

Project #1

Numeric Integration with OpenMP

Dylan Washburne

CS 475

I. GENERAL QUESTIONS

I ran this on my personal desktop computer at home. It has decent processing power to work with.

From calculating the volume via both singular threads and using linear regression to extend the data towards infinity, I believe the actual volume of the figure to approach the volume of 14, as more nodes are added and calculated. As a general note for this, singular threads gave consistent answers which kept declining as more nodes were added. I got up to 16k nodes on a single thread before the program wouldn't respond, and it kept showing signs of the area falling.

II. DATA TABLE

NUMNODES	NUMT	Time (Milliseconds)	MegaHeights/Sec
500	1	35000.1	7.14
500	2	18999.8	13.15
500	4	17000	14.70
500	8	13000	19.23
500	16	9999.99	25
1000	1	97000.1	10.31
1000	2	52000	19.23
1000	4	47000.2	21.28
1000	8	43000	23.25
1000	16	41000.1	24.39
2000	1	349000	11.4613
2000	2	188000	21.27
2000	4	166000	24.09
2000	8	153000	26.14
2000	16	149000	26.84

III. GENERAL QUESTIONS (CONT.)

In general, it seems having more nodes caused the end result to show more megaheights computed per second. In addition, adding additional threads increased the speed. In both cases however, the gain in speed was always exponentially less for each node added, meaning we hit a point of diminishing returns where simply adding more nodes would not aid us.

I think it is behaving this way simply due to the laws of efficiency behind this, such as where the basic amount of time needed for a single thread to complete one iteration of the problem cannot be reduced by adding more resources. The only way to optimize that speed is to rewrite the equation itself.

The parallel fraction for this data comes out to be 0.9689, or an approximately 97% speedup.

From the speedup of 0.9689, we can calculate a maximum speedup of 32.15 times this program could ever attain.