
Design decisions: The quarterSum function's parameters "start and finish" are used to depict the limits of the quarter section of terms. One special design decision I made to ensure that there is no overlap of the usage of terms is to floor each term index to account for if the exact quarter point or halfway point is not a whole number. Then by adding one to each start index it will be directly the next index after the last computation of the previous processor. My design decision for the quarterSum function is depicted from the sigma notation of the sum of the alternating harmonic series:

$$\sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n} = \ln 2$$

2. **Simulate your parallel addition algorithm on a serial computer, using Matlab or C. Submit your simulation program with detailed documentation.**

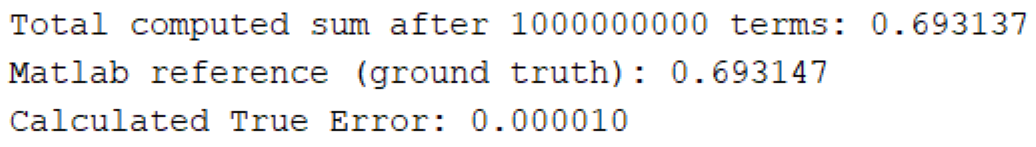
The parallel addition algorithm program is named parallelSumAlgo.m, and the program used to generate the graph for question 4c is named graphing.m. Both programs have thoroughly written comments and are zipped together with this document.

The simulation was first run without applying the single function to view the highest possible precision and a true error of 4.99e-10 was achieved. Viewing this result helped verify that the algorithm is correct in order to sum up the alternating harmonic series.



```
Total computed sum after 1000000000 terms: 0.693147180060790
Matlab reference (ground truth): 0.693147180559945
Calculated True Error: 0.000000000499155
```

Then the code was adjusted to keep all computations and variables in IEEE 32-bit format with the single function and a true error of 1e-5 was achieved.



```
Total computed sum after 1000000000 terms: 0.693137
Matlab reference (ground truth): 0.693147
Calculated True Error: 0.000010
```

3. **Submit a written project report to justify your design decisions, and include both the pseudo code of your parallel algorithm and the simulation program.**
   This written report will be submitted along with the parallel algorithm and graphing programs. This report includes the pseudocode and design decisions in question 1.
4. **In your report, answer the following questions and address the requests.**
   a) **Can you achieve, in the IEEE 32-bit floating number system, any high precision you desire by summing up the first N terms of the series and by running the summation program for a sufficiently large N? Explain your answer.**
   No, because the IEEE standard for floating point representation consists of 32 total bits containing 1 bit for the sign, 8 bits for the exponent with offset, and 23 bits for the binary decimal numbers of the mantissa. The latter section of this floating number system determines the precision of the summation of the series, and only 23 bits are allocated for this. Because there is a maximum 23 bits to represent the decimal points, you cannot achieve any wanted precision if you use a sufficiently large N.

   b) **With respect to the IEEE 32-bit floating number system, what is the minimum N value such that 1/N becomes too small to be representable?**
   The lower bound L on the positive representable value is when s = 0 (positive), and both c and f are at their minimum values (-126 and $(00\ldots0)_2$ respectively).

$$L = m * b^e$$
$$L = 1 * 2^{-126}$$
$$L = 2^{-126}$$
$$N > \frac{1}{L}$$
$$N > \frac{1}{2^{-126}}$$
$$N > 2^{126}$$

   2^{126} is still representable by the floating number system, thus the minimum N value such that 1/N becomes too small must be greater than 2^{126}, or floor(2^{126}) + 1. Using floor is to ensure any decimal value is rounded down, and one is added so that the next term is guaranteed.

What is the minimum N value such that 1/ (N-1)-1/N becomes too small to be representable?

$$\frac{1}{N-1}-\frac{1}{N}=L$$
$$\frac{N}{N(N-1)}-\frac{N-1}{N(N-1)}=L$$
$$\frac{1}{N(N-1)}=L$$
$$1=L*N^2-LN$$
$$0=L*N^2-L*N-1$$
$$0=(N^2-N-\frac{1}{L})$$
$$N=\frac{-(-1)\pm\sqrt{1^2-4(1)(-\frac{1}{L})}}{2}$$
$$N=\frac{1\pm\sqrt{1+\frac{4}{L}}}{2}$$
$$N=\frac{1\pm\sqrt{1+4*(2^{126})}}{2}$$

The minimum N value such that $\frac{1}{N-1}-\frac{1}{N}$ becomes too small to be representable is $floor(\frac{1\pm\sqrt{1+4*(2^{126})}}{2})+1$.

What is the minimum N value such that 1/N is smaller than machine precision?
For IEEE standard, machine precision ϵ_{mach} is 2^{-24}. To find the minimum N value such that 1/N is smaller than machine precision:

$$\epsilon_{mach}>\frac{1}{N}$$
$$N>\frac{1}{\epsilon_{mach}}$$
$$N>\frac{1}{2^{-24}}$$
$$N>2^{24}$$

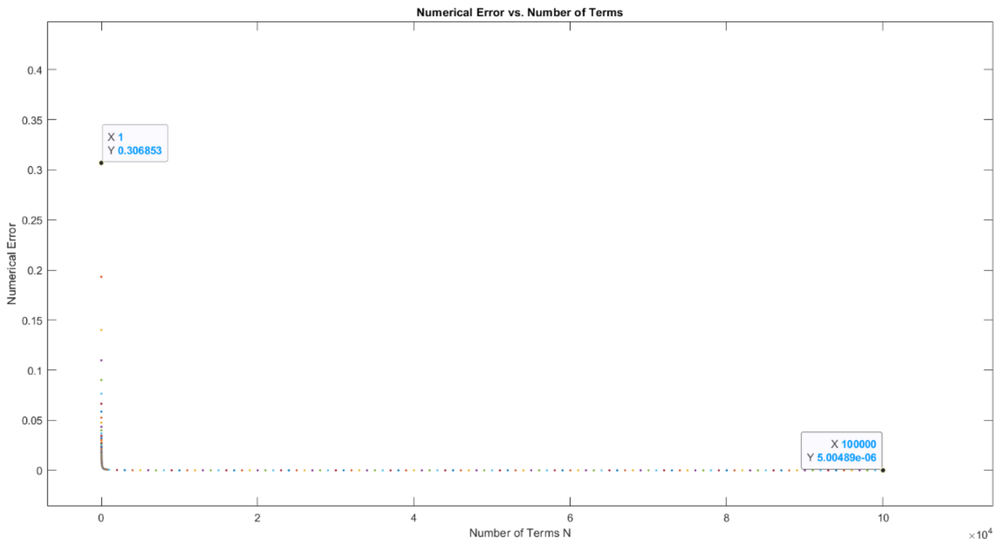
For N to be greater than 2^{24}, then N will have to be one term greater. Then the minimum value of N to be smaller than machine precision is 2^{24} + 1 or 16777217.

What is the minimum N value such that 1/(N-1)-1/N is smaller than machine precision.
Sub ϵ_{mach} into quadratic equation:

$$N = \frac{1 \pm \sqrt{1 + \frac{4}{\epsilon_{mach}}}}{2}$$
$$N = \frac{1 \pm \sqrt{1 + \frac{4}{2^{-24}}}}{2}$$
$$N = \frac{1 \pm \sqrt{67108865}}{2}$$

Taking the plus operator, N is approximately equal to 4096.500031, and the next term must be taken or the minimum value to be smaller than machine precision. Thus, the minimum N value such that $\frac{1}{N-1} - \frac{1}{N}$ is smaller than machine precision is 4097.

c) **For your algorithm to sum the first N terms of the alternating harmonic series, plot the curve of numerical error against N. Explain the behavior of the error curve. To compute the numerical error, the reference (ground truth) is the Matlab double precision value ln2 = Matlab function log(2) = 0.693147180559945....**



The curve of numerical error against N displays an exponential relationship with an asymptote that approaches machine precision. The first data point where number of terms N is 1 and numerical error is calculated to be 0.306853 is due to the first term having a value 1 which is subtracted from the true value of ln(2). As the number of terms increase, the data points of the numerical error begin to decrease towards the asymptote.

d) **Does your algorithm achieve the highest possible precision? If not, suggest a possible way to improve the precision.**

This algorithm does not achieve the highest possible precision. The algorithm implemented currently uses the pairwise summation of numbers, where the long series of terms are broken down into smaller pairs and adding them up. Theoretically, this reduces the accumulation of rounding errors. However, utilizing the IEEE 32-bit floating point format leads to limited precision, which leads to significant errors. As evident in the results from question 2, the numerical error greatly increases when being limited within the IEEE 32-bit floating number system. One option is to use the IEEE 754-2008 standard 64-bit format, and the program would be able to achieve a higher precision. But disregarding the format, the algorithm itself could be improved to achieve the highest possible precision. One algorithm, namely the Kahan summation algorithm, specifically improves the accuracy of numerical summing a list of numbers. This algorithm is especially useful with large sets of numbers and can adjust for errors during the summing process.

5. **This assignment is an individual work.**

Bonus: up to 50% bonus will be given to students who can reach a precision near 10−9 or higher and can be independently verified.