How to get a Good Graphing Score on your Parallel Programming Project

Oregon State University

Mike Bailey

mjb@cs.oregonstate.edu

Why Are These Notes Here?

This class *requires* you to do graphs that are scientifically literate.

These notes are here to help you do that in the shortest amount of time possible.

You have to do it anyway – why not do it and have time leftover?

Diverting to a File

We all have a tendency to want to write our performance results out using *printf* (or *cout*) so that we can see them on the screen. That's fine. But, then we want to get those results into a file. You could mess with file I/O, or you could use a *divert* on the command line.

If you are currently running your program like this:

./proj01

and it prints to the console via *printf*, *cout*, *fprintf* or *cerr*, then running it like this:

./proj01 >& output.csv

will write your results into the file *output.csv*

(If you do it a second time, you will probably have to remove the previous *output.csv* first.)

You can also divert the *entire* output of a looping ***script*** like this:

bash script.bash >& output

Importing into Excel – csv Files

csv stands for **comma-separated values**. It is a file format where you write your numbers out as text with commas between them. The great part is that Excel recognizes csv files and will read them in automatically.

Say you are using an fprintf that looks like this:

```
fprintf( stderr, "%2d threads ; %8d trials ; probability = %6.2f%% ; megatrials/sec = %6.2lf\n",
        NUMT, NUMTRIALS, 100.*currentProb, maxPerformance);
```

You probably did this because it looks really nice on your screen as you use this output to debug your program. But now you want to change it to get the numbers into Excel quickly and painlessly. Comment out the old way and change it to this:

```
//fprintf( stderr, "%2d threads ; %8d trials ; probability = %6.2f%% ; megatrials/sec = %6.2lf\n",
        //NUMT, NUMTRIALS, 100.*currentProb, maxPerformance);

 fprintf( stderr, "%2d, %8d, %6.2lf\n", NUMT, NUMTRIALS, maxPerformance );
```

This will now be printing just what you need to create a file in CSV format. You could divert it like this:

```
./proj01   >&   OUT.csv
```

or

```
bash  script.bash  >&  OUT.csv
```

Which would then let you read the OUT.csv file right into Excel.

Importing into Excel – csv Files


```
flip3 154% cat OUT.csv
1, 1, 1.44
1, 10, 3.99
1, 100, 8.07
1, 1000, 9.33
1, 10000, 23.40
1, 100000, 25.13
1, 500000, 25.97
2, 1, 0.23
2, 10, 4.62
2, 100, 19.26
2, 1000, 17.91
2, 10000, 34.34
2, 100000, 49.83
2, 500000, 49.27
4, 1, 0.34
4, 10, 0.259
4, 100, 16.7
4, 1000, 38.66
4, 10000, 82.39
4, 100000, 91.09
4, 500000, 91.49
8, 1, 0.26
8, 10, 2.39
8, 100, 16.21
8, 1000, 48.49
8, 10000, 137.59
8, 100000, 166.17
8, 500000, 181.62
flip3 155%
```

	A	B	C
1	1	1	1.44
2	1	10	3.99
3	1	100	8.07
4	1	1000	9.33
5	1	10000	23.4
6	1	100000	25.13
7	1	500000	25.97
8	2	1	0.23
9	2	10	4.62
10	2	100	19.26
11	2	1000	17.91
12	2	10000	34.34
13	2	100000	49.83
14	2	500000	49.27
15	4	1	0.34
16	4	10	0.259
17	4	100	16.7
18	4	1000	38.66
19	4	10000	82.39
20	4	100000	91.09
21	4	500000	91.49
22	8	1	0.26
23	8	10	2.39
24	8	100	16.21
25	8	1000	48.49
26	8	10000	137.59
27	8	100000	166.17
28	8	500000	181.62

Making Graphs

In Excel, I have had the most success with creating graphs from tables that look like this:

	A	B	C
1	1	1	1.44
2	1	10	3.99
3	1	100	8.07
4	1	1000	9.33
5	1	10000	23.4
6	1	100000	25.13
7	1	500000	25.97
8	2	1	0.23
9	2	10	4.62
10	2	100	19.26
11	2	1000	17.91
12	2	10000	34.34
13	2	100000	49.83
14	2	500000	49.27
15	4	1	0.34
16	4	10	0.259
17	4	100	16.7
18	4	1000	38.66
19	4	10000	82.39
20	4	100000	91.09
21	4	500000	91.49
22	8	1	0.26
23	8	10	2.39
24	8	100	16.21
25	8	1000	48.49
26	8	10000	137.59
27	8	100000	166.17
28	8	500000	181.62

E	F	G	H	I	J	K	L
	1	10	100	1000	10000	100000	500000
1	1.44	3.99	8.07	9.33	23.4	25.13	25.97
2	0.23	4.62	19.26	17.91	34.34	49.83	49.27
4	0.34	0.259	16.7	38.66	82.39	91.09	91.49
8	0.26	2.39	16.21	48.49	137.59	166.17	181.62

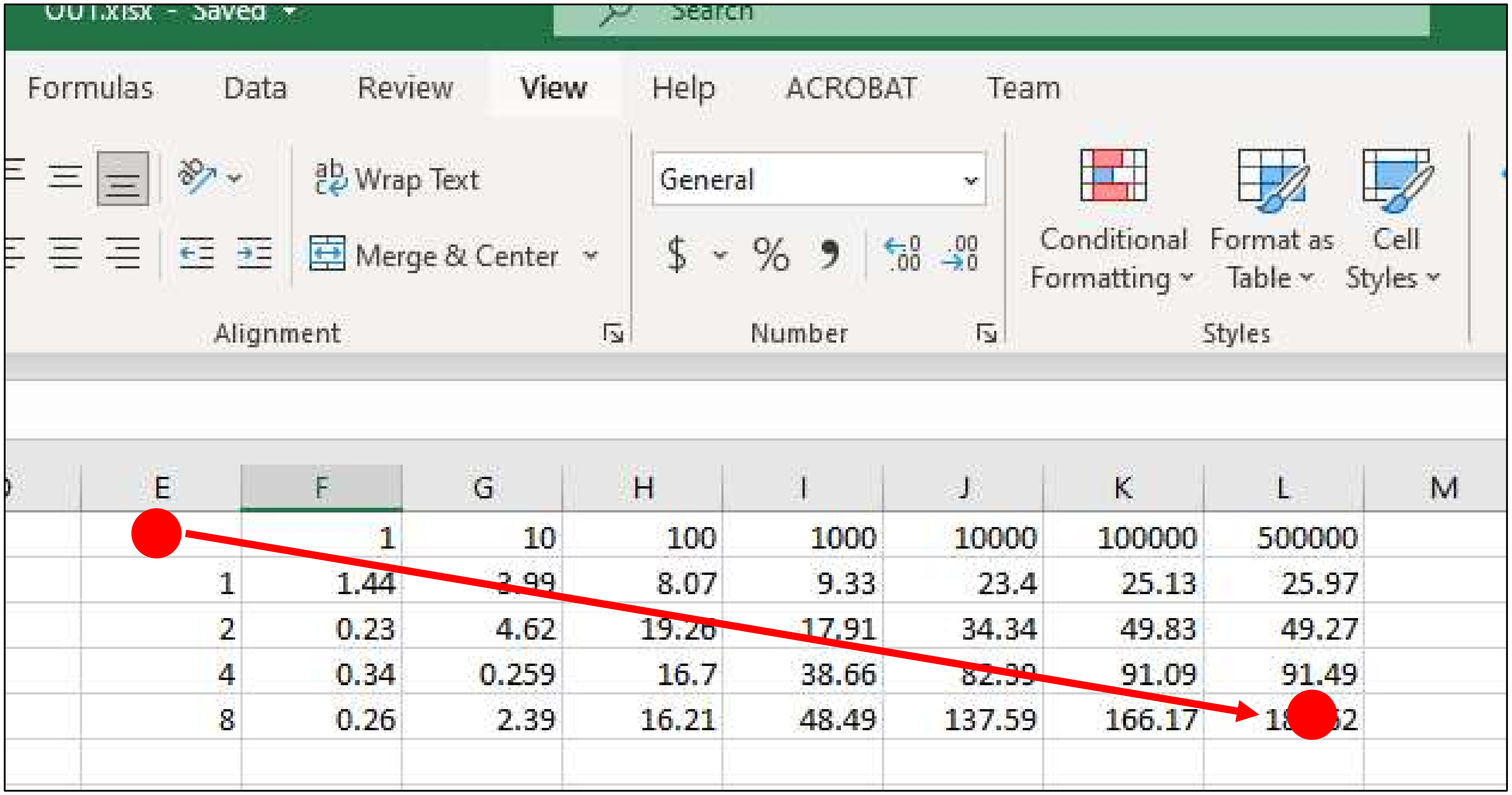
where the 1,2,4,8 rows are holding the number of threads constant, and the 1, 10, 100, 1000, etc. columns are holding the dataset size constant. The cells themselves are holding performance numbers, with higher numbers representing faster performance.

You will need to do some copying and pasting to get the linear format (from the previous page) into this 2D format, but it will be worth it!

But, you can avoid *all* that copying and pasting by using an Excel feature called **Pivot Tables**! See the Pivot Table noteset.

Making Graphs

Then, sweep over the entire table

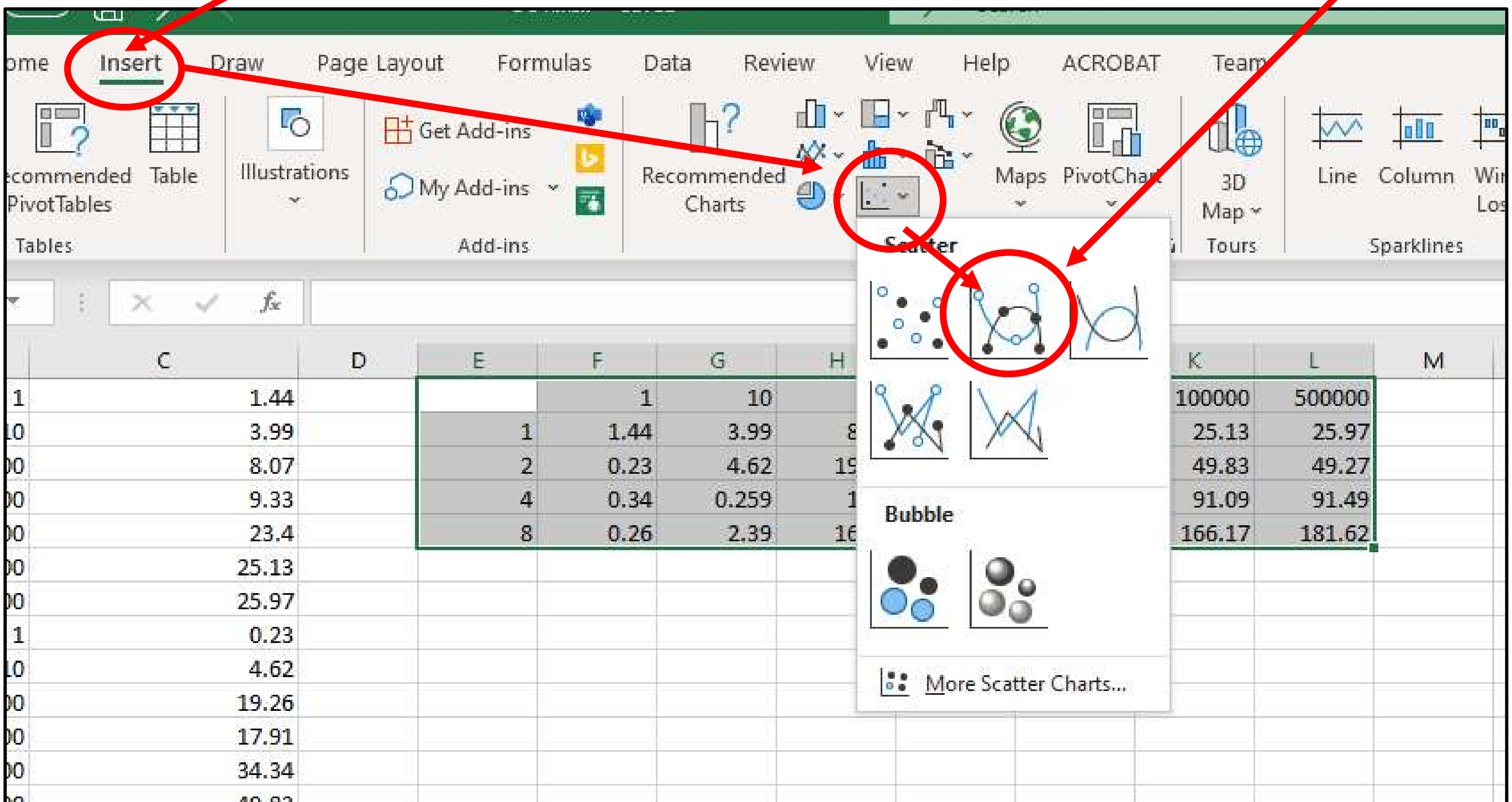


Oregon State University
Computer Graphics

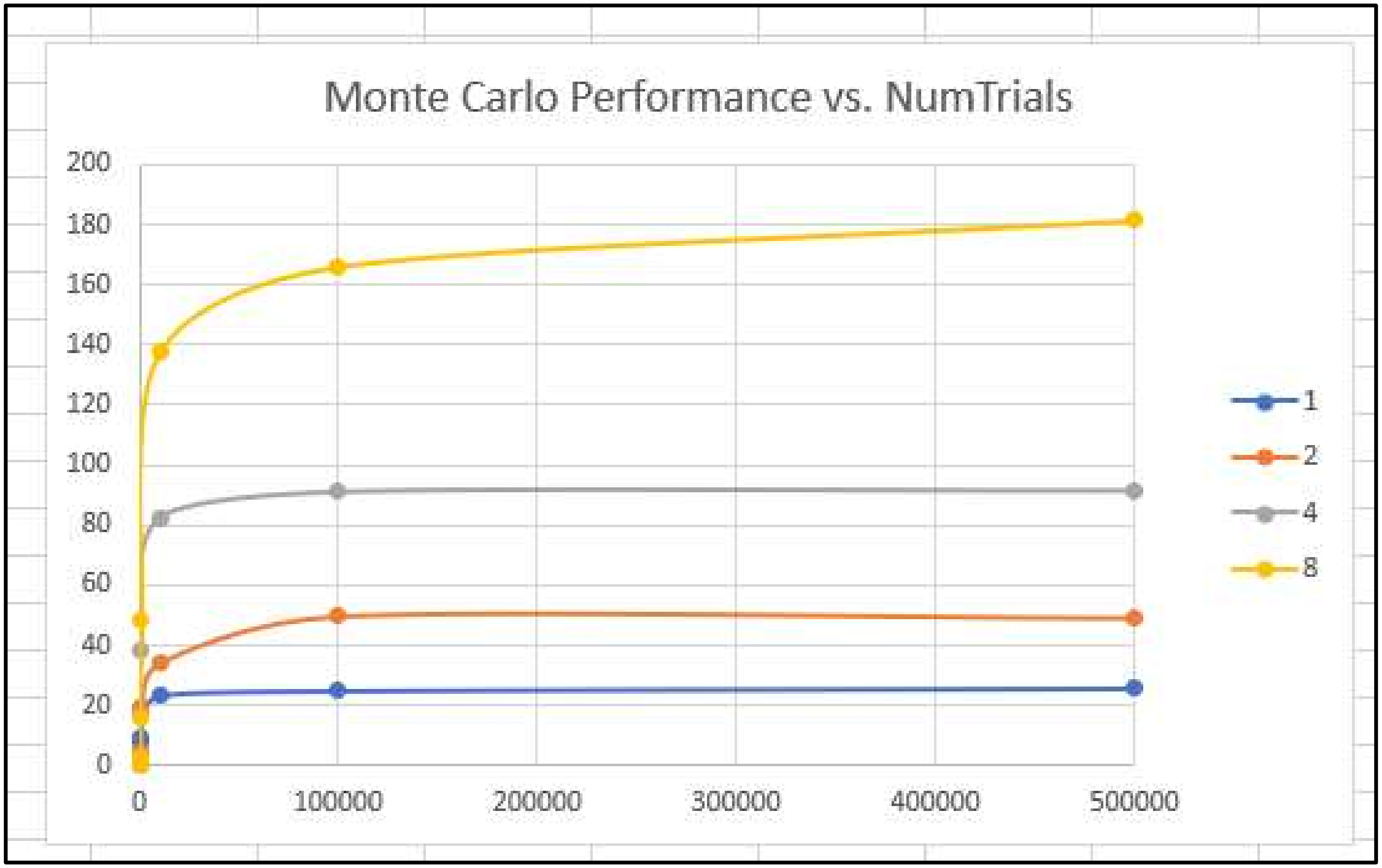
Making Graphs

Then, select Copy, and then insert it into one of the **scatterplot** options.

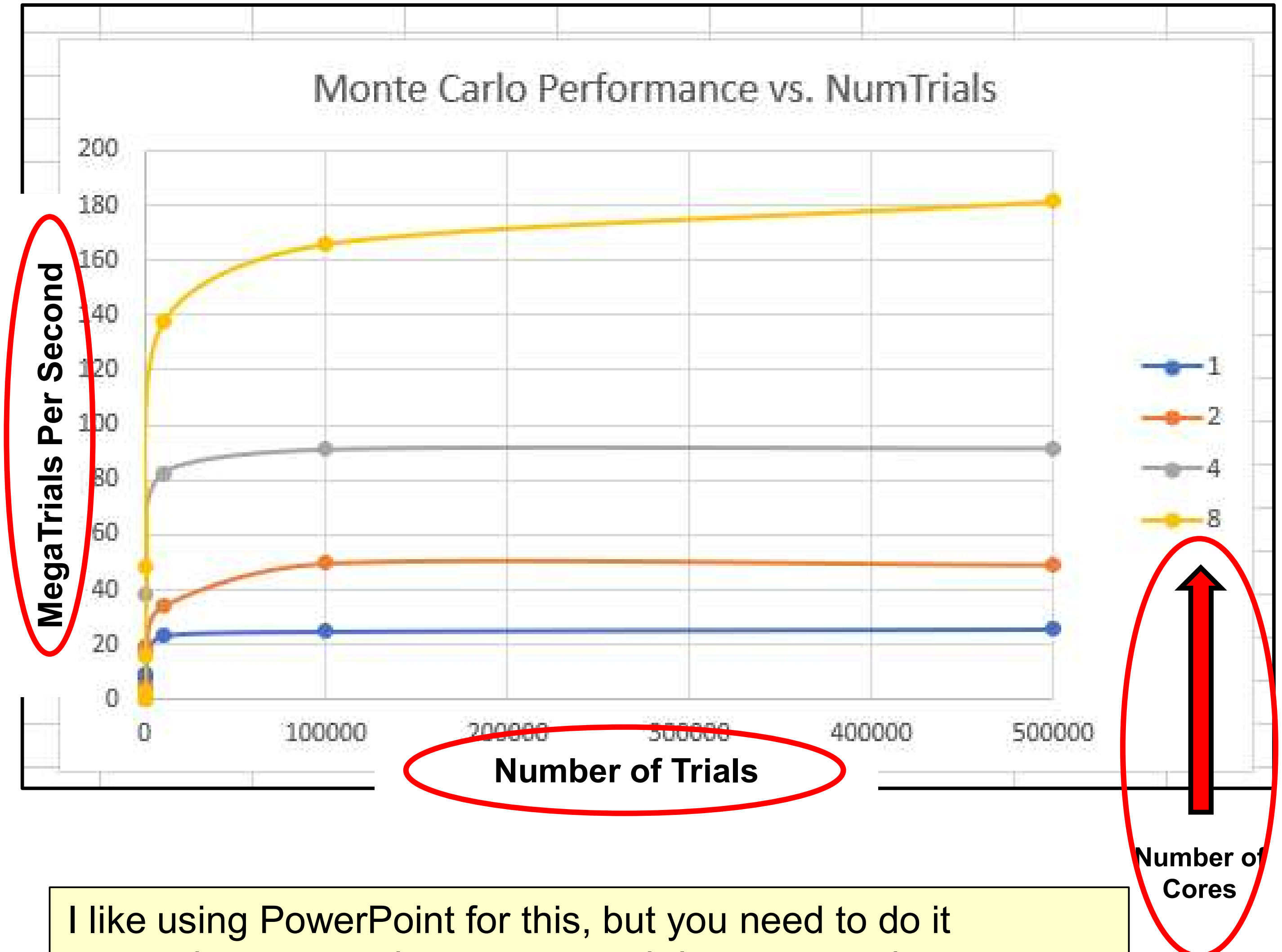
I like this one:



Here's the Graph that Excel Produced from this Data



Adding Annotation

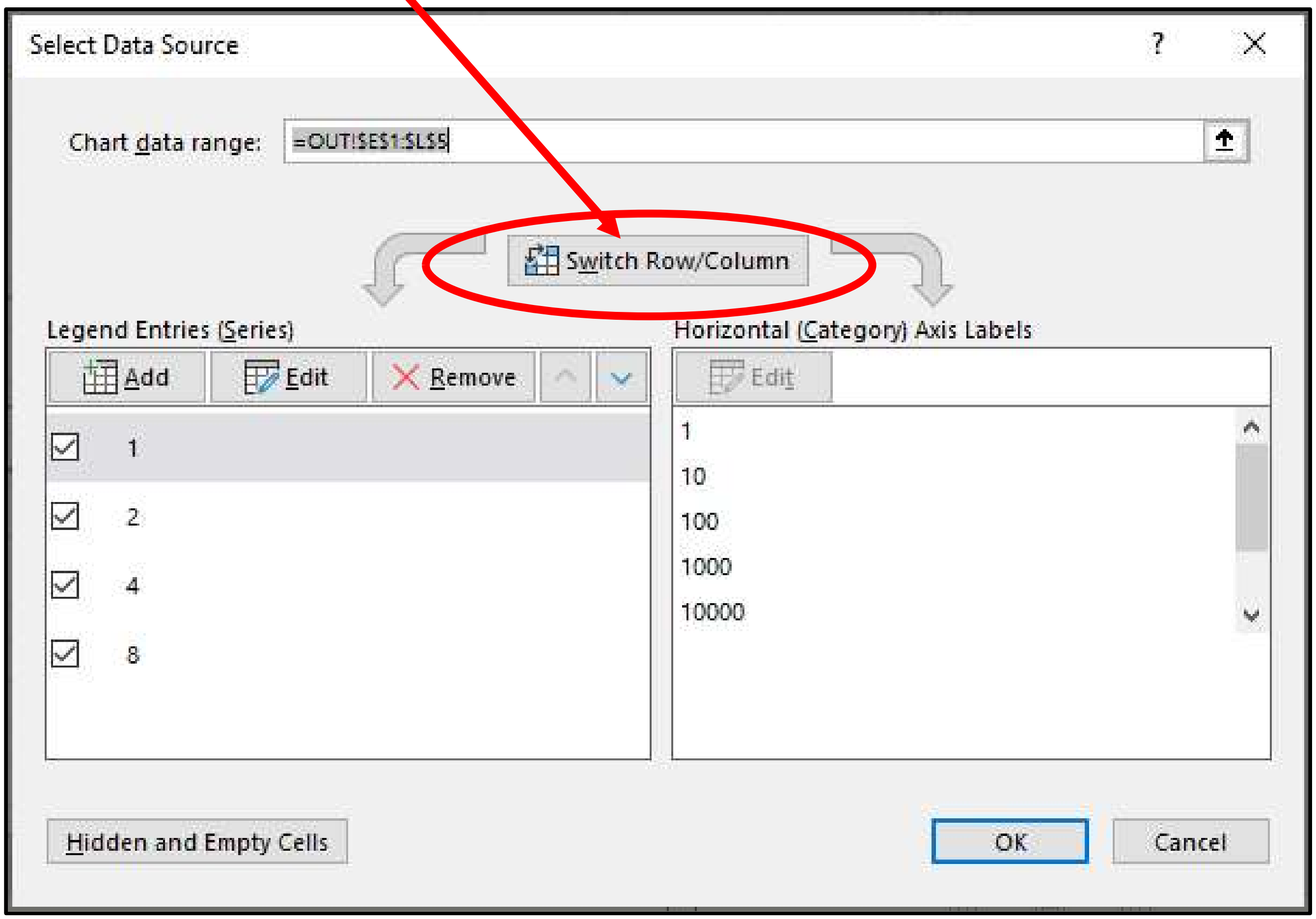


I like using PowerPoint for this, but you need to do it *somewhere* to get the proper graph into your project report.

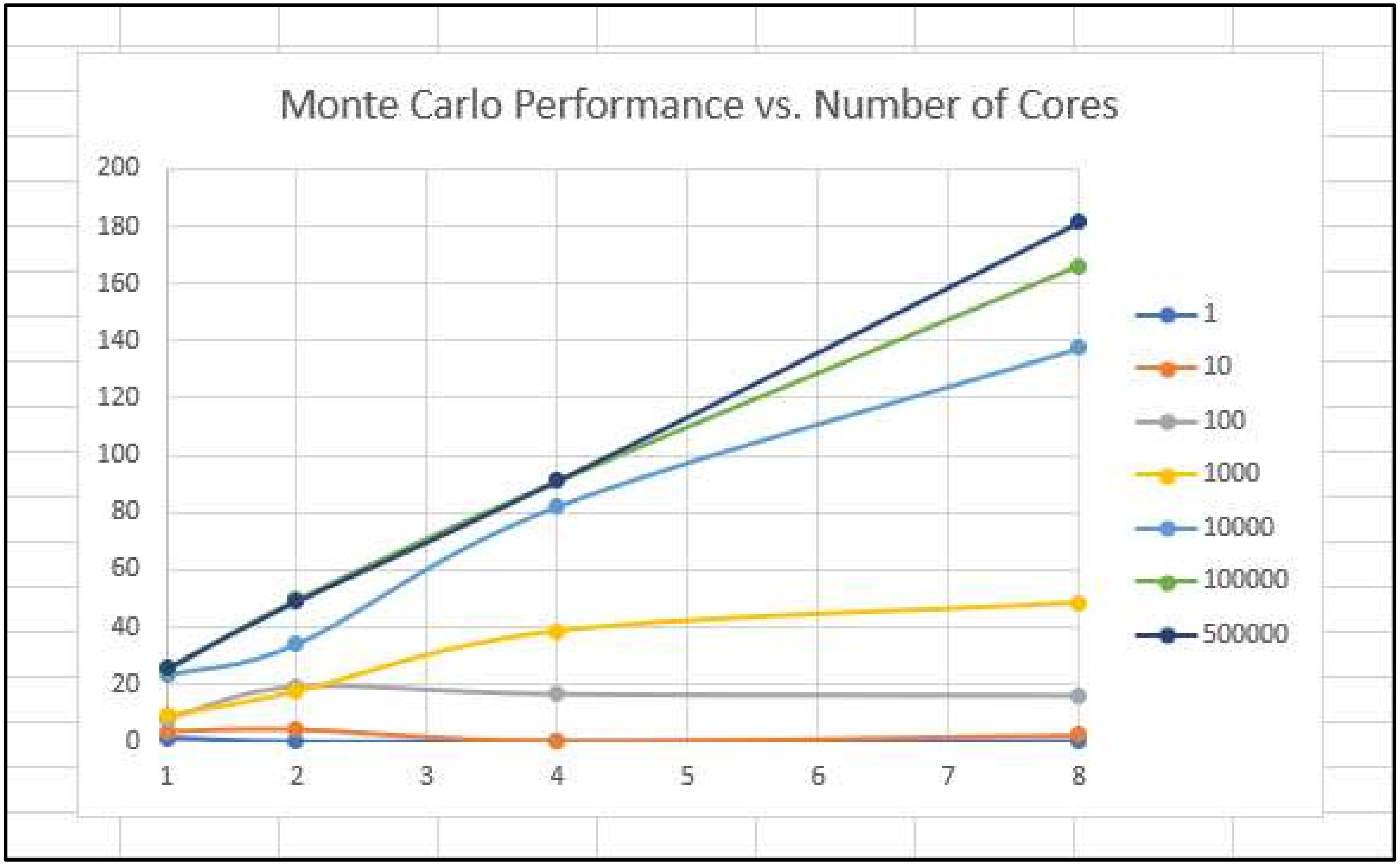
Transposing the Graph

To transpose the sense of the graph (which you also need to do), right-click on the border of the graph and then click on "**Select Data**".

Then click on "**Switch Row/Column**".



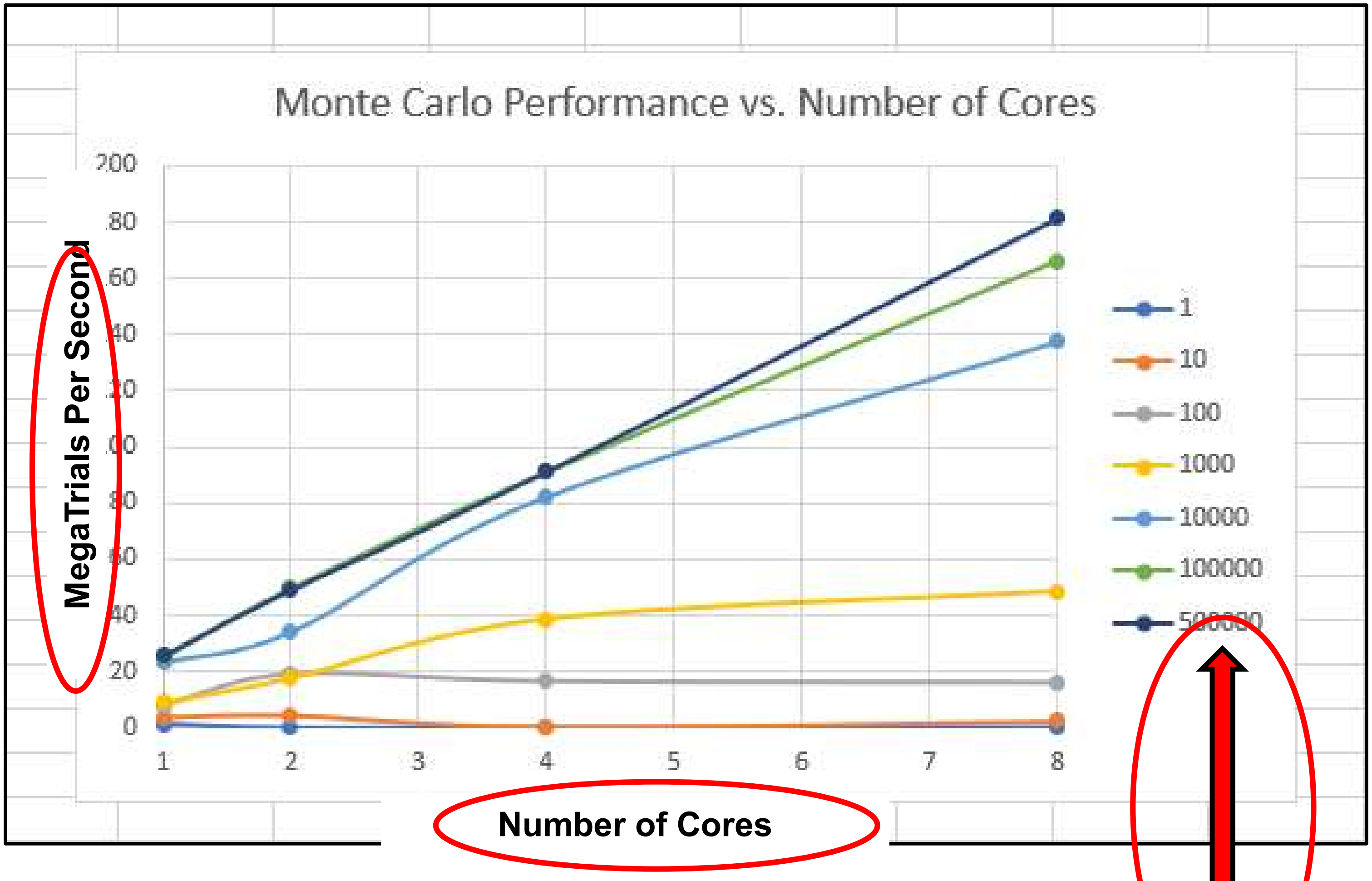
Here's the Graph that Excel Produced from this Data



You might have to adjust the X axis number maximum and minimum.

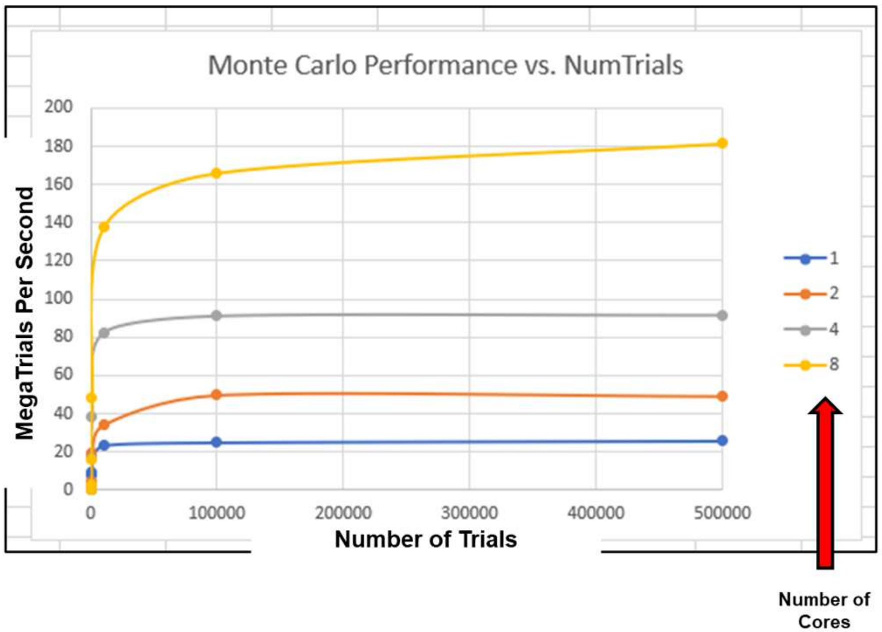
Oregon State
University
Computer Graphics

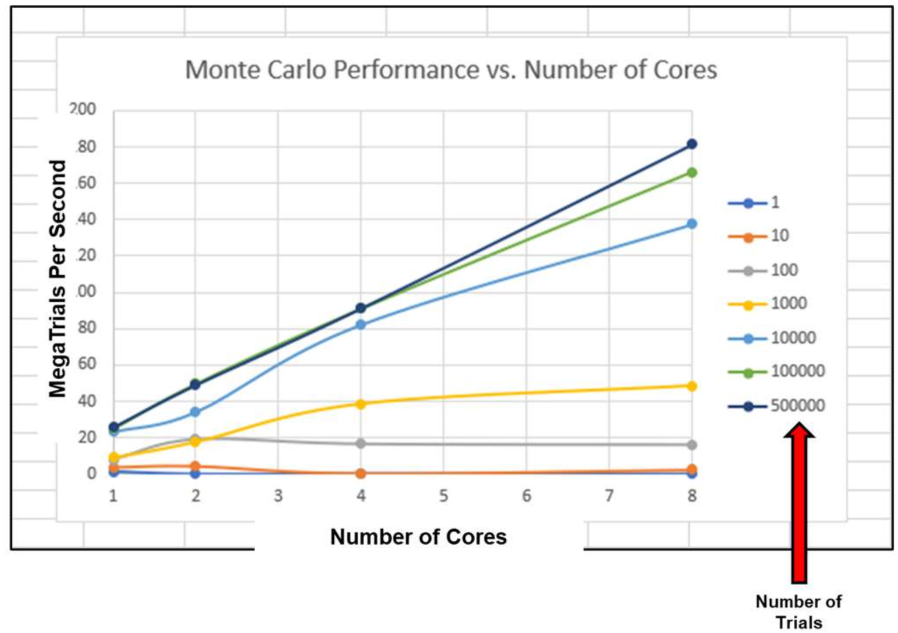
Adding Annotation



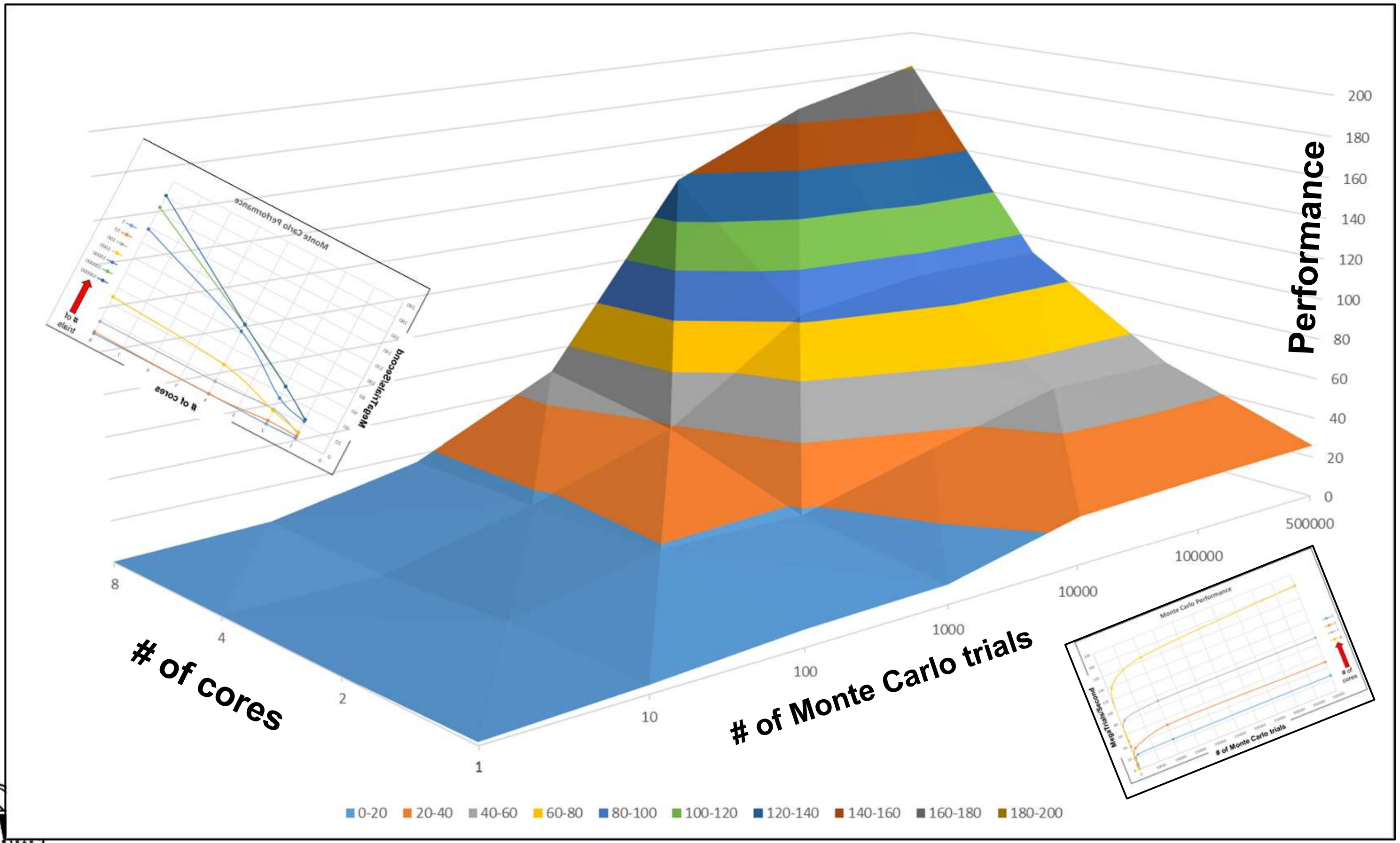
I like using PowerPoint for this, but you need to do it *somewhere* to get the proper graph into your project report.

It's the Same Data, but Each Graph Gives You a Different Insight into what the Data is Telling You





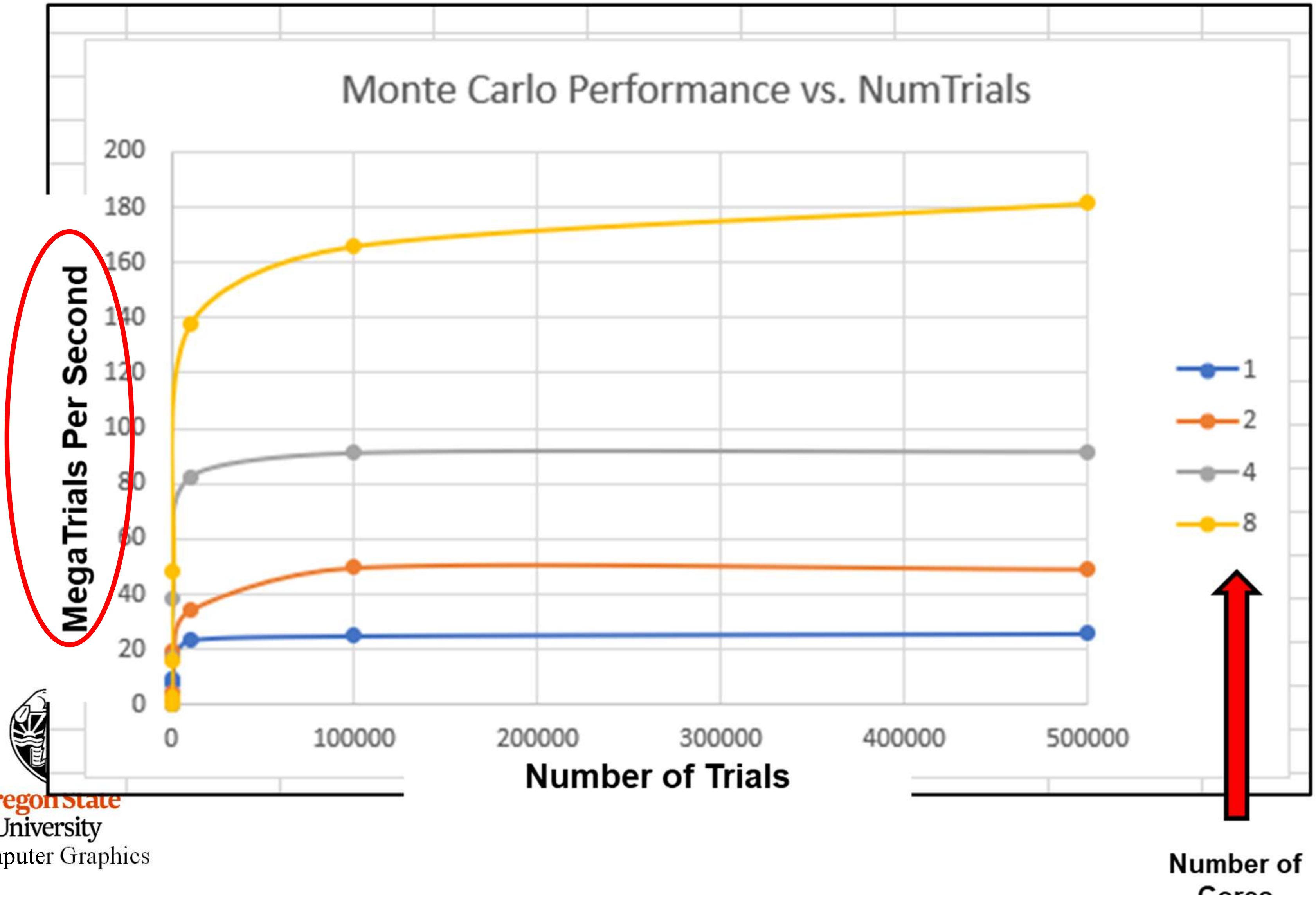
This Data is actually a *3D* Surface Plot – The 2D Graphs are sets of 2D *slices* through the 3D Surface



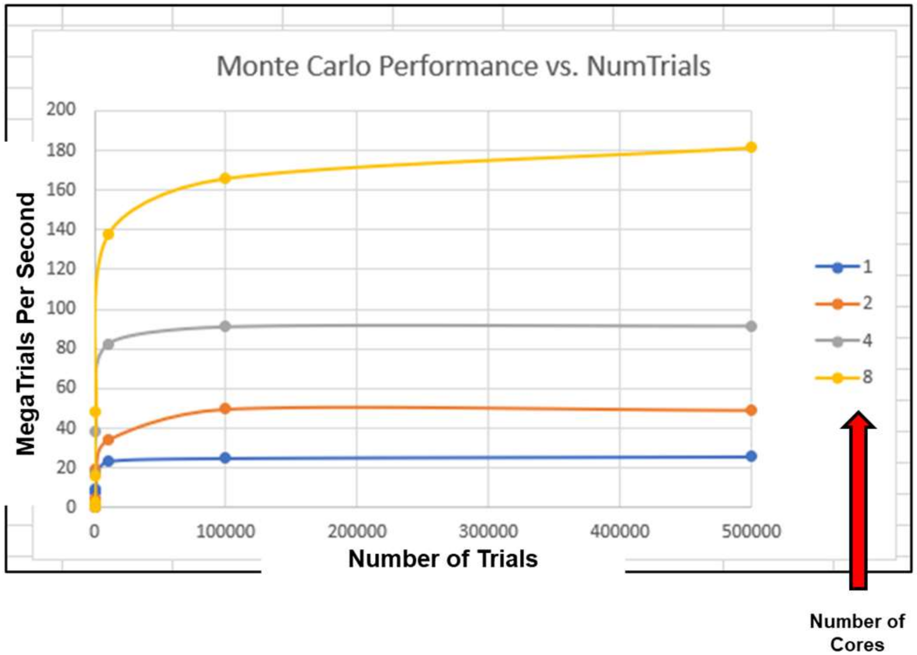
Oregon State University
Computer Graphics

Making Graphs

When we plot, we will all put ***execution performance on the Y axis*** (as opposed to putting elapsed time on the Y axis). Thus, as far as performance goes, ***up will mean "good"***. So, for example:



Making Graphs



As you can tell, these performance measurements will be far more intelligible when examined as a graph than as raw numbers. ***Thus, you are expected to have access to a good automated graphing package.*** If you don't have one now or can't get access to one – go get one!

Hand-drawn graphs, whether analog or digital, will not be accepted for your assignments.

You will also need a word processor, with a way to import your tables and graphs, and with a way to turn that document into PDF.