MIKE
Powered by DHI

MIKE 3 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 3 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 3 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 3 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 and the k-epsilon based turbulence 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 3 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 1core/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 3 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 three meshes with different resolution (see Table 3.1). The meshes are generated specifying the value for the maximum area of 0.04, 0.005 and 0.00125 degree^2, respectively. For the vertical discretization a sigma/z-level discretization is applied. The sigma depth is -100m. 5 sigma layers is used and 10 z-level layers with the layers thicknesses 1000m, 1000m, 1000m, 400m, 200m, 100m, 50m, 50m, 50m and 50m (from bottom to top). The simulation period for the benchmarks covers 2 days starting 1 January 2004 for the simulations using mesh A and B. The simulation period is reduced to 6 hours for the simulations using mesh C.

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

Coriolis force and tidal potential with 11 components (default values) are included in the simulations. For the bed resistance the roughness height of 0.05m is specified. For horizontal eddy viscosity the Smagorinsky formulation is used with a Smagorinsky factor of 0.5 and for the vertical viscosity the k-ε formulation is applied. Density effect is not included.

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

Table 3.1 Computational mesh for the Mediterranean Sea case

Mesh	Element shape	Elements 2D	Elements 3D	Max. area Degree^2
Mesh A	Triangular	11287	124269	0.04
Mesh B	Triangular	80968	923956	0.005
Mesh C	Triangular	323029	3683388	0.00125

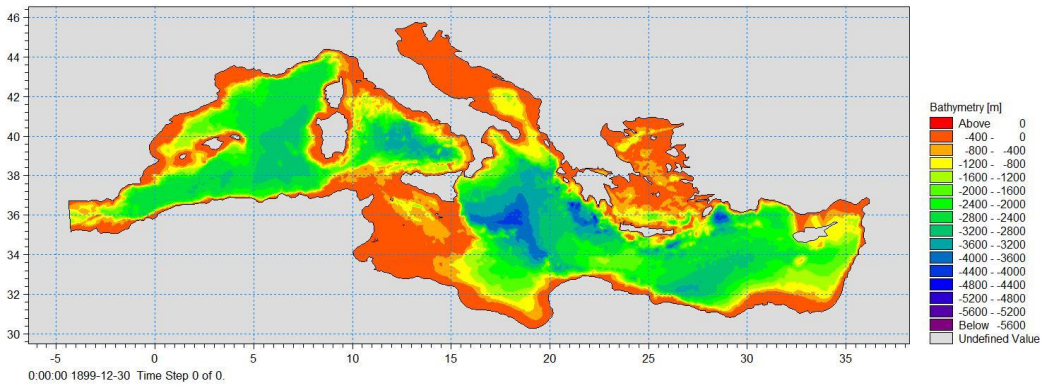

Figure 3.1 Bathymetry for the Mediterranean Sea case

The average time step for the simulations using Mesh A, B and C is 17.65s, 5.61s and 2.86s, respectively, for both the first-order scheme and the higher-order scheme in time and space.

3.2 Gulf of Mexico

This test case has been established for benchmarking of the MIKE 3 Flow Model FM.

3.2.1 Description

The flow in the Gulf of Mexico is characterized by a loop current system.

3.2.2 Setup

Simulations are performed using six meshes with different resolution (see Table 3.2). The mesh is created by specifying the maximum area in four different sub-areas. For mesh A the values for the maximum area are in the range 0.032 to 0.128. For mesh B, C D, E, F and G the values for maximum area are divided by 2, 4, 8, 16, 32 and 64, respectively, compared to the values used for mesh A. For the vertical discretization a sigma/z-level discretization is applied. The sigma depth is -100m. 10 sigma layers is used and 30 z-level layers with the layers thicknesses 500m, 500m, 500m, 500m, 400m, 400m, 300m, 300m, 200m, 200m, 200m, 200m, 200m, 150m, 150m, 100m, 100m, 50m, 50m, 30m, 25m, 25m, 20m, 20m, 15m, 15m, 15m, 15m, 10m, and 10m (from bottom to top). The simulation period for the simulations using six meshes covers 24, 18, 12, 9, 6, 3 and 2 hours.

Wind, Coriolis force and tidal potential with 11 components (default values) are included in the simulations. For the bed resistance the roughness height of 0.05m is specified. For horizontal eddy viscosity the Smagorinsky formulation is used with a Smagorinsky factor of 0.5 and for the vertical viscosity the k-ε formulation is applied. Density effect is included where the density is a function of the temperature and the salinity.

At the five open boundaries in the domain Flather boundary conditions are applied in the hydrodynamic calculations. The discharge from the Mississippi river, the Atachafalya river and the Brazos river are included.

The shallow water equations and the transport equations for temperature and salinity are solved using the higher-order scheme in time and space, while the transport equations for the turbulent kinetic energy and dissipation of the turbulent kinetic energy are solved using the first-order scheme in time and space.

Table 3.2 Computational mesh for the Gulf of Mexico case

Mesh	Element shape	Elements 2D	Elements 3D	Max. area Degree^2
Mesh A	Triangular	9874	256950	0.032-0128
Mesh B	Triangular	17369	441795	0.016-0.064
Mesh C	Triangular	32767	837746	0.008-0.032
Mesh D	Triangular	65773	1675605	0.004-0.016
Mesh E	Triangular	130524	3331724	0.002-0.008
Mesh F	Triangular	261991	6673420	0.001-0.004
Mesh G	Triangular	522356	13320907	0.0005-0.002

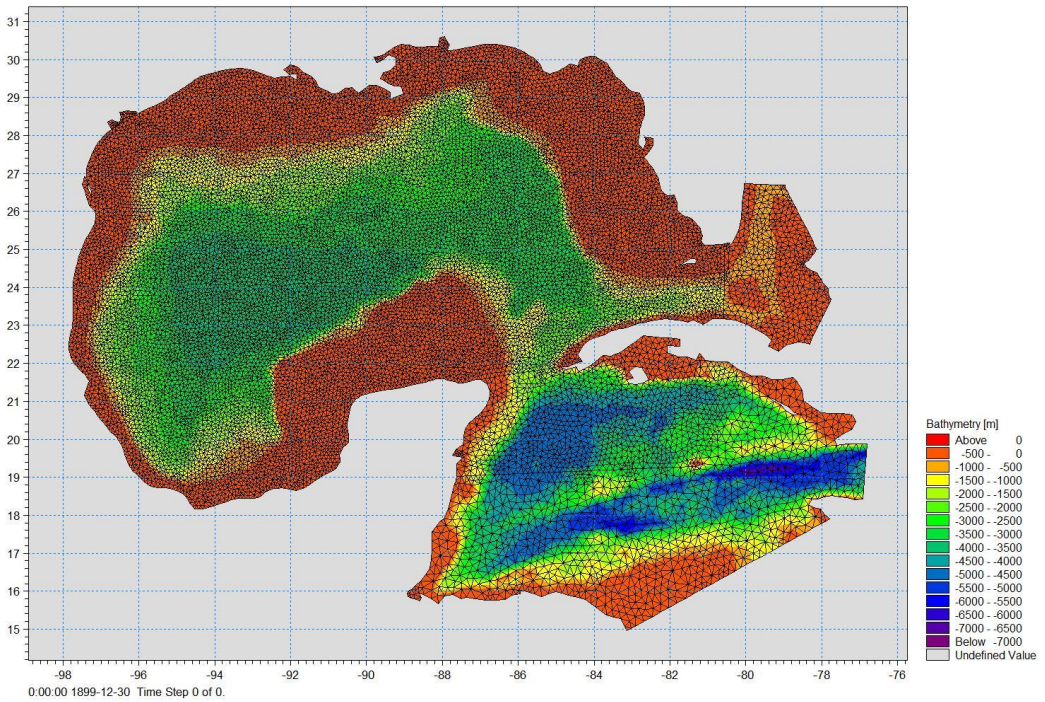

Figure 3.2 Bathymetry for the Gulf of Mexico case

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	89.02	1.00	119.04	1.00
	2	107.76	0.82	108.89	1.09
Mesh B	1	1357.93	1.00	2186.55	1.00
	2	911.8	1.48	1438.55	1.51
Mesh C	1	1293.81	1.00	2157.98	1.00
	2	789.89	1.63	1276.73	1.69

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 using 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	1685.06	1.00
	12	177.32	9.50
Mesh B	1	38925.21	1.00
	12	4443.06	8.76
Mesh C	1	37455.84	1.00
	12	3848.24	9.73

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	18.92	14.15	1.99	1.48
	2	15.63	15.47	1.64	1.62
Mesh B	1	28.66	17.80	3.27	2.03
	2	42.69	27.05	4.87	3.08
Mesh C	1	28.95	17.35	2.97	1.78
	2	47.41	29.33	4.87	3.01

4.2 Gulf of Mexico

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 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	90.19	1.00	115.42	1.00
	2	64.76	1.39	79.07	1.45
Mesh B	1	158.11	1.00	212.15	1.00
	2	98.25	1.60	127.48	1.66
Mesh C	1	309.28	1.00	422.48	1.00
	2	186.61	1.65	268.59	1.57
Mesh D	1	618.8	1.00	857.31	1.00
	2	389.78	1.58	548.03	1.56
Mesh E	1	1178.73	1.00	1647.66	1.00
	2	710.64	1.65	1028.25	1.60
Mesh F	1	1523.31	1.00	2124.62	1.00
	2	902.8	1.68	1303.72	1.62
Mesh G	1	3168.56	1.00	4389.19	1.00
	2	1811.78	1.74	2544.44	1.72

Table 4.5 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 using 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	1638.18	1.00
	12	176.73	9.26
Mesh B	1	3207.49	1.00
	12	357.09	8.98
Mesh C	1	6936.05	1.00
	12	779.25	8.90
Mesh D	1	14260.73	1.00
	12	1486.08	9.59
Mesh E	1	27383.13	1.00
	12	2793.51	9.80
Mesh F	1	36074.01	1.00
	12	3643.09	9.90
Mesh G	1	74911.67	1.00
	12	7467.28	10.03

Table 4.6 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	18.16	14.19	1.95	1.53
	2	25.29	20.71	2.72	2.23
Mesh B	1	20.28	15.11	2.25	1.68
	2	32.64	25.16	3.63	2.80
Mesh C	1	22.42	16.41	2.51	1.84
	2	37.16	25.82	4.17	2.90
Mesh D	1	23.04	16.63	2.40	1.73
	2	36.58	26.02	3.81	2.71
Mesh E	1	23.23	16.61	2.36	1.69
	2	38.53	26.63	3.93	2.71
Mesh F	1	23.68	16.97	2.39	1.71
	2	39.95	27.67	4.03	2.79
Mesh G	1	23.64	17.06	2.35	1.70
	2	41.34	29.44	4.12	2.93

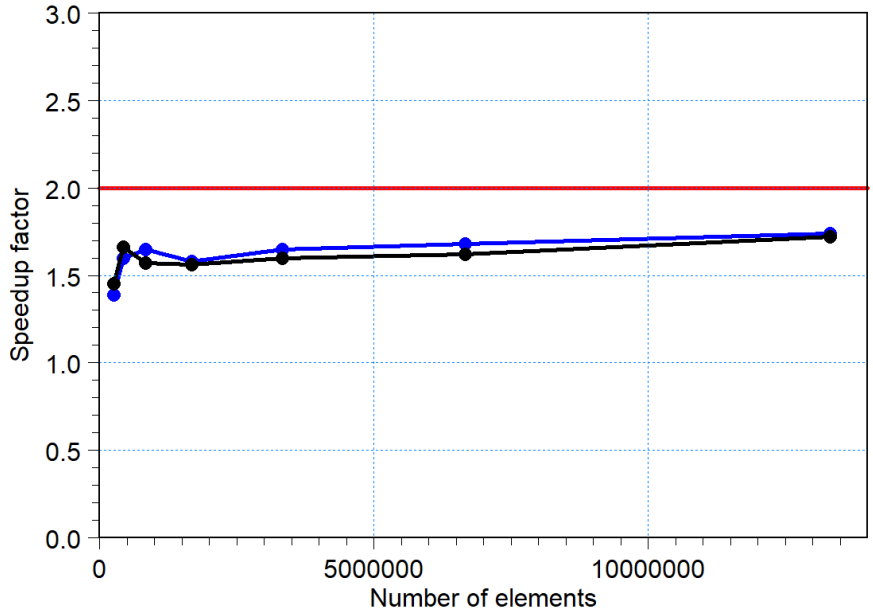

Figure 4.1 Speedup factor, $t_{GPU(1)}/t_{GPU(n)}$, for two GPUs relative to a single GPU using higher-order scheme in time and space. Blue line: single precision; Black line: double precision; Red line: Ideal speedup factor

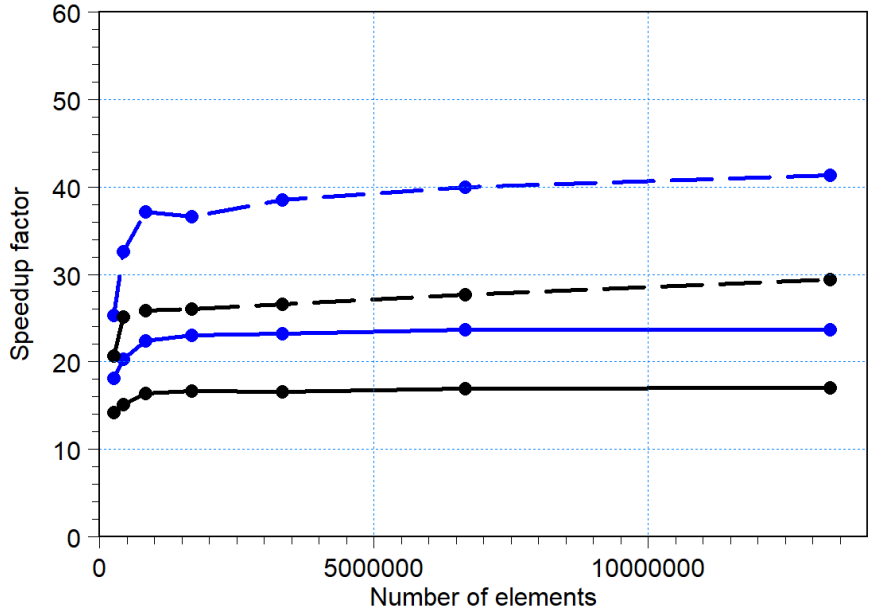

Figure 4.2 Speedup factor, $t_{CPU(1)}/t_{GPU(n)}$, using only one or both of the GPUs on the Tesla K80 using higher-order scheme. Blue line: single precision; Black line: double precision and solid line: 1 GPU and dash line: 2 GPUs

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 a Tesla P100 card (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	60.09	1.00	65.64	1.00
	2	85.65	0.70	85.47	0.76
Mesh B	1	466.83	1.00	665.61	1.00
	2	395.09	1.18	508.79	1.30
Mesh C	1	377.28	1.00	583.68	1.00
	2	231.33	1.63	353.95	1.64

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	28.04	25.67	2.95	2.70
	2	19.67	19.71	2.07	2.07
Mesh B	1	83.38	58.48	9.51	6.67
	2	98.52	76.50	11.24	8.73
Mesh C	1	99.27	64.17	10.19	6.59
	2	161.91	105.82	16.63	10.87

5.2 Gulf of Mexico

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 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	48.61	1.00	54.91	1.00
	2	48.66	0.99	54.67	1.00
Mesh B	1	68.98	1.00	82.86	1.00
	2	63.37	1.08	71.88	1.15
Mesh C	1	110.56	1.00	147.5	1.00
	2	86.27	1.28	109.31	1.34
Mesh D	1	206.90	1.00	289.85	1.00
	2	139.64	1.48	186.67	1.55
Mesh E	1	379.73	1.00	533.39	1.00
	2	227.64	1.66	318.91	1.67
Mesh F	1	466.97	1.00	662.13	1.00
	2	273.61	1.70	381.14	1.73
Mesh G	1	913.91	1.00	1298.46	1.00
	2	502.10	1.82	750.32	1.73

Table 5.4 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 4.5.

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	33.70	29.83	3.63	3.21
	2	33.66	29.96	3.63	3.23
Mesh B	1	46.49	38.70	5.17	4.30
	2	50.61	44.62	5.63	4.96
Mesh C	1	62.73	47.02	7.04	5.28
	2	80.39	63.45	9.03	7.12
Mesh D	1	68.92	49.20	7.18	5.12
	2	102.12	76.39	10.64	7.96
Mesh E	1	72.11	51.33	7.35	5.23
	2	120.29	85.86	12.27	8.75
Mesh F	1	77.25	54.48	7.80	5.50
	2	131.84	94.64	13.31	9.55
Mesh G	1	81.96	57.69	8.17	5.75
	2	149.19	99.83	14.87	9.95

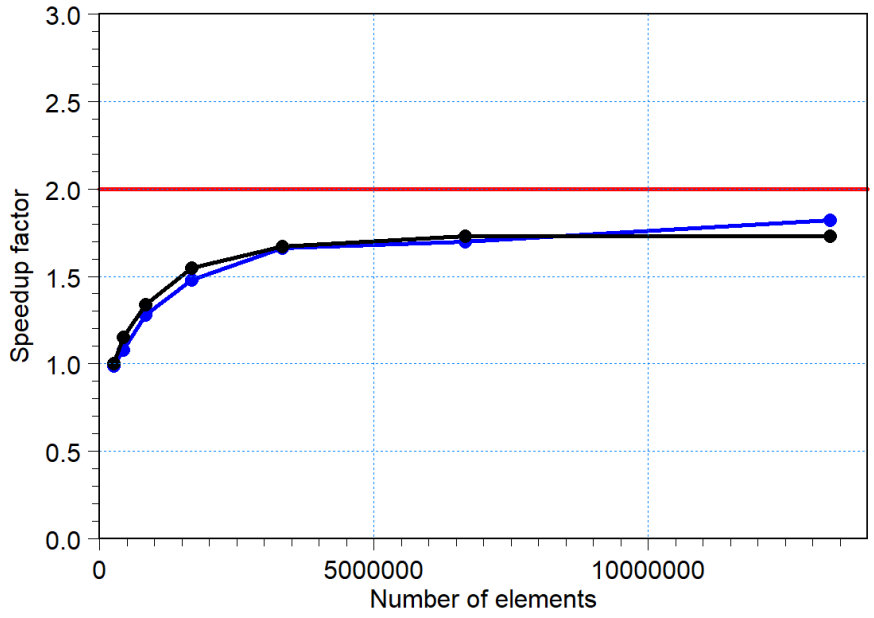

Figure 5.1 Speedup factor, $t_{GPU(1)}/t_{GPU(n)}$, for two GPUs relative to a single GPU using higher-order scheme in time and space. Blue line: single precision; Black line: double precision; Red line: Ideal speedup factor

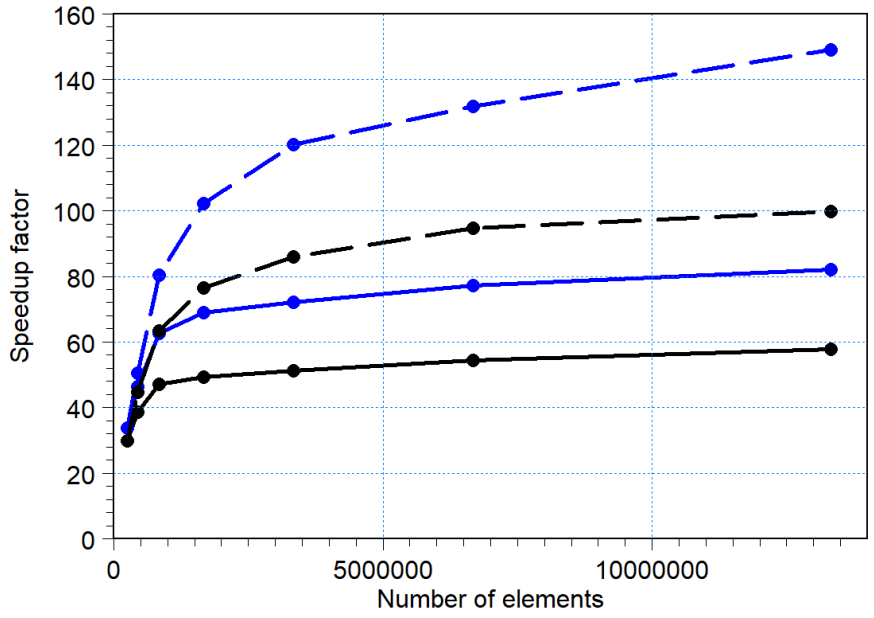

Figure 5.2 Speedup factor, $t_{CPU(1)}/t_{GPU(n)}$, for one and two Tesla P100 cards using higher-order scheme compared to the timings in Table 4.5. Blue line: single precision; Black line: double precision and solid line: 1 GPU and dash line: 2 GPUs

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 a Tesla V100 card (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	75.07	1.00	77.93	1.00
	2	118.84	0.63	120.58	0.64
Mesh B	1	391.83	1.00	520.46	1.00
	2	449.61	0.87	529.95	0.98
Mesh C	1	247.06	1.00	401.72	1.00
	2	193.23	1.27	268.38	1.49

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	22.44	21.62	2.36	2.27
	2	14.17	13.97	1.49	1.47
Mesh B	1	99.34	74.79	11.33	8.53
	2	86.57	73.45	9.88	8.38
Mesh C	1	151.60	93.23	15.57	9.57
	2	193.84	139.56	19.91	14.33

6.2 Gulf of Mexico

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 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	51.08	1.00	55.27	1.00
	2	64.96	0.78	68.64	0.80
Mesh B	1	66.91	1.00	75.53	1.00
	2	77.92	0.85	84.60	0.89
Mesh C	1	95.90	1.00	119.04	1.00
	2	98.71	0.97	111.36	1.06
Mesh D	1	160.35	1.00	215.81	1.00
	2	137.41	1.16	170.79	1.26
Mesh E	1	275.84	1.00	387.36	1.00
	2	199.67	1.38	254.17	1.52
Mesh F	1	329.31	1.00	470.89	1.00
	2	203.72	1.61	279.53	1.68
Mesh G	1	628.35	1.00	920.53	1.00
	2	352.41	1.78	509.00	1.80

Table 6.4 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 4.5.

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	32.07	29.63	3.45	3.19
	2	25.21	23.86	2.72	2.57
Mesh B	1	47.93	42.46	5.33	4.72
	2	41.16	37.91	4.58	4.22
Mesh C	1	72.32	58.26	8.12	6.54
	2	70.26	62.28	7.89	6.99
Mesh D	1	88.93	66.08	9.26	6.88
	2	103.78	83.49	10.81	8.70
Mesh E	1	99.27	70.69	10.12	7.21
	2	137.14	107.73	13.99	10.99
Mesh F	1	109.54	76.60	11.06	7.73
	2	177.07	129.05	17.88	13.03
Mesh G	1	119.21	81.37	11.88	8.11
	2	212.56	147.17	21.18	14.67

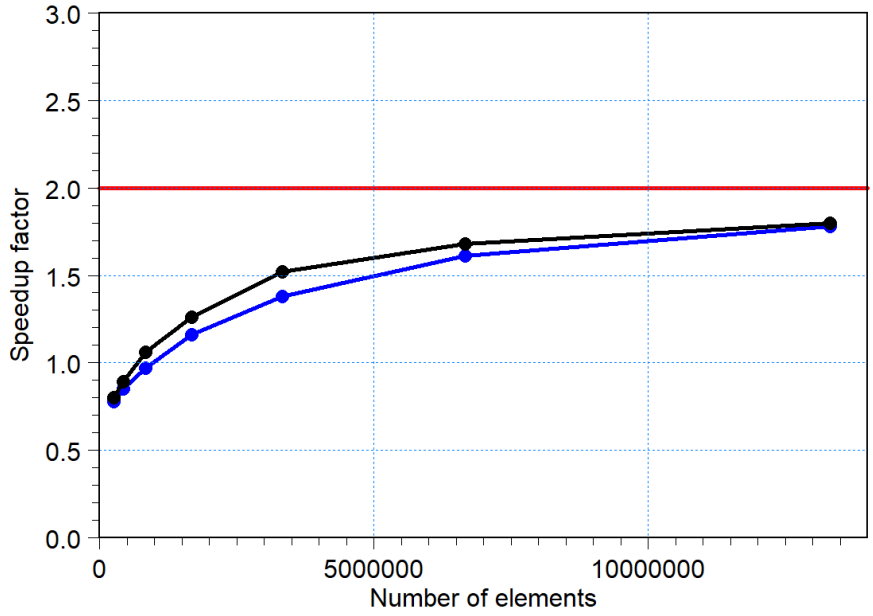

Figure 6.1 Speedup factor, $t_{GPU(1)}/t_{GPU(n)}$, for two GPUs relative to a single GPU using higher-order scheme in time and space. Blue line: single precision; Black line: double precision; Red line: Ideal speedup factor

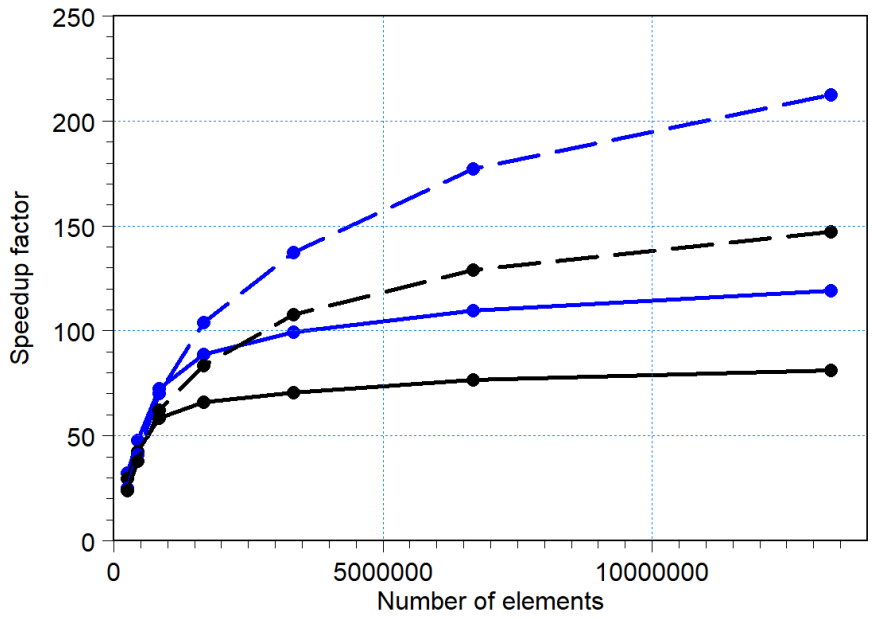

Figure 6.2 Speedup factor, $t_{CPU(1)}/t_{GPU(n)}$, for one and two Tesla V100 cards using higher-order scheme compared to the timings in Table 4.5. Blue line: single precision; Black line: double precision and solid line: 1 GPU and dash line: 2 GPUs

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 Gulf of Mexico 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.

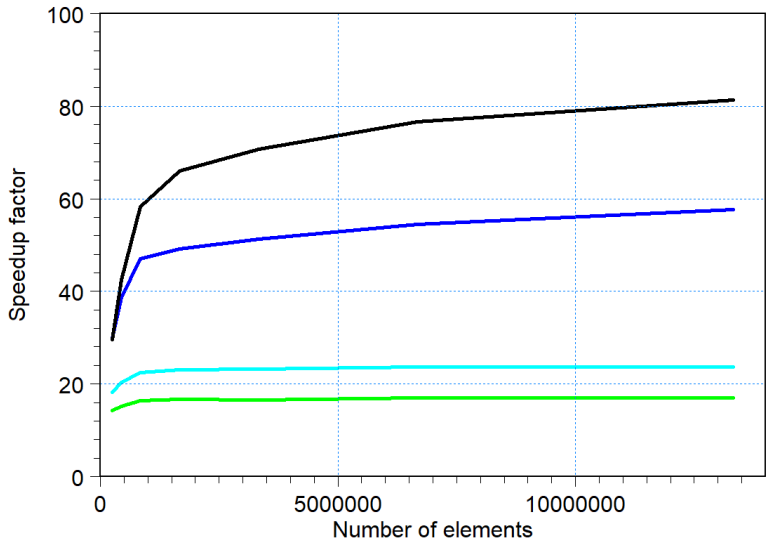

Figure 7.1 Comparison of the speedup factor, $t_{CPU(1)}/t_{GPU(1)}$, using different graphics cards. The Gulf of Mexico case with higher-order scheme in time and space. Green line: Tesla K80 card (1 GPU); Light blue line: Tesla K80 card (2 GPU); Blue line: Tesla P100 card; Black line: Tesla V100 card.

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 large number of elements in the computational grid, the difference in performance becomes very significant between the different GPU cards.

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 approximately 1.4 can be obtained. It is important to remember that the single precision results are less accurate than the double precision results.

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

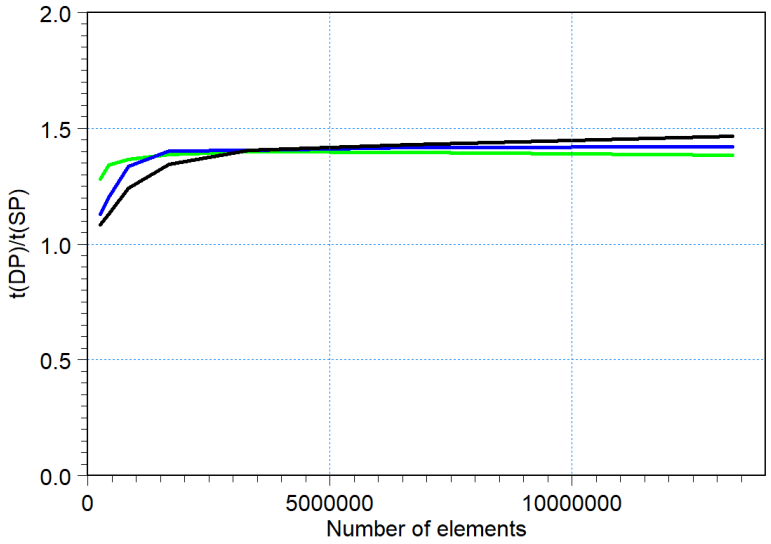

Figure 7.2 Comparison of the ratio between double and single precision calculations using different graphics cards (1 GPU). The Mediterranean Sea case with higher-order scheme in time and space. Green line: Tesla K80 card (1 GPU); Blue line: Tesla P100 card; Black line: Tesla V100 card.

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 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. Of course, the communication overhead will increase 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 3 Flow Model FM are identical to the CPU version of MIKE 3 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 new GPU version of MIKE 3 Flow Model FM depends highly on the graphics card and the model setup. When evaluating the performance by comparing with a 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.