Etude 6 - Numbers

Pair Members: Anthony Dickson, Marcus Lee

1 Harmonic Numbers

Definition The harmonic numbers are defined as the initial partial sums of the series:

$$\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + ...$$

The above can also be represented as follows:

$$H(n) \;=\; \frac{1}{1} \;+\; \frac{1}{2} \;+\; \frac{1}{3} \;+\; .... \;+\; \frac{1}{n-1} \;+\; \frac{1}{n}$$

Method Using the above formula, we created a java application which calculates $H(n)$ using both single and double precision floating point numbers. Our java application computes $H(n)$ in both order of largest term first ($\frac{1}{1} + \frac{1}{2} + ... + \frac{1}{n}$) and reverse order ($\frac{1}{n} + \frac{1}{n-1} + ... + \frac{1}{1}$). The results for calculating $H(10000)$ are shown below in figure 1.1.

Figure 1.1: Results of calculating $H(10000)$.

Largest Term First:
Single Precision: 9.787613
Double Precision: 9.787606036044348

Reverse order:
Single Precision: 9.787604
Double Precision: 9.787604

Observations As we can see in figure 1.1, the results for $H(10000)$ computed from largest term first are different when using single and double precision numbers. However, the single and double precision results are the same

when $H(10000)$ was calculated in reverse order. Furthermore, the single precision results and the double precision results are different based on the order that the terms were added together. These obversations can be seen in $H(n)$ for values on either side of 10000 as shown in figure 1.2 and figure 1.3.

However, the single and double precision results for reverse order in figure 1.3 show a small difference (0.000001). So perhaps $H(n)$ calculated in reverse order is infact different for single and double precision numbers (albeit very slightly and perhaps not significantly).

Conclusion Computing harmonic numbers in both single and double precision did not produce answers that agree. The harmonic numbers computed in single precision tended to be slightly bigger than the harmonic numbers computed in double precision, but only had about half the number of digits.

Computing harmonic numbers in reverse order produced results that mostly agree, and also differed to the results that were produced when computing from the largest term first. However, the results that differ only differ by a very small amount and the difference could be considered to be insignificant.

2 Standard Deviation

Definitions In this report method one refers to the population standard deviation calculated as follows:

$$\sigma = \sqrt{\frac{\sum(a_i - \bar{a})^2}{n}}$$

and method two refers to the population standard deviation calculated as follows:

$$\sigma = \sqrt{\frac{\sum a_i^2 - (\sum a_i)^2/n}{n}}$$

The fixed value added to each number in a is denoted by c.

Method We created a java application which copmutes the standard deviation using both method one and method two. For each method we computed the result in single and double precision. The values for a were generated randomly for most tests. Some tests added a fixed number to each number in a. Values used for n ranged from 10 to $1,000,000$, with values for the fixed value in a similar range.

Figure 2.1: Test with small n value.
$n = 100$
$c = 0$

Method (1):
Single Precision: $\sigma = 27.901434$
Double Precision: $\sigma = 27.90143365492175$

Method (2):
Single Precision: $\sigma = 27.90079$
Double Precision: $\sigma = 27.900786727259145$

Observations From small n values (10-100) to large n values (around 1 million) there seems to be very little difference in the results produced by method one and method two. This can be seen in figures 2.1-3.

However, when large values for c are used method two starts to show some inconsistencies. For example, in figure 2.4 we can observe a difference of about 8.3 (roughly 30% of 27.9) between the single and double preicision results of method two.

In figure 2.5 a c value many times bigger than n was chosen and again, the single precision computation yielded odd results, this time being way off. This may be due to the term $(\sum a_i)^2/n$ becoming too large, possibly causing an integer overflow.

Double precision numbers did not suffer from the problems that single precision numbers did in these experiments. The double precision results remained relatively identical between method one and method two.

Conclusion • The two methods used to compute standard deviation did not always agree.

Figure 2.5: Test with large fixed number.
$n = 100$
$c = 10,000,000$

Method (1):
Single Precision: $\sigma = 27.100739$
Double Precision: $\sigma = 27.100737997331365$

Method (2):
Single Precision: $\sigma = 3,276.8$
Double Precision: $\sigma = 27.092065259038485$

• The level of precision only sems to matter with negative numbers (figure 2.4) or large numbers (figure 2.5).

• Adding a fixed value to all the elements of a sequence of numbers should not change the standard deviation. Most of our implementations showed this, except for the single precision implementation of method two.

• Method one should generally be preferred over method two because method one did not suffer from issues with large numbers in our experiments.

3 An Identity

For identity task, Java code (Identity class) has considered given equation and calculated output for single precision and double precision data types. Output for same is as per figure 3.1.

As per the above output for different inputs, Single precision till single decimal points satisfy the equation. if there are more than one decimals after points then even for single precision point equations fails even though it is mathematically correct. In case of Double precision it fails every time irrespective of given input.

4 Figures

Figure 1.1: Results of calculating $H(10000)$.

Largest Term First:
Single Precision: 9.787613
Double Precision: 9.787606036044348

Reverse order:
Single Precision: 9.787604
Double Precision: 9.787604

Figure 1.2: Results of calculating $H(1000)$.
Largest Term First:
Single Precision: 7.48547843
Double Precision: 7.485470860550343

Reverse order:
Single Precision: 7.4854717
Double Precision: 7.4854717

Figure 1.3: Results of calculating $H(100000)$.
Largest Term First:
Single Precision: 12.090851
Double Precision: 12.090146129863335

Reverse order:
Single Precision: 12.090153
Double Precision: 12.090152

Figure 2.1: Test with small n value.
$n = 100$
$c = 0$

Method (1):
Single Precision: $\sigma = 27.901434$
Double Precision: $\sigma = 27.90143365492175$

Method (2):
Single Precision: $\sigma = 27.90079$
Double Precision: $\sigma = 27.900786727259145$

Figure 2.2: Test with small n and fixed values.
$n = 100$
$c = 10$

Method (1):
Single Precision: $\sigma = 27.901434$
Double Precision: $\sigma = 27.90143365492175$

Method (2):
Single Precision: $\sigma = 27.90079$
Double Precision: $\sigma = 27.900786727259145$

Figure 2.3: Test with large n value.
$n = 775,585$
$c = 0$

Method (1):
Single Precision: $\sigma = 447,564.5$
Double Precision: $\sigma = 447,616.7790107431$

Method (2):
Single Precision: $\sigma = 447,423.94$
Double Precision: $\sigma = 447,476.46364418085$

Figure 2.4: Test with large negative fixed number.
$n = 100$
$c = -10,000$

Method (1):
Single Precision: $\sigma = 27.901434$
Double Precision: $\sigma = 27.90143365492175$

Method (2):
Single Precision: $\sigma = 36.203865$
Double Precision: $\sigma = 27.900786708008454$

Figure 2.5: Test with large fixed number.
$n = 100$
$c = 10,000,000$

Method (1):
Single Precision: $\sigma = 27.100739$
Double Precision: $\sigma = 27.100737997331365$

Method (2):
Single Precision: $\sigma = 3,276.8$
Double Precision: $\sigma = 27.092065259038485$

Figure 3.1 Output:
Single Precision Input x = 2.5 y = 3.5
Single Precision Left Side: 2.5 Right Side: 2.5 Result: $true$
Double Precision Input x = 2.5848 y = 3.5234
Double Precision Left Side: 2.5848 Right Side: 2.5848000000000005 Result: $false$

Single Precision Input x = 2.51 y = 3.5
Single Precision Left Side: 2.51 Right Side: 2.5100002 Result: $false$ Double Precision Input x = 2.5848 y = 3.5234
Double Precision Left Side: 2.5848 Right Side: 2.5848000000000005 Result: $false$