COP5522

PROJECT PROGRESS REPORT

FAROOQ MAHMUD AND DAE HYUN SUNG

1. PROGRESS SUMMARY

We chose to do the project on Fast Fourier Transform (FFT). We were able to successfully create a program that can take samplings of multiple sine waves for the period of 2π, and sum them to create a complex waveform upon which to perform FFT. We successfully recreated the Cooley-Tukey algorithm with big $O(nlogn)$ using complex math operations to perform calculations at each sampling as well as another implementation using two for loops with equal big $O(nlogn)$. We verified the calculation results of the algorithms to be correct which sets us up for the next portion of project which will be to parallelize the algorithm.

2. COMPLEX NUMBER MATH BACKGROUND

The main operations performed in the program are trigonometric sin(Θ), cos(Θ), complex summation, and complex multiplication. Sine wave defined using complex numbers equals

$$r * cos(\Theta) \; + \; ir * sin(\Theta)$$

where Θ is phase and r is amplitude. The summation operation of complex umbers is simple in that one only must add the real parts together and the imaginary parts together. The multiplication of complex numbers is defined as below.

$$(a + bi) * (c + di) = (ac - bd) + (ad + bc)i$$

3. ALGORITHM & OPTIMIZATION

There exist multiple variations of the Cooley-Tukey algorithm implementation for FFT, all fundamentally having big $O(nlogn)$. We tested two different implementations, one by recursion, another iterating through two for loops. Of the two, the implementation that iterates through two for loops was much more performant even though both implementations are big $O(nlogn)$. Therefore, we will likely go with the iterative solution which is better-suited for parallelization and has better serial performance.

Power of 2	N	Recursion Time (ms)	Iterative Time (ms)
4	16	0.00233333	0.000333333
8	256	0.0276667	0.006
12	4096	0.619	0.175
16	65536	19.973	3.79767
20	1048576	266.644	72.6267
24	16777216	4933.23	1406.08
28	268435456	196077	26245.6

Figure 1: Recursive vs. Iterative Performance