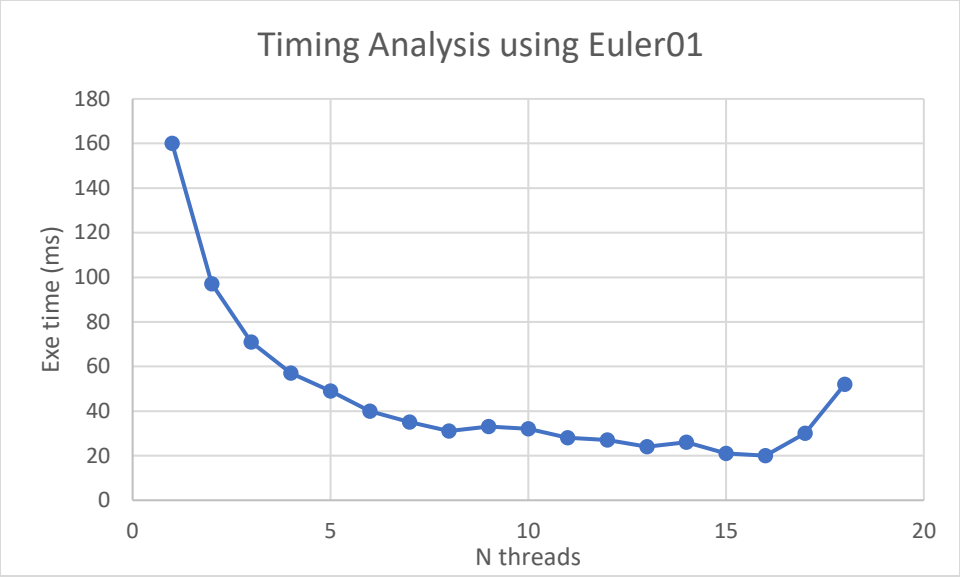

N	s	ms
1	0.16	160
2	0.097	97
3	0.071	71
4	0.057	57
5	0.049	49
6	0.04	40
7	0.035	35
8	0.031	31
9	0.033	33
10	0.032	32
11	0.028	28
12	0.027	27
13	0.024	24
14	0.026	26
15	0.021	21
16	0.02	20
17	0.03	30
18	0.052	52

The results show that initially as thread count increases the execution time decreases. The biggest performance enhancement came from N=1 to N=2 – the performance time gets close to cut in half (160 to 97 ms). After the initial gain there is a law of diminishing returns with respect to thread number. This is because of the additional overhead that it costs to add threads to the execution. After a certain point (looks like N=16) the additional threads add more overhead than the time they save, which makes parallelization past that point not worth it.

I have also tried to parallelize the calculation of the function f(x) by separately calculating exp(sin(x)) and cos(x/40), but in practice this reduced execution time. I believe this also was because of the additional

overhead. If the tasks were complex then the sections construct would be used to improve performance of calculating f(x).