MIKE 21 Flow Model FM

Parallelisation using GPU

Benchmarking report

DHI headquarters
Agern Allé 5
DK-2970 Hørsholm
Denmark

+45 4516 9200 Telephone
+45 4516 9333 Support
+45 4516 9292 Telefax

mike@dhigroup.com
www.mikepoweredbydhi.com

PLEASE NOTE

CONTENTS

MIKE 21 Flow Model FM
Parallelisation using GPU
Benchmarking report

1 Vision and Scope

A set of well-defined test cases for the GPU version of the MIKE 21 Flow Model FM, has been established. The test-suite is used for testing the performance across platforms with different graphics cards. It is essential that it is possible to run the simulation with different spatial resolutions to be able to evaluate the scalability of the parallelisation. The main focus is to benchmark the GPU parallelisation of the flexible mesh modelling system. For comparison, simulations have also been performed using the CPU version of MIKE 21 Flow Model FM.

2 Methodology

2.1 GPU Parallelisation

The GPU computing approach uses the computer's graphics card to perform the computational intensive calculations. This approach is based on CUDA by NVIDIA and can be executed on NVIDIA graphics cards with Compute Capability 3.0 or higher.

Depending on the available hardware it is possible to launch a simulation using a single or multiple GPUs. The multiple GPU approach is based on the domain decomposition concept, where the communication between the processors is done using MPI (Message Passing Interface).

Currently, only the computational intensive hydrodynamic calculations are performed on the GPU. The additional calculations are for each sub-domain in the domain decomposition performed locally on the CPU and these calculations are further parallelised based on the shared memory approach, OpenMP.

As default, the program uses one MPI process per GPU, but it is possible to assign more processes to the same GPU. In this way simulations, where the hydrodynamic calculations are less time consuming than the calculations performed in the other modules, will benefit from the MPI parallelisation.

2.2 Hardware

The benchmarks have been performed using the following hardware platforms and GPUs:

Table 2.1 Hardware platforms used for benchmarking

	Computer	Processor	Memory	Operating system	GPUs
1	Microsoft Azure (Instance NC12)	Intel®Xeon® E5-2690 v3 (12 cores, 2.60 GHz)	112 GB	Windows 10 Pro, 64-bit	1 x Tesla K80 (dual card)
2	Microsoft Azure (Instance NC12 v2)	Intel®Xeon® E5-2690 v4 (12 cores, 2.60 GHz)	224 GB	Windows 10 Pro, 64-bit	2 x Tesla P100
3	Microsoft Azure (Instance NC12 v3)	Intel®Xeon® E5-2690 v4 (12 cores, 2.60 GHz)	224 GB	Windows 10 Pro, 64-bit	2 x Tesla V100

Table 2.2 GPU specifications

GPU	Compute Capability	Number of CUDA cores	Memory (GB)	Bandwidth (GB/s)	GPU Clock (MHz)	Single/Double precision floating point performance
Tesla K80 (dual card)	3.7	2 x 2496	2 x 12	2 x 240	562	2 x 2.8 Tflops / 2 x 0.94 Tflops
Tesla P100	6.0	3584	16	732	1189	8.0 Tflops / 4.0 Tflops
Tesla V100	7.0	5120	16	897	1245	14.0 Tflops / 7.0 Tflops

2.3 Software

All benchmarks have been performed using the MIKE 2019 Release. The CUDA 9.2 library is used in the MIKE 2019 Release. In the present benchmark the NVIDIA graphics driver 398.75 has been used for hardware platform 1, 2 and 3, and the graphics driver is running in TCC mode, with ECC enabled.

2.4 Performance of the GPU Parallelisation

The parallel performance of the GPU version of MIKE 21 Flow Model FM compared to the CPU version is illustrated by measuring the speedup factor, $t_{CPU(m)}/t_{GPU(n)}$. Here $t_{CPU(m)}$ is the elapsed time using the existing CPU version (m subdomains and 1 core/thread) and $t_{GPU(n)}$ is the elapsed time using the GPU version (n subdomains and 1 core/thread for the CPU part of the calculation). The elapsed time is the total elapsed time (excluding pre- and post-processing). The performance metric is highly dependent on not only the GPU hardware but also the CPU hardware. The parallel performance of multi-GPUs is illustrated by measuring the speedup factor, $t_{GPU(1)}/t_{GPU(n)}$. In the simulations one MPI process is used per GPU. For the GPU simulations the number of threads per block on the GPU is 128.

Per default the calculations performed on the GPU are done in double precision. However, since some GPUs have a significantly lower double precision floating point performance than single precision floating point performance, it is possible to force the calculations on the GPU to be performed in single precision. For this reason, the benchmarking has been done using both single and double precision calculations.
Be aware that using single precision calculations will affect the accuracy of the simulation results, since single precision calculations are less accurate than double precision calculations.

The ratio between the specified theoretical single and double precision floating point performance is not equal to the actual measured performance ratio between single and double precision. This becomes evident when comparing the values in the last column of Table 2.2, where the theoretical single precision performance is a factor 2-3 higher than

the theoretical double precision performance, to the actual measured difference between single and double precision as presented in the benchmarking below.

3 Description of Test Cases

3.1 Mediterranean Sea

This test case has been established for benchmarking of the MIKE 21 Flow Model FM.

3.1.1 Description

In the Western parts of the Mediterranean Sea tides are dominated by the Atlantic tides entering through the Strait of Gibraltar, while the tides in the Eastern parts are dominated by astronomical tides, forced directly by the Earth-Moon-Sun interaction.

3.1.2 Setup

The bathymetry is shown in Figure 3.1. Simulations are performed using five meshes with different resolution (see Table 3.1). The meshes are generated specifying the value for the maximum area of 0.04, 0.005, 0.00125, 0.0003125 and 0.000078125 degree^2, respectively. The simulation period for the benchmarks covers 2 days starting 1 January 2004 for the simulations using mesh A, B and C. The simulation period is reduced to 6 hours for the simulations using mesh D and 3 hours for mesh E.

At the Atlantic boundary a time varying level boundary is applied. The tidal elevation data is based on global tidal analysis (Andersen, 1995).

For the bed resistance the Manning formulation is used with a Manning number of 32. For the eddy viscosity the Smagorinsky formulation is used with a Smagorinsky factor of 1.5. Tidal potential is applied with 11 components (default values).

The shallow water equations are solved using both the the higher-order scheme in time and space.

Table 3.1 Computational mesh for the Mediterranean Sea case

Mesh	Element shape	Elements	Nodes	Max. area Degree^2
Mesh A	Triangular	11287	6283	0.04
Mesh B	Triangular	80968	41825	0.005
Mesh C	Triangular	323029	164161	0.00125
Mesh D	Triangular	1292116	651375	0.0003125
Mesh E	Triangular	5156238	2588665	0.000078125

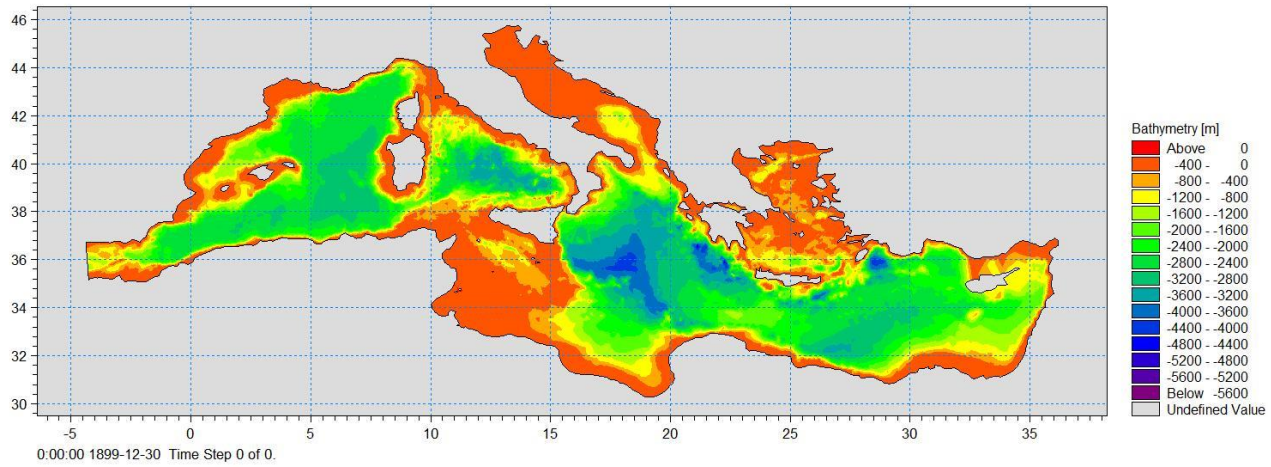

Figure 3.1 Bathymetry for the Mediterranean Sea case

The averaged time step for the simulations using Mesh A, B, C, D and E is 17.65s, 5.61s, 2.86s, 1.43s and 0.69s, respectively, when using the higher-order scheme in time and space.

3.2 EA2D Test 8A

This test is Test 8A in the benchmarks test developed during the Joint Defra/Environment Agency research programme. This tests the package's capability to simulate shallow inundation originating from a point source and from rainfall applied directly to the model grid, at a relatively high resolution. This test case has been established for benchmarking of the MIKE 21 Flow model FM.

3.2.1 Description

The modelled area is approximately 0.4 km by 0.96 km and covers entirely the DEM provided and shown in Figure 3.2. Ground elevations span a range of ~21m to ~37m.

The flood is assumed to arise from two sources:

- a uniformly distributed rainfall event illustrated by the hyetograph in Figure 3.3. This is applied to the modelled area only (the rest of the catchment is ignored).
- a point source at the location (264896, 664747) (Map projection: British national grid), and illustrated by the inflow time series in Figure 3.4. (This may for example be assumed to arise from a surcharging culvert.)

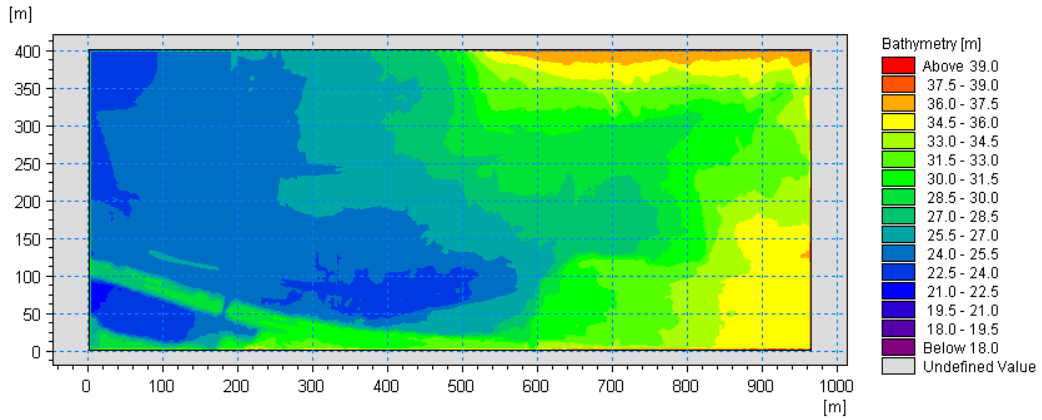

Figure 3.2 Bathymetry for the EA2D Test8A case

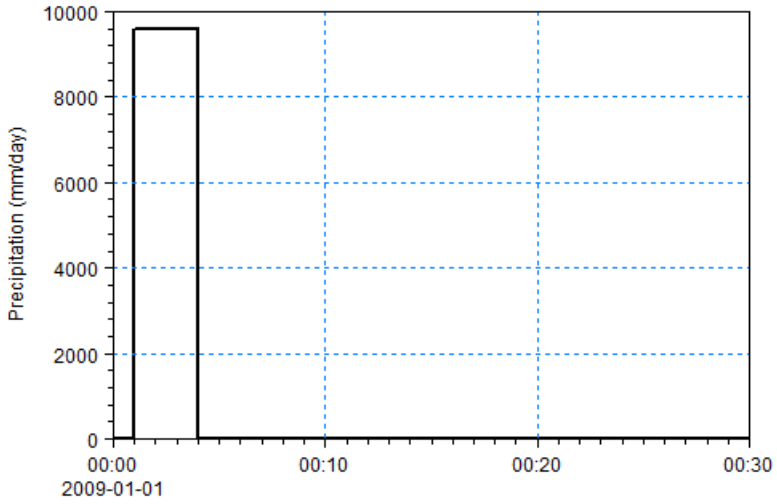

Figure 3.3 Rainfall

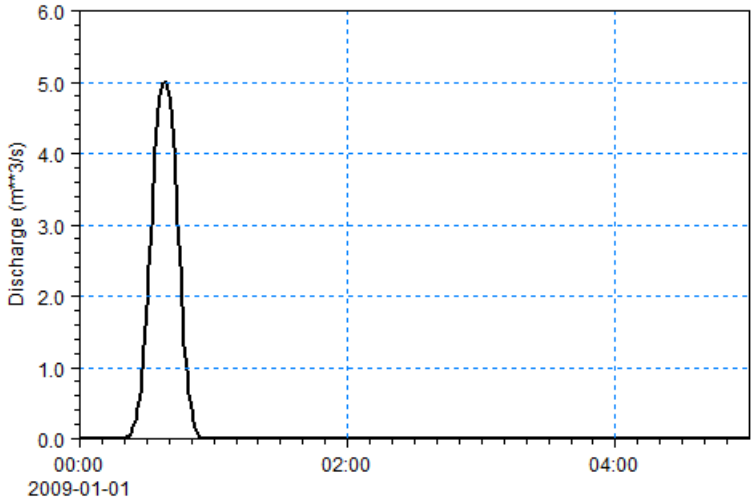

Figure 3.4 Discharge from source

DEM is a 0.5m resolution Digital Terrain Model (no vegetation or buildings) created from LiDAR data collected on 13th August 2009 and provided by the Environment Agency (http://www.geomatics-group.co.uk). Model grid resolution should be 2m (or ~97000 nodes in the 0.388 km^2 area modelled).

All buildings at the real location (Cockenzie Street and surrounding streets in Glasgow, UK) are ignored and the modelling is carried out using the “bare-earth” DEM provided.

A land-cover dependent roughness value is applied, with 2 categories: 1) Roads and pavements; 2) Any other land cover type. Manning’s n = 0.02 is applied for roads and pavements n = 0.05 everywhere else.

All boundaries in the model area are closed (no flow) and the initial condition is dry bed. The model is run until time T = 5 hours to allow the flood to settle in the lower parts of the modelled domain.

3.2.2 Setup

Simulations are performed using four meshes with different resolution (see Table 3.2). The four meshes uses regular quadrilateral elements with grid spacing 2m, 1m, 0.5m and 0.25m, respectively. Mesh A corresponds to the original mesh used in the EA2D test, and the additional meshes are obtained by refining this mesh.

Table 3.2 Computational mesh for the EA2D Test 8A case

Mesh	Element shape	Elements	Nodes	Grid spacing metres
Mesh A	Quadrilateral	95719	96400	2
Mesh B	Quadrilateral	384237	385600	1
Mesh C	Quadrilateral	1539673	1542400	0.5
Mesh D	Quadrilateral	6164145	6169600	0.25

The shallow water equations are solved using the first-order scheme in time and space.

The averaged time step for the simulation using Mesh A, B, C and D is 0.22s, 0.10s, 0.5s and 0.027s, respectively.

4 Benchmarking using Tesla K80

These tests have been performed using a GPU instance on Microsoft Azure specified as hardware platform 1 in Table 2.1. The simulations have been performed using a Tesla K80 card (1 and 2 subdomains and 1 thread). For comparison, simulations have also been performed without GPU acceleration (1 and 12 subdomains and 1 thread).

4.1 Mediterranean Sea

Table 4.1 Computational time, $t_{GPU(n)}$,using GPU acceleration (1 and 2 subdomains and 1 thread) and speedup factor ,$t_{GPU(1)}/t_{GPU(n)}$. The simulations are carried out using single precision (SP) and double precision (DP) and using higher-order scheme in time and space

Mesh	No. of GPUs n	SP		DP	
		Time (s) $t_{GPU(n)}$	Speedup Factor $t_{GPU(1)}/t_{GPU(n)}$	Time (s) $t_{GPU(n)}$	Speedup Factor $t_{GPU(1)}/t_{GPU(n)}$
Mesh A	1	11.84	1.00	13.09	1.00
	2	16.64	0.71	16.31	0.80
Mesh B	1	112.36	1.00	150.87	1.00
	2	80.57	1.39	98.14	1.53
Mesh C	1	722.81	1.00	1027.47	1.00
	2	421.24	1.71	574.79	1.78
Mesh D	1	631.08	1.00	944.53	1.00
	2	337.44	1.87	497.37	1.89
Mesh E	1	2670.52	1.00	4984.95	1.00
	2	1411.39	1.89	2098.36	2.37

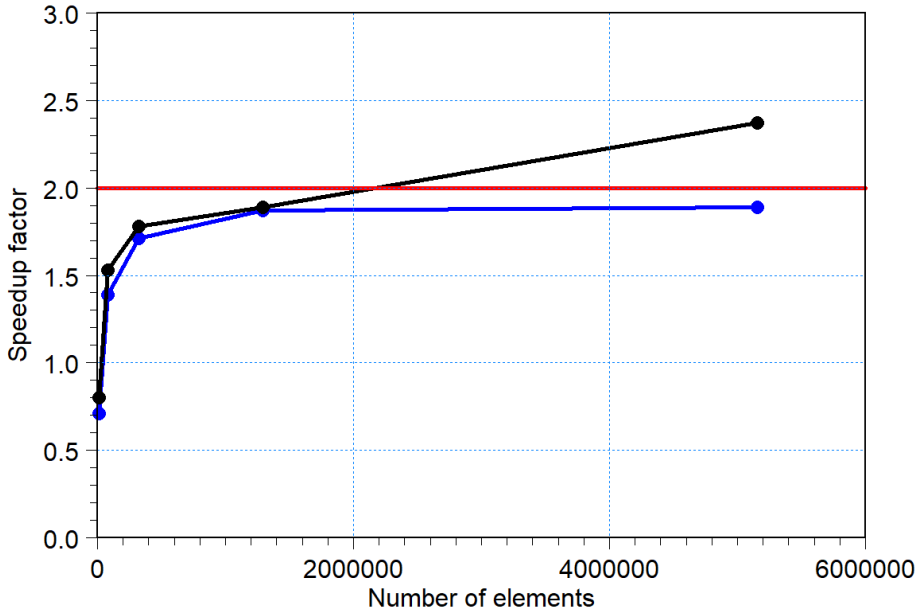

Figure 4.1 Speedup factor, $t_{GPU(1)}/t_{GPU(n)}$, for two GPUs relative to a single GPU using higher-order scheme in time and space. Blue line: single precision; Black line: double precision; Red line: Ideal speedup factor

Table 4.2 Computational time, $t_{CPU(n)}$, using no GPU acceleration (1 and 12 subdomains and 1 thread) and speedup factor, $t_{CPU(1)}/t_{CPU(n)}$. The simulations are carried out higher-order scheme in time and space

Mesh	No. of domains n	Time (s) $t_{CPU(n)}$	Speedup Factor $t_{CPU(1)}/t_{CPU(n)}$
Mesh A	1	234.69	1.00
	12	22.73	10.32
Mesh B	1	5760.86	1.00
	12	531.55	10.83
Mesh C	1	44922.76	1.00
	12	4485.60	10.01
Mesh D	1	44895.63	1.00
	12	4464.63	10.05

Table 4.3 Speedup factors, $t_{CPU(1)}/t_{GPU(n)}$ and $t_{CPU(12)}/t_{GPU(n)}$. The simulations are carried out using single precision (SP) and double precision (DP) and using higher-order scheme in time and space

Mesh	No. of GPUs n	Speedup Factor $t_{CPU(1)}/t_{GPU(n)}$		Speedup Factor $t_{CPU(12)}/t_{GPU(n)}$	
		SP	DP	SP	DP
Mesh A	1	19.82	17.92	1.91	1.73
	2	14.10	14.38	1.36	1.39
Mesh B	1	51.27	38.18	4.73	3.52
	2	71.50	58.70	6.59	5.41
Mesh C	1	62.15	43.72	6.20	4.36
	2	106.64	78.15	10.64	7.80
Mesh D	1	71.14	47.53	7.07	4.72
	2	133.04	90.26	13.23	8.97

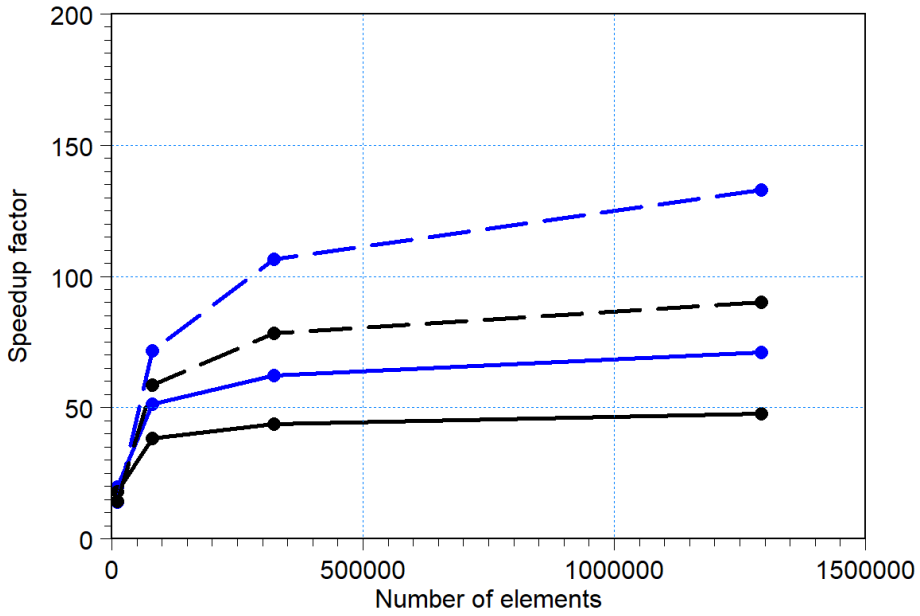

Figure 4.2 Speedup factor, $t_{CPU(1)}/t_{GPU(n)}$, using only one or both of the GPUs on the Tesla K80 card using higher-order scheme. Blue line: single precision; Black line: double precision. Solid line: 1 GPU; Dash line: 2 GPUs

4.2 EA2D Test 8A

Table 4.4 Computational time, $t_{GPU(n)}$, using GPU acceleration (1 and 2 subdomains and 1 thread) and speedup factor ,$t_{GPU(1)}/t_{GPU(n)}$. The simulations are carried out using single precision (SP) and double precision (DP) and using first-order scheme in time and space

Mesh	No. of GPUs n	SP		DP	
		Time (s) $t_{GPU(n)}$	Speedup Factor $t_{GPU(1)}/t_{GPU(n)}$	Time (s) $t_{GPU(n)}$	Speedup Factor $t_{GPU(1)}/t_{GPU(n)}$
Mesh A	1	126.73	1.00	155.61	1.00
	2	121.14	1.04	122.15	1.27
Mesh B	1	633.62	1.00	821.72	1.00
	2	417.69	1.51	544.82	1.50
Mesh C	1	4040.28	1.00	5365.66	1.00
	2	2300.37	1.75	3101.47	1.73

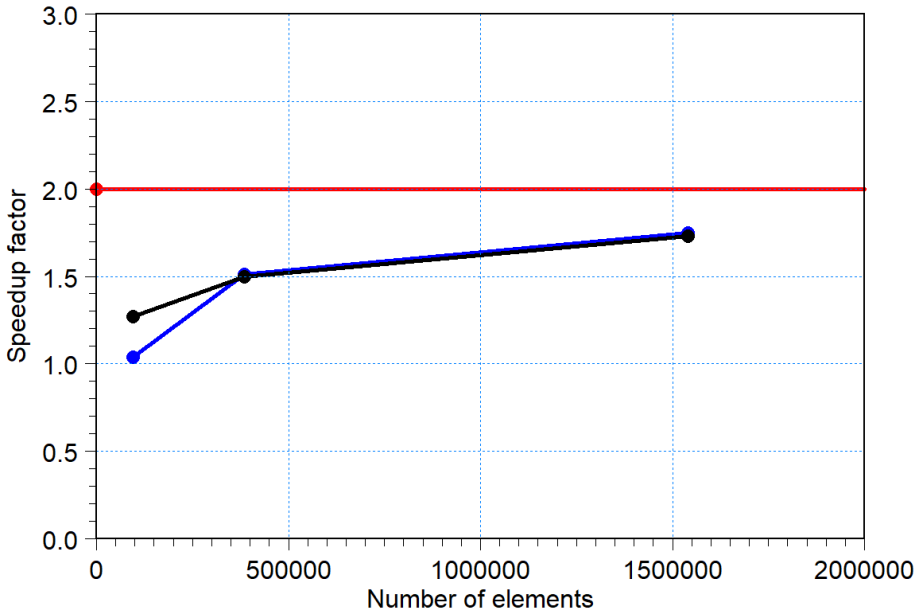

Figure 4.3 Speedup factor, $t_{GPU(1)}/t_{GPU(n)}$, for two GPUs relative to a single GPU. Blue line: single precision; Black line: double precision; Red line: Ideal speedup factor

Table 4.5 Computational time, $t_{CPU(n)}$, using no GPU acceleration (1 and 12 domains and 1 thread) and speedup factor, $t_{CPU(1)}/t_{CPU(n)}$. The simulations are carried out using first-order scheme in time and space

Mesh	No. of domains n	Time (s) $t_{CPU(n)}$	Speedup Factor $t_{CPU(1)}/t_{CPU(n)}$
Mesh A	1	1810.65	1.00
	12	299.31	6.04
Mesh B	1	14702.13	1.00
	12	2406.84	6.10
Mesh C	1	115122.3	1.00
	12	19390.88	5.93

Table 4.6 Speedup factors, $t_{CPU(1)}/t_{GPU(n)}$ and $t_{CPU(12)}/t_{GPU(n)}$. Simulations are carried out using single precision (SP) and double precision (DP) and using first-order scheme in time and space

Mesh	No. of GPUs n	Speedup Factor $t_{CPU(1)}/t_{GPU(n)}$		Speedup Factor $t_{CPU(12)}/t_{GPU(n)}$	
		SP	DP	SP	DP
Mesh A	1	14.28	11.63	2.36	1.92
	2	14.94	14.82	2.47	2.45
Mesh B	1	23.20	17.89	3.79	2.92
	2	35.19	26.98	5.76	4.41
Mesh C	1	28.49	21.45	4.79	3.61
	2	50.04	37.11	8.42	6.25

5 Benchmarking using Tesla P100

These tests have been performed using a GPU instance on Microsoft Azure specified as hardware platform 2 in Table 2.1. The simulations have been performed using one and two Tesla P100 cards (1 and 2 subdomains and 1 thread).

5.1 Mediterranean Sea

Table 5.1 Computational time, $t_{GPU(n)}$,using GPU acceleration (1 and 2 subdomains and 1 thread) and speedup factor ,$t_{GPU(1)}/t_{GPU(n)}$. The simulations are carried out using single precision (SP) and double precision (DP) and using higher-order scheme in time and space

Mesh	No. of GPUs n	SP		DP	
		Time (s) $t_{GPU(n)}$	Speedup Factor $t_{GPU(1)}/t_{GPU(n)}$	Time (s) $t_{GPU(n)}$	Speedup Factor $t_{GPU(1)}/t_{GPU(n)}$
Mesh A	1	7.30	1.00	7.81	1.00
	2	9.27	0.78	9.31	0.83
Mesh B	1	46.33	1.00	58.21	1.00
	2	42.79	1.08	47.88	1.21
Mesh C	1	250.99	1.00	347.77	1.00
	2	163.73	1.53	220.03	1.58
Mesh D	1	202.17	1.00	314.05	1.00
	2	112.12	1.80	162.56	1.93
Mesh E	1	836.14	1.00	1296.09	1.00
	2	425.03	1.96	676.13	1.91

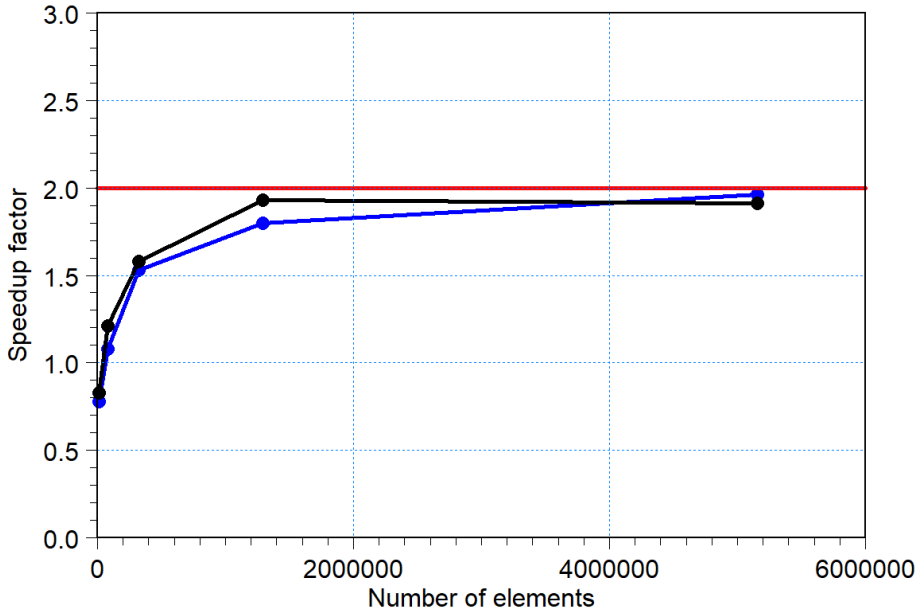

Figure 5.1 Speedup factor, $t_{GPU(1)}/t_{GPU(n)}$, for two GPUs relative to a single GPU using higher-order scheme in time and space. Blue line: single precision; Black line: double precision; Red line: Ideal speedup factor

Table 5.2 Speedup factors, $t_{CPU(1)}/t_{GPU(n)}$ and $t_{CPU(12)}/t_{GPU(n)}$. The simulations are carried out using single precision (SP) and double precision (DP) and using higher-order scheme in time and space. The timings $t_{CPU(1)}$ and $t_{CPU(12)}$ are the timings from hardware platform 1 as in Table 4.2.

Mesh	No. of GPUs n	Speedup Factor $t_{CPU(1)}/t_{GPU(n)}$		Speedup Factor $t_{CPU(12)}/t_{GPU(n)}$	
		SP	DP	SP	DP
Mesh A	1	32.14	30.04	3.11	2.91
	2	25.31	25.20	2.45	2.44
Mesh B	1	124.34	98.96	11.47	9.13
	2	134.63	120.31	12.42	11.10
Mesh C	1	178.98	129.17	17.87	12.89
	2	274.37	204.16	27.39	20.38
Mesh D	1	222.06	142.95	22.08	14.21
	2	400.42	276.17	39.82	27.46

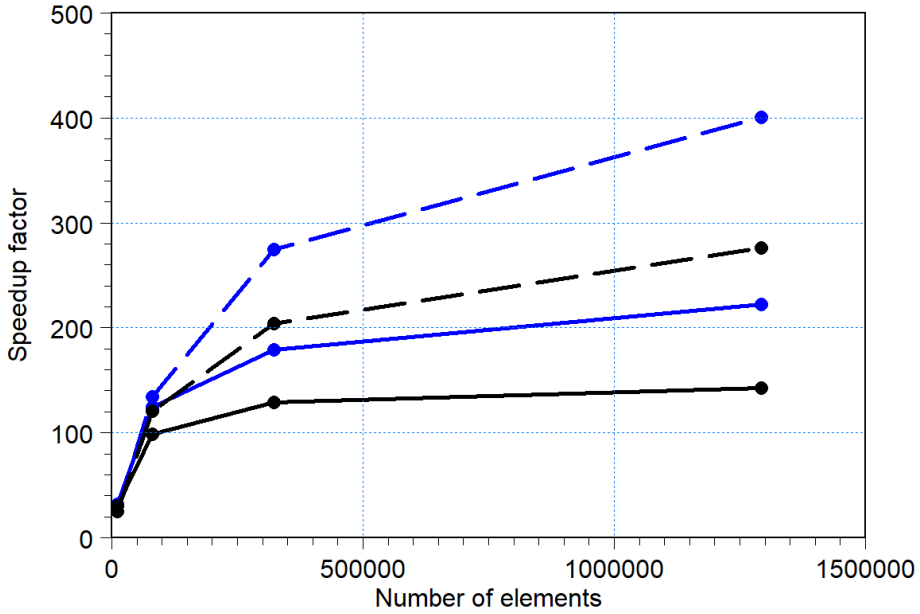

Figure 5.2 Speedup factor, $t_{CPU(1)}/t_{GPU(n)}$, for one and two Tesla P100 cards using higher-order scheme compared to the timings in Table 4.2. Blue line: single precision; Black line: double precision. Solid line: 1 GPU; Dash line: 2 GPUs

5.2 EA2D Test 8A

Table 5.3 Computational time, $t_{GPU(n)}$, using GPU acceleration (1 and 2 subdomains and 1 thread) and speedup factor ,$t_{GPU(1)}/t_{GPU(n)}$. The simulations are carried out using single precision (SP) and double precision (DP) and using first-order scheme in time and space

Mesh	No. of GPUs n	SP		DP	
		Time (s) $t_{GPU(n)}$	Speedup Factor $t_{GPU(1)}/t_{GPU(n)}$	Time (s) $t_{GPU(n)}$	Speedup Factor $t_{GPU(1)}/t_{GPU(n)}$
Mesh A	1	73.58	1.00	80.29	1.00
	2	73.97	0.99	75.04	1.06
Mesh B	1	294.10	1.00	345.06	1.00
	2	233.53	1.25	257.12	1.34
Mesh C	1	1595.85	1.00	2007.26	1.00
	2	991.51	1.60	1232.83	1.62

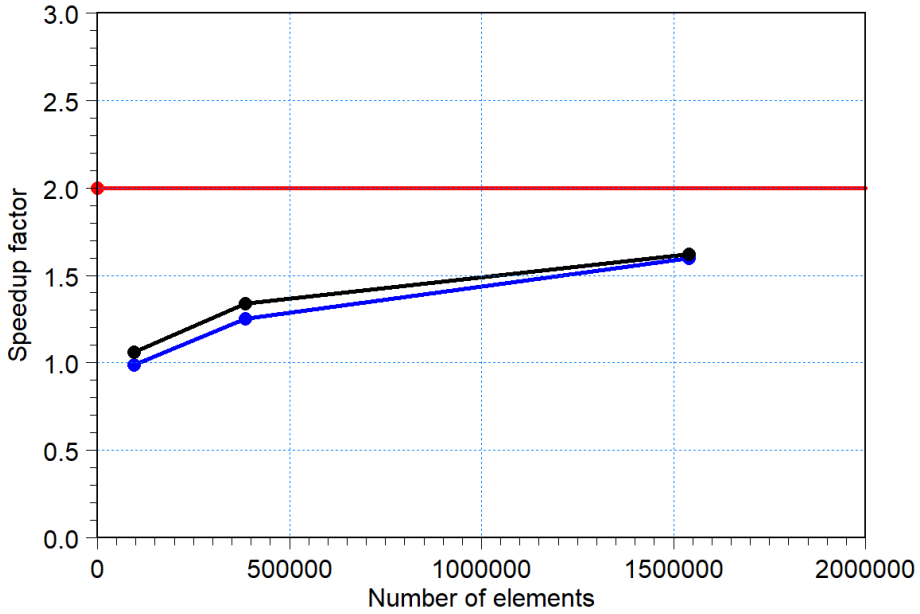

Figure 5.3 Speedup factor, $t_{GPU(1)}/t_{GPU(n)}$, for two GPUs relative to a single GPU. Blue line: single precision; Black line: double precision; Red line: Ideal speedup factor

Table 5.4 Speedup factors, $t_{CPU(1)}/t_{GPU(n)}$ and $t_{CPU(12)}/t_{GPU(n)}$. Simulations are carried out using single precision (SP) and double precision (DP) and using first-order scheme in time and space. The timings $t_{CPU(1)}$ and $t_{CPU(12)}$ are the timings from hardware platform 1 as in Table 4.5.

Mesh	No. of GPUs n	Speedup Factor $t_{CPU(1)}/t_{GPU(n)}$		Speedup Factor $t_{CPU(12)}/t_{GPU(n)}$	
		SP	DP	SP	DP
Mesh A	1	24.60	22.55	4.06	3.72
	2	24.47	24.12	4.04	3.98
Mesh B	1	49.99	42.60	8.18	6.97
	2	62.95	57.18	10.30	9.36
Mesh C	1	72.13	57.35	12.15	9.66
	2	116.10	93.38	19.55	15.72

6 Benchmarking using Tesla V100

These tests have been performed using a GPU instance on Microsoft Azure specified as hardware platform 3 in Table 2.1. The simulations have been performed using one and two Tesla V100 cards (1 and 2 subdomains and 1 thread).

6.1 Mediterranean Sea

Table 6.1 Computational time, $t_{GPU(n)}$,using GPU acceleration (1 and 2 subdomains and 1 thread) and speedup factor ,$t_{GPU(1)}/t_{GPU(n)}$. The simulations are carried out using single precision (SP) and double precision (DP) and using higher-order scheme in time and space

Mesh	No. of GPUs n	SP		DP	
		Time (s) $t_{GPU(n)}$	Speedup Factor $t_{GPU(1)}/t_{GPU(n)}$	Time (s) $t_{GPU(n)}$	Speedup Factor $t_{GPU(1)}/t_{GPU(n)}$
Mesh A	1	7.11	1.00	7.03	1.00
	2	9.93	0.71	9.84	0.71
Mesh B	1	33.95	1.00	40.23	1.00
	2	36.25	0.93	39.12	1.02
Mesh C	1	150.31	1.00	214.19	1.00
	2	116.11	1.29	146.93	1.45
Mesh D	1	113.02	1.00	181.71	1.00
	2	67.29	1.67	105.24	1.72
Mesh E	1	431.06	1.00	716.8	1.00
	2	230.8	1.86	387.63	1.84

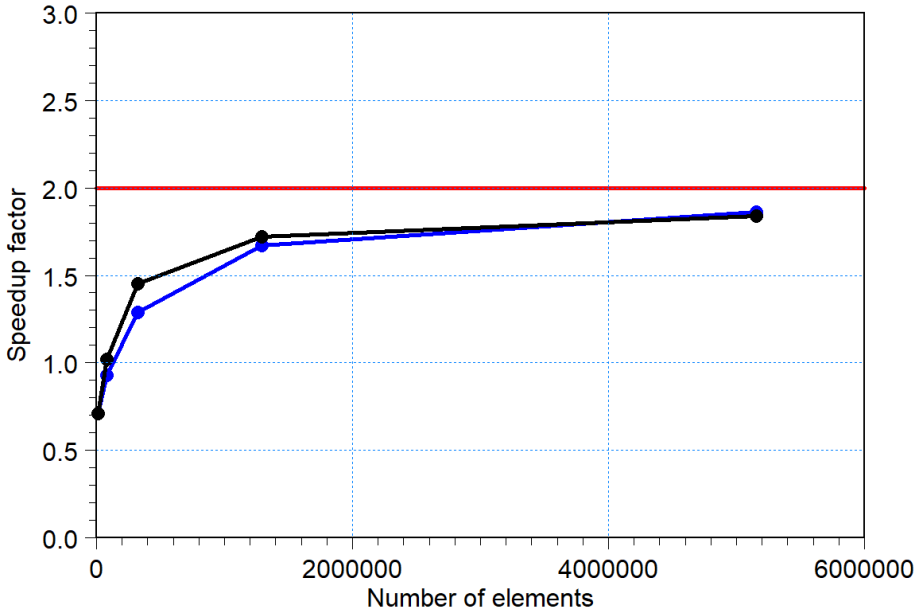

Figure 6.1 Speedup factor, $t_{GPU(1)}/t_{GPU(n)}$, for two GPUs relative to a single GPU using higher-order scheme in time and space. Blue line: single precision; Black line: double precision; Red line: Ideal speedup factor

Table 6.2 Speedup factors, $t_{CPU(1)}/t_{GPU(n)}$ and $t_{CPU(12)}/t_{GPU(n)}$. The simulations are carried out using single precision (SP) and double precision (DP) and using higher-order scheme in time and space. The timings $t_{CPU(1)}$ and $t_{CPU(12)}$ are the timings from hardware platform 1 as in Table 4.2.

Mesh	No. of GPUs n	Speedup Factor $t_{CPU(1)}/t_{GPU(n)}$		Speedup Factor $t_{CPU(12)}/t_{GPU(n)}$	
		SP	DP	SP	DP
Mesh A	1	33.00	33.38	3.19	3.23
	2	23.63	23.85	2.28	2.30
Mesh B	1	169.68	143.19	15.65	13.21
	2	158.92	147.26	14.66	13.58
Mesh C	1	298.86	209.73	29.84	20.94
	2	386.89	305.74	38.63	30.52
Mesh D	1	397.23	247.07	39.50	24.57
	2	667.19	426.60	66.34	42.42

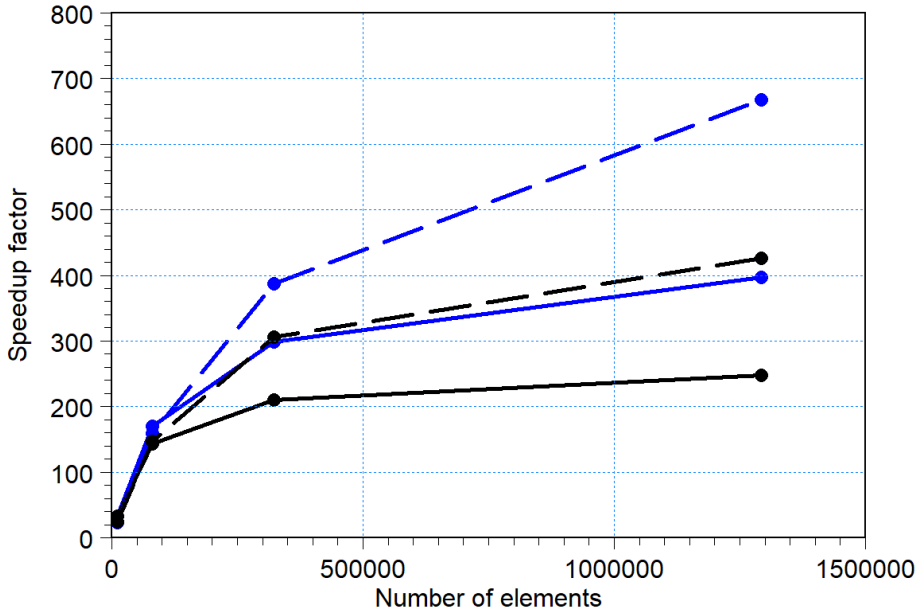

Figure 6.2 Speedup factor, $t_{CPU(1)}/t_{GPU(n)}$, for one and two Tesla V100 cards using higher-order scheme compared to the timings in Table 4.2. Blue line: single precision; Black line: double precision. Solid line: 1 GPU; Dash line: 2 GPUs

6.2 EA2D Test 8A

Table 6.3 Computational time, $t_{GPU(n)}$, using GPU acceleration (1 and 2 subdomains and 1 thread) and speedup factor ,$t_{GPU(1)}/t_{GPU(n)}$. The simulations are carried out using single precision (SP) and double precision (DP) and using first-order scheme in time and space

Mesh	No. of GPUs n	SP		DP	
		Time (s) $t_{GPU(n)}$	Speedup Factor $t_{GPU(1)}/t_{GPU(n)}$	Time (s) $t_{GPU(n)}$	Speedup Factor $t_{GPU(1)}/t_{GPU(n)}$
Mesh A	1	67.52	1.00	75.09	1.00
	2	76.50	0.88	75.91	0.98
Mesh B	1	244.58	1.00	269.17	1.00
	2	209.83	1.16	225.11	1.19
Mesh C	1	1207.52	1.00	1437.88	1.00
	2	797.06	1.51	966.15	1.48

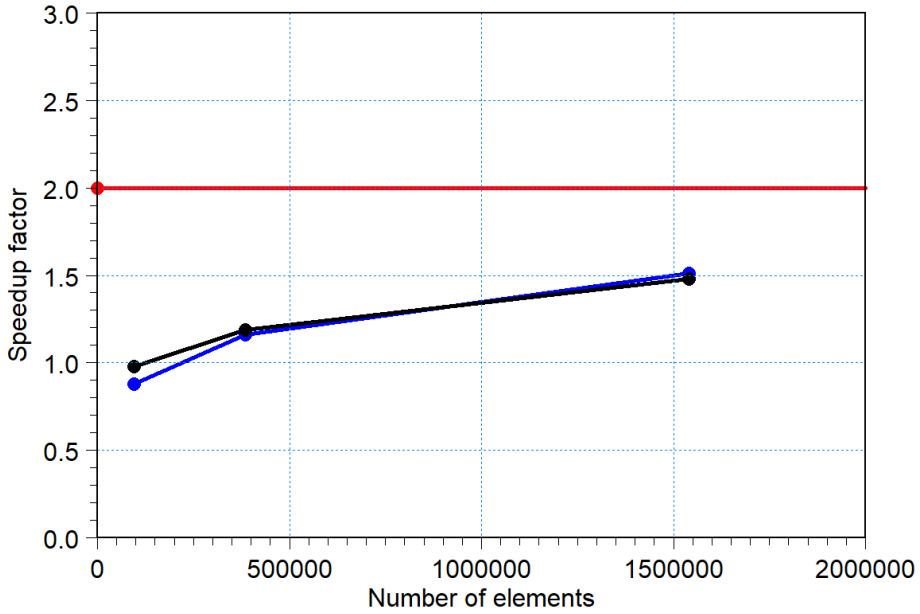

Figure 6.3 Speedup factor, $t_{GPU(1)}/t_{GPU(n)}$, for two GPUs relative to a single GPU. Blue line: single precision; Black line: double precision; Red line: Ideal speedup factor

Table 6.4 Speedup factors, $t_{CPU(1)}/t_{GPU(n)}$ and $t_{CPU(12)}/t_{GPU(n)}$. Simulations are carried out using single precision (SP) and double precision (DP) and using first-order scheme in time and space. The timings $t_{CPU(1)}$ and $t_{CPU(12)}$ are the timings from hardware platform 1 as in Table 4.5.

Mesh	No. of GPUs n	Speedup Factor $t_{CPU(1)}/t_{GPU(n)}$		Speedup Factor $t_{CPU(12)}/t_{GPU(n)}$	
		SP	DP	SP	DP
Mesh A	1	26.81	24.11	4.43	3.98
	2	23.66	23.85	3.91	3.94
Mesh B	1	60.11	54.62	9.84	8.94
	2	70.06	65.31	11.47	10.69
Mesh C	1	95.33	80.06	16.05	13.48
	2	144.43	119.15	24.32	20.07

7 Discussion

The performance strongly depends on the graphics card. This is illustrated in Figure 7.1, which shows the speedup factor, $t_{CPU(1)}/t_{GPU(1)}$, for the Mediterranean Sea case with higher-order scheme in time and space for the various graphics cards tested in this report. Since the Tesla K80 is a dual card consisting of two GPUs, it can be seen as one entity having two GPUs or it can be thought of as two separate GPUs. In this report the latter have been chosen. This approach eliminates the time spent on domain decomposition and communication between subdomains from the comparison.

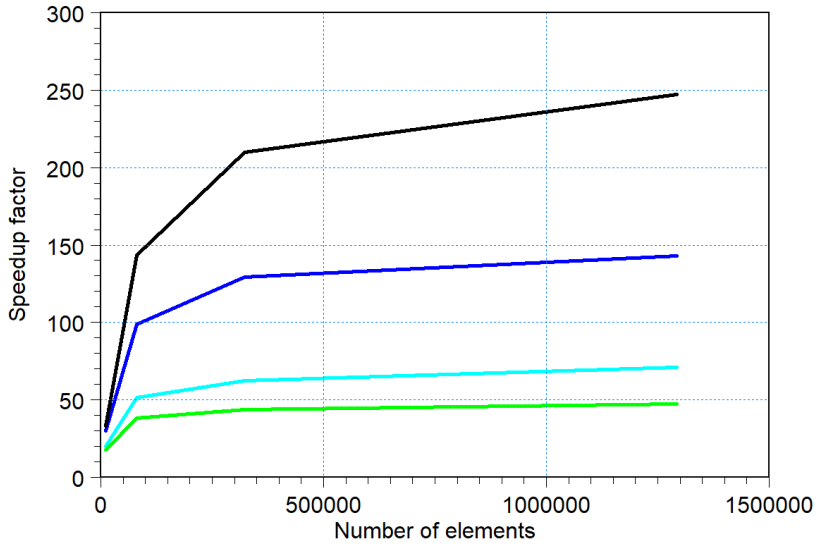

Figure 7.1 Comparison of the speedup factor, $t_{CPU(1)}/t_{GPU(1)}$, using different graphics cards. The Mediterranean Sea case with higher-order scheme in time and space. Green line: Tesla K80 card (1 GPU); Light blue line: Tesla K80 card (2 GPU); Blue line: Tesla P100 card; Black line: Tesla V100 card.

The results presented in Figure 7.1 clearly show that the Tesla V100 card performs better than the Tesla P100 card, which again performs better than the Tesla K80 card. Considering the GPU hardware specifications listed in Table 2.2, this is very much expected. For simulations having a small number of elements in the computational grid, the difference in performance is small between the different GPU cards. However, for simulations having a large number of elements in the computational grid, the difference becomes significant.

As seen in Figure 7.2 the simulations are faster when using single precision floating point calculations than when using double precision floating point calculations. Considering the hardware specifications in Table 2.2 this is expected, since all the tested GPUs have a higher theoretical single precision performance than double precision performance. Furthermore, the amount of floating point data that is transferred between the CPU and GPU is halved when using single precision instead of double precision calculations. For simulations having a small number of elements in the computational grid, the speedup factor using single precision compared to using double precision is small. However, for simulations having a large number of elements in the computational grid, a speedup factor between 1.6 and 1.8 can be obtained. It is important to remember that the single precision results are less accurate than the double precision results.

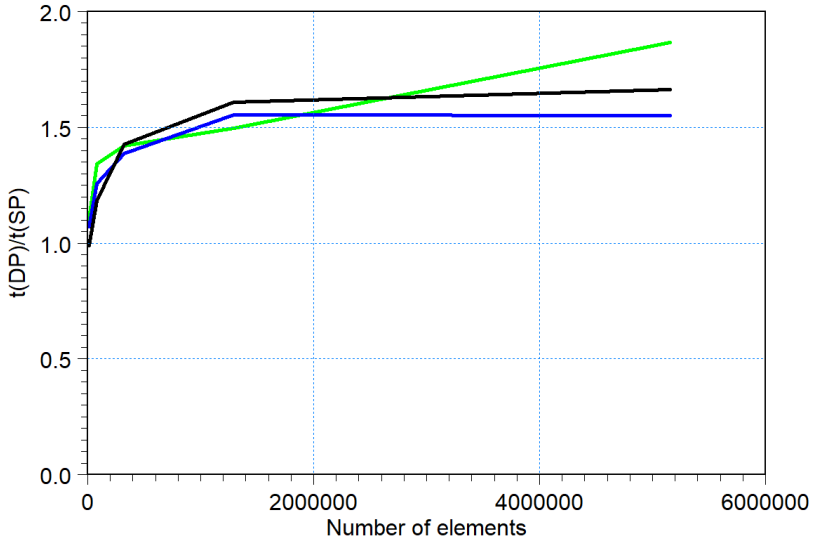

Figure 7.2 Comparison of the ratio between double and single precision calculations using different graphics cards (1 GPU). The Mediterranean Sea case with higher-order scheme in time and space. Green line: Tesla K80 card (1 GPU); Blue line: Tesla P100 card; Black line: Tesla V100 card.

On high-end shared memory workstations pure MPI parallelisation is typically more efficient than parallelisation using GPU acceleration when the number of elements is small. When the number of processors is increased for fixed problem size the efficiency will decrease. The suboptimal speedups can be explained by the workload imbalance and a high communication overhead. For large problems the efficiency will decrease for increasing problem size due to the increase in memory access time. This means that for large problems the use of parallelisation utilising GPU acceleration will significantly reduce the computational time compared to using pure MPI parallelisation. Especially when using multiple GPUs. This conclusion is of course very dependent on both the CPU and GPU hardware considered.

When the number of wet elements in the considered problem is sufficiently high, it is possible to obtain nearly ideal speed-up using multiple GPUs relative to using one GPU. Depending on the considered problem it is even possible to get superlinear speed-up. Of course, the communication overhead increases when using multiple GPUs, but as long as the problem is large enough for each GPU to have full work-load the scalability over multiple GPUs is very good.

8 Conclusions

The overall conclusions of the benchmarks are

- The numerical scheme and the implementation of the GPU version of the MIKE 21 Flow Model FM are identical to the CPU version of MIKE 21 Flow Model FM. Simulations without flooding and drying produces identical results using the two versions. Simulations with extensive flooding and drying produce results that may contain small differences.
- The performance of the GPU version of MIKE 21 Flow Model FM depends highly on the graphics card and the model setup. When evaluating the performance by comparing with a single core (no parallelisation) CPU simulation the performance also depends highly on the specifications for the CPU.
- The speedup factor of simulations with no flooding and drying increases with increasing number of elements in the computational mesh. When the number of elements becomes very large there is very limited or no increase in the speedup factor for increasing number of elements.
- The use of multi-GPU shows excellent performance. To get the optimal speedup factor a large number of elements is required for each sub-domain.
- Even on high-end shared-memory workstations the use of GPU can significantly improve the performance compared to the use of pure MPI.

9 References

/1/ Andersen, O.B., 1995, Global ocean tides from ERS-1 and TOPEX/POSEIDON altimetry, J. Geophys Res. 100 (C12), 25,249-25,259.

/2/ Néelz S., Pender G., 2010, Benchmarking of 2D Hydraulic Modelling Packages, Report published by Environment Agency, www.environment-agency.gov.uk, Copies of this report are available from the publications catalogue: http://publications.environment-agency.gov.uk