Dimensions	Runtime Serial seconds	Runtime Parallel on Darwin seconds	Runtime Parallel Float Version
640x360x10	1.107252	.001838	.002973
1280x720x15	6.547220	.002240	.003558
1920x1080x20	19.478880	.002246	.002626
2560x1440x25	43.558538	.001748	.003329
3200x1800x30	113.509289	.002094	.002917
3840x2160x35	…	.002200	.002241
4096x2160x40	…	.002298	.002208
4096x2160x45	…	.002983	.002504
4096x2160x50	…	.002004	.004040
4096x2160x60	…	.002678	.002603

The runtime for the serial version of the fractal program will continue to increase for the larger images and throughout each dimensional input we maintain around a 1000 speedup or greater. Note that for the runtimes on darwin I was not consistently getting the same runtime for the same dimensional input which could be due to the allocation of resources on the hpc server or some overhead that is not correlated to the parallel program. I say this is not due to the parallel program due to the fact that the parallel program computes the correct fractal even with its varied runtime on DARWIN. However as the time varied, it did not increase past .0040 seconds for any dimensional input.

I do not notice a considerable difference between the float and double data type implementations. However due to the fact that float data types use half the memory usage as double data types we should see a faster runtime with the float implementation. The reason that we do not may be due to the fact that the parallel program is not bounded by memory usage but by computation, so by using float instead of double, which would be a memory solution, may not add much speedup to our parallel implementation.