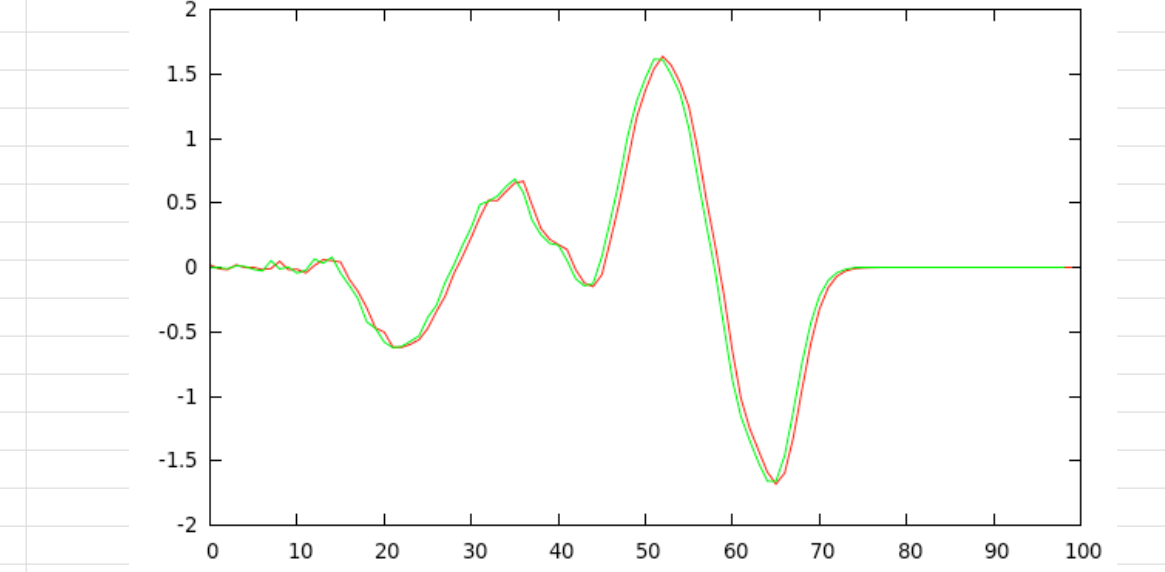
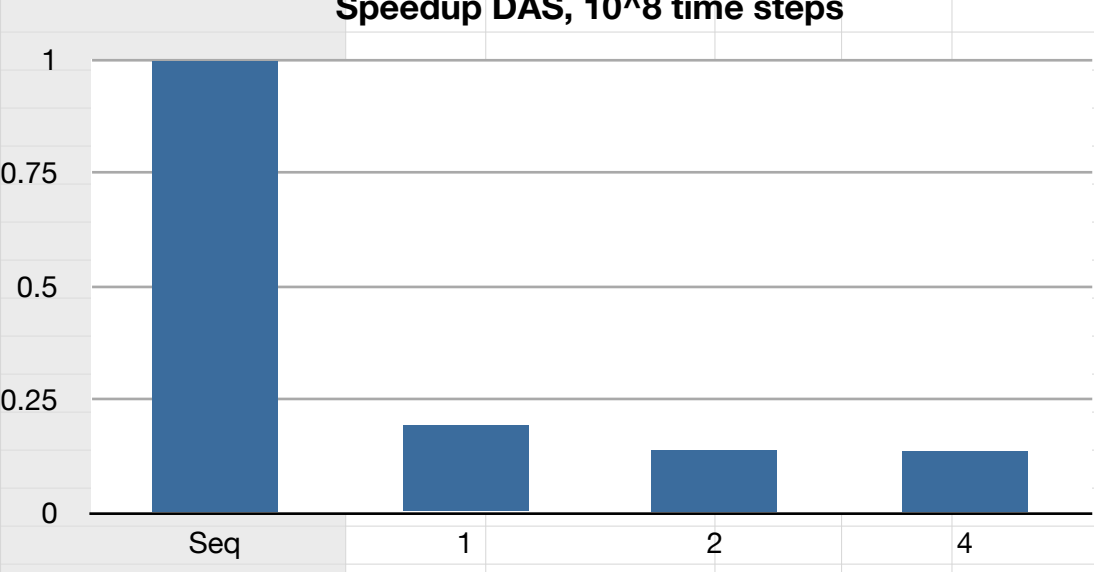
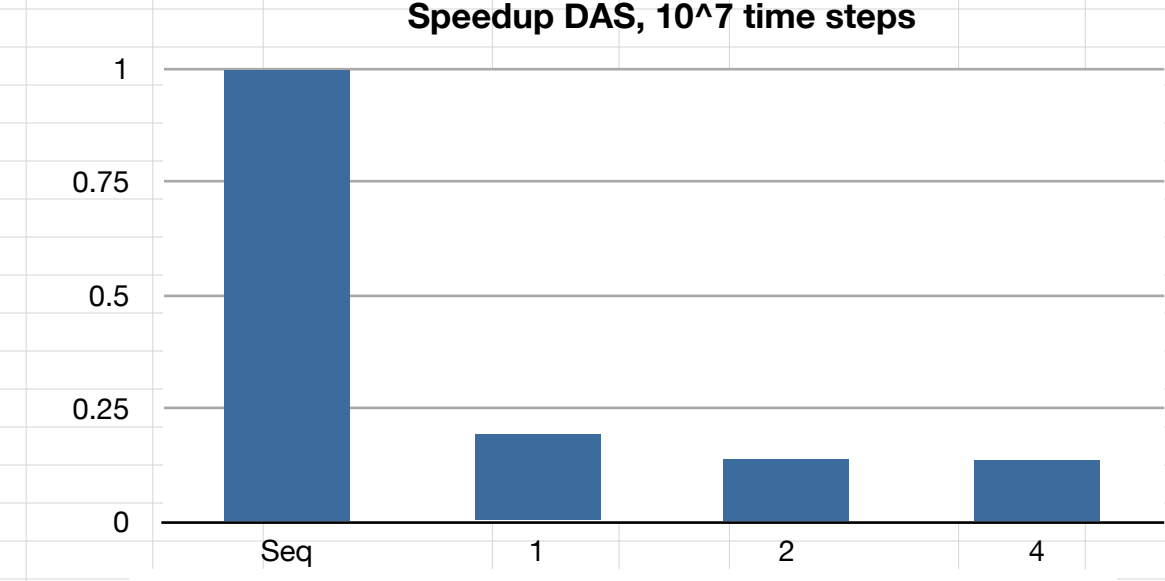
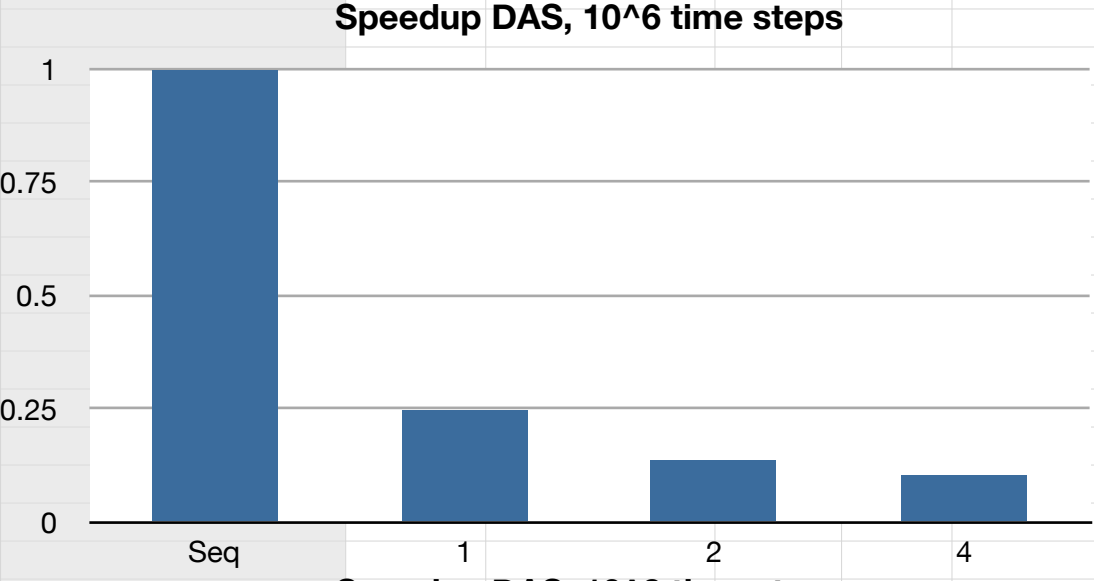
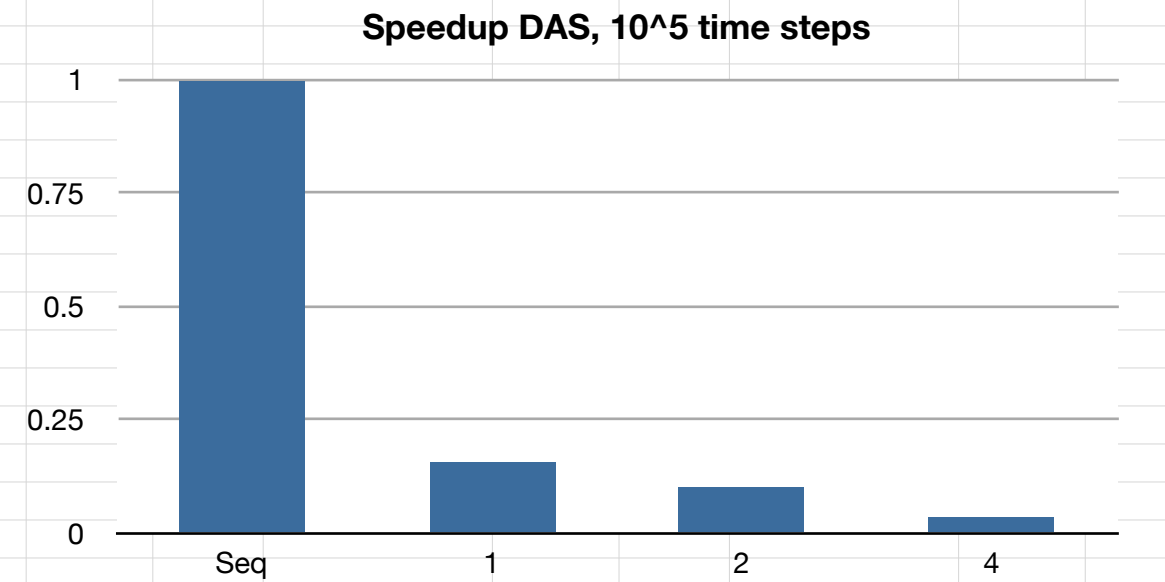
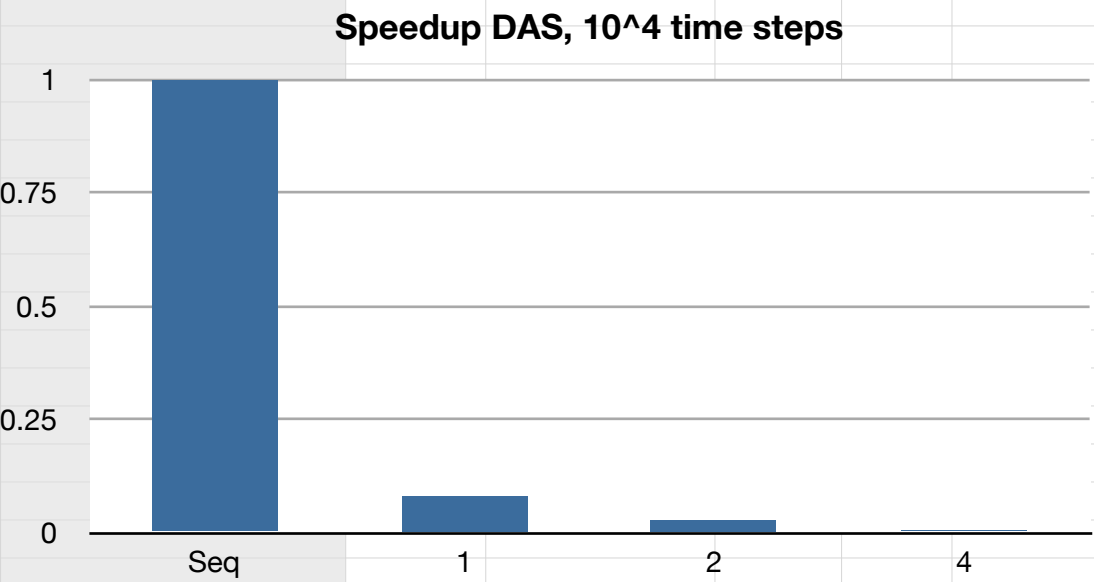


DAS															
Aantal punten: 200	(1) Sequential			(2) 8 MPI procs, 1 node			(3) 1 MPI proc, 8 nodes			(4) 8 MPI procs, 8 nodes					
Stappen in tijd	Time	Time Normalized	Speedup	Time	Time Normalized	Speedup	Time	Time Normalized	Speedup	Time	Time Normalized	Speedup			
10000	0.0053587	2.68E-09	1	0.0664	3.32E-08	8E-02	0.19631	9.82E-08	3E-02	1.17411	5.87E-07	5E-03			
100000	0.0533379	2.67E-09	1	0.34342	1.72E-08	2E-01	0.52877	2.64E-08	1E-01	1.52251	7.61E-08	4E-02			
1000000	0.533373	2.67E-09	1	2.16295	1.08E-08	2E-01	3.90889	1.95E-08	1E-01	5.13418	2.57E-08	1E-01			
10000000	5.33127	2.67E-09	1	2.76925	1.38E-08	2E-01	38.3325	1.92E-08	1E-01	39.26	1.96E-08	1E-01			
100000000	53.3313	2.67E-09	1	24.2042	1.21E-08	2E-01	414.377	2.07E-08	1E-01	376.039	1.19E-08	2E-01			



In these graphs you can see that using MPI in these calculations is actually slowing down the execution. Due to a minor bug in our code we couldn't run simulations with > 200 points. We varied the number of time steps to increase the calculation time. However, this approach uses much more MPI messages (because there are more 'halo cells' to be passed through).

We have found the cause of the bug (bufferoverflow at line #216), however we could not fix it before the deadline. When increasing the number of (independent) calculations we expect a *much* higher performance because there will be less message passing.

Also, to check the validity of the results we compared the results of assignment1_1 with current results. They almost match perfectly.

UPDATE 21.30u: We also ran the tests with a variable number of points (at a fixed number of time steps). Because there is (relatively) much less messaging passing needed the MPI program performs MUCH better.

Time steps: 100 000	(1) Sequential			(2) 8 MPI procs, 1 node			(3) 1 MPI proc, 8 nodes			(4) 8 MPI procs, 8 nodes					
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	0.58823	5.88E-08	5E-02	0.52371	5.24E-08	5E-02	1.50493	1.50E-07	2E-02			
1000	0.26529	2.65E-09	1	0.33138	3.31E-09	8E-01	0.61714	6.17E-09	4E-01	1.65283	1.65E-08	2E-01			
10000	3.13025	3.13E-09	1	2.78143	2.78E-09	1E+00	2.11306	2.11E-09	1E+00	2.91886	2.92E-09	1E+00			
100000	92.9069	6.59E-08	1	80.1024	8.01E-09	8E+00	78.5648	7.86E-09	8E+00	48.4409	4.84E-09	1E+01			
1000000	428.36	3.52E-07	1	584.96	5.85E-09	6E+01	447.843	4.48E-09	8E+01	503.222	5.03E-09	7E+01			

