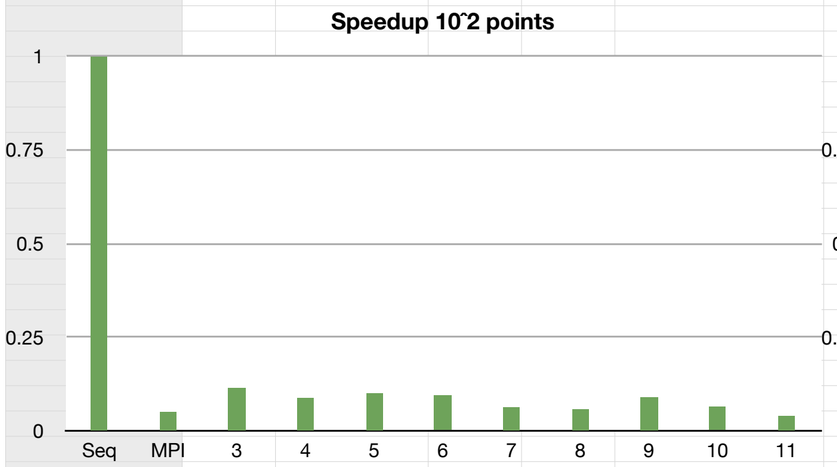
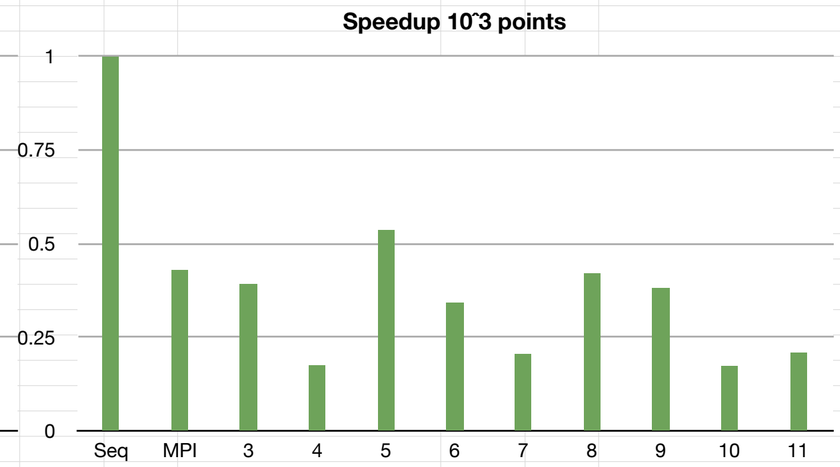
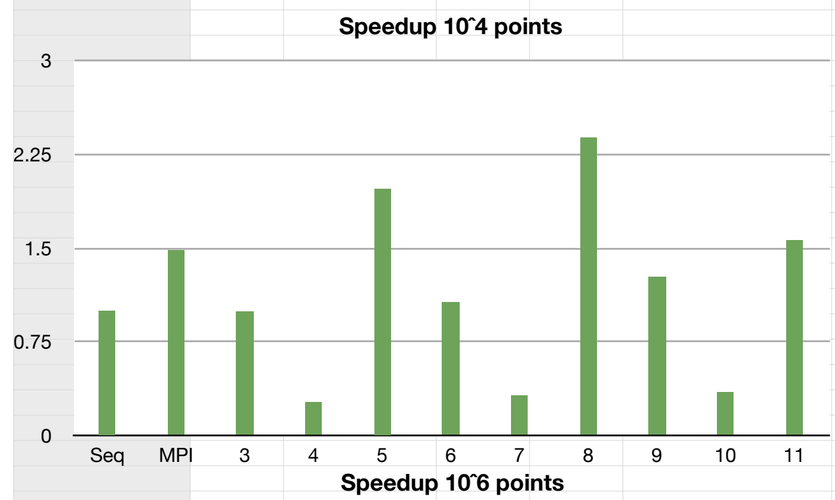
Time steps: 100 000	(1) Sequential			(3) OpenMP, t = 2, csize = 10, static			(4) OpenMP, t = 2, csize = 10, dynamic			(5) OpenMP, t = 2, csize = 10, guided		
Point calculated	**Time**	**Time Normalized**	**Speedup**	**Time**	**Time Normalized**	**Speedup**	**Time**	**Time Normalized**	**Speedup**	**Time**	**Time Normalized**	**Speedup**
100	0.0270453	2.70E-09	1	0.236302	2.36E-08	1E-01	0.30602	3.06E-08	9E-02	0.26977	2.70E-08	1E-01
1000	0.26529	2.65E-09	1	0.676485	6.76E-09	4E-01	1.509	1.51E-08	2E-01	4.49445	4.94E-09	5E-01
10000	3.13025	3.13E-09	1	3.15133	3.15E-09	1E+00	11.7499	1.17E-08	3E-01	1.58468	1.58E-09	2E+00
100000	92.9069	6.59E-08	1	62.1771	6.22E-09	1E+01	155.035	1.55E-08	4E+00	13.6929	1.37E-09	5E+01
1000000	428.36	3.52E-07	1	335.563	3.36E-09	1E+02	1129.31	1.13E-08	3E+01	561.261	5.61E-09	6E+01

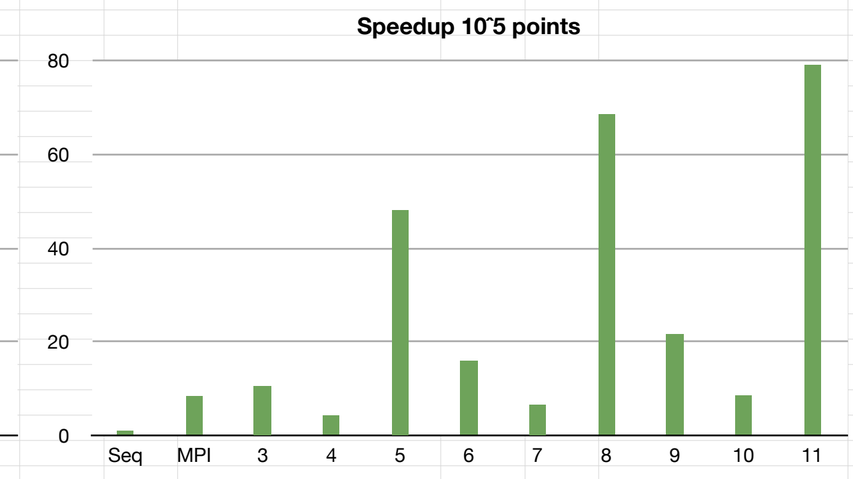
Time steps: 100 000	(2) 1 MPI proc, 8 nodes			(6) OpenMP, t = 4, csize = 10, static			(7) OpenMP, t = 4, csize = 10, dynamic			(8) OpenMP, t = 4, csize = 10, guided		
Point calculated	**Time**	**Time Normalized**	**Speedup**	**Time**	**Time Normalized**	**Speedup**	**Time**	**Time Normalized**	**Speedup**	**Time**	**Time Normalized**	**Speedup**
100	0.523705	5.24E-08	5E-02	0.285995	2.86E-08	9E-02	0.42356	4.24E-08	6E-02	0.46479	4.65E-08	6E-02
1000	0.617138	6.17E-09	4E-01	0.77435	7.74E-09	3E-01	1.28829	1.29E-08	2E-01	0.63095	6.31E-09	4E-01
10000	2.11306	2.11E-09	1E+00	2.93565	2.94E-09	1E+00	8.81979	9.82E-09	3E-01	1.3138	1.31E-09	2E+00
100000	78.5648	7.86E-09	8E+00	41.5324	4.15E-09	2E+01	99.8018	9.98E-09	7E+00	9.61923	9.62E-10	7E+01
1000000	447.843	4.48E-09	8E+01	315.254	3.15E-09	1E+02	888.185	8.88E-09	4E+01	519.058	5.19E-09	7E+01

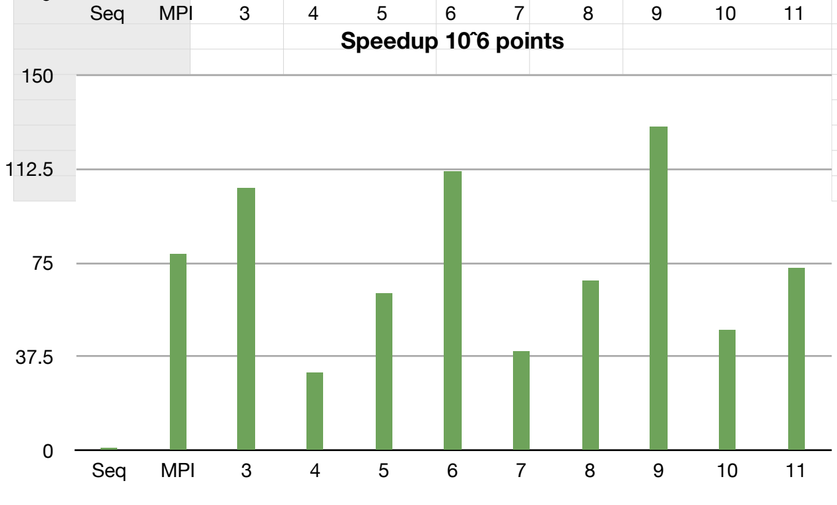
Time steps: 100 000	(9) OpenMP, t = 8, csize = 10, static			(10) OpenMP, t = 8, csize = 10, dynamic			(11) OpenMP, t = 8, csize = 10, guided		
Point calculated	**Time**	**Time Normalized**	**Speedup**	**Time**	**Time Normalized**	**Speedup**	**Time**	**Time Normalized**	**Speedup**
100	0.299236	2.99E-08	9E-02	0.41965	4.20E-08	6E-02	0.66659	6.67E-08	4E-02
1000	0.694184	6.94E-09	4E-01	1.52	1.52E-08	2E-01	1.26775	1.27E-08	2E-01
10000	2.46634	2.47E-09	1E+00	9.07272	9.07E-09	3E-01	2.00645	2.01E-09	2E+00
100000	30.5617	3.06E-09	2E+01	77.738	7.77E-09	8E+00	8.3387	8.34E-10	8E+01
1000000	271.94	2.72E-09	1.2956608	729.637	7.30E-09	5E+01	482.889	4.83E-09	7E+01











The use of openMP on calculations containing less than 10^4 points slows down the execution. When calculating 10^4 points *sometimes* we can observe a speedup. The highest speedup (at 10^4) is achieved with 4 threads. Creating more threads does the code not let run any faster because of the overhead of the threads.

However, when increasing the problem to 10^5 points we gain time by creating extra threads (see graph 'Speedup 10^5 points').

In some cases the static scheduler achieves the highest speedups, in other cases the guided scheduler. We expect this is because the fact that the work is more equally divided between the CPU's dos not compensate for the time needed to create extra chunks.