Aufgabe 1:

e)

Dynamic scheduling:

	1 Thread	2 Threads	4 Threads	8 Threads	16 Threads
Serial	0.523s	0.501s	0.5014s	0.499s	0.508s
Parallel	0.526s	0.263s	0.138s	0.102s	0.091s

f)

Static scheduling:

	1 Thread	2 Threads	4 Threads	8 Threads	16 Threads
Sequenziell	0.503s	0.508s	0.507s	0.513s	0.507s
Parallel	0.634s	0.560s	0.486s	0.451s	0.249s

The static scheduling is slower for the parallelization since it can't react to load imbalances between threads. The distribution of load is done once at the beginning of the loop and can't be changed later, possibly leading to idling threads. Without static scheduling the distribution is done dynamically at runtime and can react accordingly to load imbalances, so the threads get a more evenly distributed load.

Aufgabe 2:

a) + b)

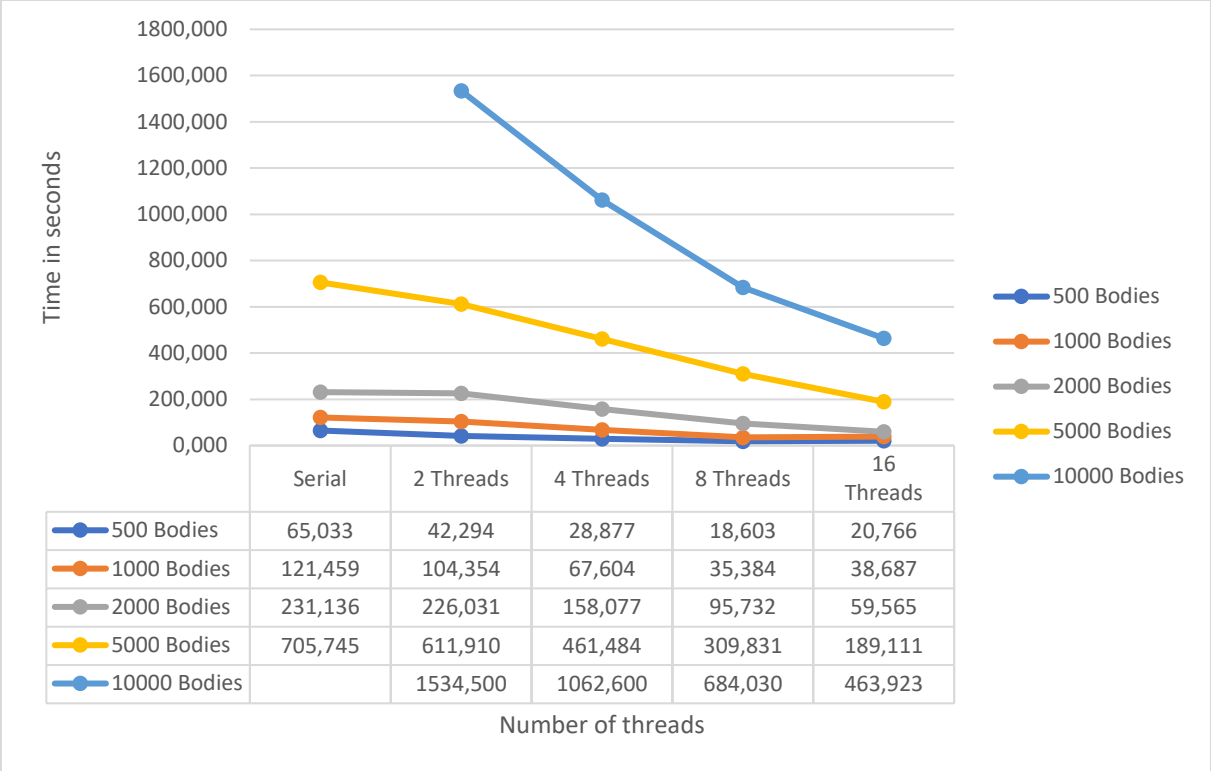
	1 Thread	2 Threads	4 Threads	8 Threads
Serial	1.814s	-	-	-
Parallel	1.946s	1.947s	1.948s	1.949s
Parallel (optimized)	2.444s	1.074s	0.586s	0.398s

In our first design for parallelization, we called daxpy_par not in parallel and our parallelization in daxpy_par itself was realized using #pragma omp task.

While trying to optimize our code, we realized our error of calling daxpy_par in serial and changed it to a parallel call, using a #pragma omp parallel-section and the #pragma omp single-construct, so the call gets made only once and by a single thread.

Aufgabe 3:

e)



	Serial	2 Threads	4 Threads	8 Threads	16 Threads
500 Bodies	65,033	42,294	28,877	18,603	20,766
1000 Bodies	121,459	104,354	67,604	35,384	38,687
2000 Bodies	231,136	226,031	158,077	95,732	59,565
5000 Bodies	705,745	611,910	461,484	309,831	189,111
10000 Bodies		1534,500	1062,600	684,030	463,923

Parallelization only makes sense after reaching 1000 bodies or more, since the runtime can be reduced by a significant amount. Everything below 1000 bodies is sped up a little, but not as effectively as with more bodies.

But as the graph shows with a higher number of bodies the saved time is increased by a large amount.

Our table is missing a value for the serial calculation of 10000 bodies, since it took over 30 minutes to complete.