Will Suitor (wgs231)
Lab 3 - Test Documentation

Key

Degree - The degree of the polynomial solved (x^2, x^3, etc.)
Number of Polynomials Solved - the number of runs of the program with that degree from which the averages are derived.
Average Total Iteration - the total number of guesses it took to find a polynomial that fit (per run)
Average Number of Bests - the total amount of times the program found a better polynomial than the current best
Max Single Guess Iteration Time - the maximum amount of time it took the program to generate a new random polynomial (derived from previous best) in milliseconds
Min Single Guess Iteration Time - the minimum amount of time it took the program to generate a new random polynomial (derived from previous best) in milliseconds
Average Single Guess Iteration Time - the average amount of time it took the program to generate a new random polynomial (derived from previous best) in milliseconds
Average Time Per Equation - the average amount of time it took to find a matching equation

General Testing Info: All these tests were performed on the i6 server with the number of threads specified. The maximum fitness allowed for each solution is 1 (produced by sum squared distance) and all the initial points fall within a range of -5 to 5.

*Tested on the i6 server
*For more detailed info on specifics for changes, see READ_ME.me

BASELINE

Baseline Test - 1 Thread

Degree	Number of Polynomials Solved	Average Total Iteration	Average Number of Bests	Max Single Guess Iteration Time (ms)	Min Single Guess Iteration Time (ms)	Average Single Guess Iteration Time (ms)	Average Time Per Equation
2	100	11,220.5	36.76	48	0	0.01	.447 sec
3	5	16,336.1	42.3	48	0	0.01	11.30 sec
4	1	1,466,124	742	51	0	0.01	3min 9sec

Baseline Test - 2 Threads

Degree	Number of Polynomials Solved	Average Total Iteration	Average Number of Bests	Max Single Guess Iteration Time (ms)	Min Single Guess Iteration Time (ms)	Average Single Guess Iteration Time (ms)	Average Time Per Equation
2	100	50,765.4	51.61	50	0	0.01	.422 sec
3	5	55,372.5	55.08	50	0	0.01	42.83 sec
4	1	2,088,923	634	51	0	0.01	4min 34sec

Baseline Test - 3 Threads

Degree	Number of Polynomials Solved	Average Total Iteration	Average Number of Bests	Max Single Guess Iteration Time (ms)	Min Single Guess Iteration Time (ms)	Average Single Guess Iteration Time (ms)	Average Time Per Equation
2	100	22,889.9	35.71	50	0	0.01	.289 sec
3	5	37,198.4	42.23	50	0	0.01	1min 18sec

4th degree excluded here because the run was exceedingly slow (15+ min).

Baseline Test - 4 Threads

Degree	Number of Polynomials Solved	Average Total Iteration	Average Number of Bests	Max Single Guess Iteration Time (ms)	Min Single Guess Iteration Time (ms)	Average Single Guess Iteration Time (ms)	Average Time Per Equation
2	100	13, 956.3	31.21	50	0	0.01	1.42 sec
3	5	13,733	8.29	52	0	0.01	27.59 sec

5th and some 4th degree polynomials excluded because they are remarkably slow on most runs. 5th degree equations take at least half an hour and 4th degree equations are wildly inconsistent in how fast they complete their runs. The majority of runs complete in less than a second for most of low degrees. However, when the program runs slow, it is exceedingly slow. For example, 3rd degree polynomials frequently conclude in <.5 sec but some can take up to a minute and a half.

Tests After Including Sign Change Case

1 Thread

Degree	Number of Polynomials Solved	Average Total Iteration	Average Number of Bests	Max Single Guess Iteration Time (ms)	Min Single Guess Iteration Time (ms)	Average Single Guess Iteration Time (ms)	Average Time Per Equation
2	100	20,238.2	26.81	50	0	0.01	.481 sec
3	5	7,741.56	3.84	49	0	0.01	48.07 sec

2 Threads

Degree	Number of Polynomials Solved	Average Total Iteration	Average Number of Bests	Max Single Guess Iteration Time (ms)	Min Single Guess Iteration Time (ms)	Average Single Guess Iteration Time (ms)	Average Time Per Equation
2	100	15,424.5	24.23	50	0	0.01	.346 sec
3	5	3,145.01	2.23	48	0	0.01	7.86 sec

3 Threads

Degree	Number of Polynomials Solved	Average Total Iteration	Average Number of Bests	Max Single Guess Iteration Time (ms)	Min Single Guess Iteration Time (ms)	Average Single Guess Iteration Time (ms)	Average Time Per Equation
2	100	25,530.1	25.15	50	0	0.01	.523 sec
3	5	7,590.15	2.82	50	0	0.01	6.405 sec

4 Threads

Degree	Number of Polynomials Solved	Average Total Iteration	Average Number of Bests	Max Single Guess Iteration Time (ms)	Min Single Guess Iteration Time (ms)	Average Single Guess Iteration Time (ms)	Average Time Per Equation
2	100	24,802.1	23.27	50	0	0.01	1.32 sec
3	5	47,559.4	29.15	52	0	0.01	9.62 sec

Including the sign change case increased the efficiency of the program. Before including this case, the program would pretty frequently get stuck (before a reseed) finding a best fit equation in the wrong direction. When the reseed eventually happened, it might orient itself in the correct direction or not. The negative case allows the program to correctly orient the polynomial without requiring a reseed. Of course, the ultimate goal would be to completely remove the need for a reseed but, on current performance, it is not feasible.

*Here, a reseed occurs when the program has gotten stuck and is no longer making progress. To combat this, the program clears the existing calculated coefficients and reseeds with new random coefficients.

Tests After Deleting the Random Reduction

1 Thread

Degree	Number of Polynomials Solved	Average Total Iteration	Average Number of Bests	Max Single Guess Iteration Time (ms)	Min Single Guess Iteration Time (ms)	Average Single Guess Iteration Time (ms)	Average Time Per Equation
2	100	20,662.4	26.39	48	0	0.01	.240 sec
3	5	4,920.43	2.5	48	0	0.01	26.51 sec

2 Threads

Degree	Number of Polynomials Solved	Average Total Iteration	Average Number of Bests	Max Single Guess Iteration Time (ms)	Min Single Guess Iteration Time (ms)	Average Single Guess Iteration Time (ms)	Average Time Per Equation
2	100	24,094.3	25.51	50	0	0.01	.162sec
3	5	34,786.4	30.1	48	0	0.01	1min 13.9sec

3 Threads

Degree	Number of Polynomials Solved	Average Total Iteration	Average Number of Bests	Max Single Guess Iteration Time (ms)	Min Single Guess Iteration Time (ms)	Average Single Guess Iteration Time (ms)	Average Time Per Equation
2	100	25,530.1	25.15	50	0	0.01	.101 sec
3	5	7,590.15	2.82	50	0	0.01	40.92 sec

4 Threads

Degree	Number of Polynomials Solved	Average Total Iteration	Average Number of Bests	Max Single Guess Iteration Time (ms)	Min Single Guess Iteration Time (ms)	Average Single Guess Iteration Time (ms)	Average Time Per Equation
2	100	24,802.1	23.27	50	0	0.01	.497 sec
3	5	47,559.4	29.15	52	0	0.01	.395 sec

*The calculations for time were done separately from the iteration run for this change.

This change improved performance somewhat counterintuitively. It would seem that as you approach a closer fit, reducing the amount of random change would speed up the program. However, I believe that this change improved performance largely because of the inherent inefficiency of my program. By not reducing the amount of random change, my program can more easily jump between a set of random best-fits. Instead of getting stuck with an impossible set of coefficients and then reseeding, it can jump to a markedly different set, regardless of current fitness.

Final Tests

1 Thread

Degree	Number of Polynomials Solved	Average Total Iteration	Average Number of Bests	Max Single Guess Iteration Time (ms)	Min Single Guess Iteration Time (ms)	Average Single Guess Iteration Time (ms)	Average Time Per Equation
2	100	9,993.37	25.36	49	0	0.01	.168 sec
3	5	12,931.8	28.16	49	0	0.01	17.70 sec
4	2	20,663.6	30.94	50	0	0.01	37.10 sec

4 Threads

Degree	Number of Polynomials Solved	Average Total Iteration	Average Number of Bests	Max Single Guess Iteration Time (ms)	Min Single Guess Iteration Time (ms)	Average Single Guess Iteration Time (ms)	Average Time Per Equation
2	100	5,119.1	18.35	51	0	0.01	.053 sec
3	5	118, 862.5	60.17	50	0	0.01	12.57 sec
4	2	142,570	69.5	50	0	0.01	16.37

The changes made here further increased performance. I reduced the amount of iterations required so the program can be reseeded more quickly if it gets stuck early. In general, the most effective changes seem to be the ones that increase the amount of randomness, rather than decrease it.

However, the program still struggles with 5th degree polynomials. It handles any degree below that relatively well and consistently. I would need to change a few things with the reseed again to optimize for 5th degree. The current best changes a bit more randomly for 5th degree, so what works for the lower degrees doesn't work as well for the 5th degree.