

MIKE Powered by DHI

GPU - Guidelines



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

COPYRIGHT

This document refers to proprietary computer software, which is protected by copyright. All rights are reserved. Copying or other reproduction of this manual or the related programmes is prohibited without prior written consent of DHI. For details, please refer to your 'DHI Software Licence Agreement'.

LIMITED LIABILITY

The liability of DHI is limited as specified in your DHI Software License Agreement:

In no event shall DHI or its representatives (agents and suppliers) be liable for any damages whatsoever including, without limitation, special, indirect, incidental or consequential damages or damages for loss of business profits or savings, business interruption, loss of business information or other pecuniary loss arising in connection with the Agreement, e.g. out of Licensee's use of or the inability to use the Software, even if DHI has been advised of the possibility of such damages.

This limitation shall apply to claims of personal injury to the extent permitted by law. Some jurisdictions do not allow the exclusion or limitation of liability for consequential, special, indirect, incidental damages and, accordingly, some portions of these limitations may not apply.

Notwithstanding the above, DHI's total liability (whether in contract, tort, including negligence, or otherwise) under or in connection with the Agreement shall in aggregate during the term not exceed the lesser of EUR 10.000 or the fees paid by Licensee under the Agreement during the 12 months' period previous to the event giving rise to a claim.

Licensee acknowledge that the liability limitations and exclusions set out in the Agreement reflect the allocation of risk negotiated and agreed by the parties and that DHI would not enter into the Agreement without these limitations and exclusions on its liability. These limitations and exclusions will apply notwithstanding any failure of essential purpose of any limited remedy.

PRINTING HISTORY

February 2016	Release 2016
January 2018	Release 2017
November 2018.....	Release 2019
November 2022.....	Release 2023

CONTENTS

MIKE Powered by DHI GPU - Guidelines

1	Background	1
2	Hardware	3
3	GPU Configuration	5
4	Model Setup	7
5	Frequently Asked Questions	9
5.1	Why does the MIKE software not find my GPU?	9
5.2	How do I change the driver mode of the GPU?	10
5.3	Does my GPU have a CUDA Double Precision switch?	10
5.4	Which GPUs have been tested by DHI?	11
5.5	How many subdomains and threads should I use?	11
5.6	How do I run GPU simulations in batch?	12
5.7	Why did my GPU simulation crash?.....	14

1 Background

In MIKE 21 Flow Model FM, MIKE 3 Flow Model FM (Shallow Water equations) and MIKE 21/3 Coupled Model FM, the Hydrodynamic module and the Transport module are capable of utilising graphics processing units (GPUs) for parallelisation.

Reaching optimal performance of the GPU is not always straightforward. There are a number of pitfalls that should be avoided and a number of things worth keeping in mind.

These guidelines are divided into four sections and will address some of the issues related to: Hardware, GPU Configuration and Model Setup. Finally, the fourth section consists of a number of Frequently Asked Questions (FAQs).

2 Hardware

Some of the things worth considering are related to the GPU hardware and the installation of it:

- Only NVIDIA GPUs of compute capability 6.0 or higher are supported. A list of GPUs and their compute capability is found at: <https://developer.nvidia.com/cuda-gpus>.
- A GPU is not just a GPU. Some GPUs are suitable for gaming, some for visualisation and others for scientific computations. To obtain good performance in a scientific computing context the GPU should have a high number of Streaming Multiprocessors and cores (shading units). Other important specifications are the amount of memory and the bandwidth. An essential value is the theoretical double precision floating-point performance (FLOPS rate). Some GPUs have excellent single precision floating-point performance, but less impressive double precision floating-point performance.
- When installing the GPU in the computer, make sure to install the GPU to the fastest PCI Express slot available. Otherwise, an unnecessary data transfer bottleneck may have been created. When utilising multiple GPUs the PCI Express slots should have identical speed in order to obtain optimal performance.
- Among the fastest PCI Express slots available, choose the slot with the most physical space around the GPU, such that down-clocking due to reaching the heat limit (GPU driver specified) and the risk of computer shut down due to overheating are reduced. Potentially, an extra cooling system is needed to ensure a sufficient heat transportation.
- Make sure that the computer has a power supply, which is large enough to cover the total power consumption of all the installed GPUs.
- When utilising multiple GPUs the best performance is obtained using identical GPUs. Otherwise, the GPU with the lowest performance dictates the overall performance.

3 GPU Configuration

After having installed the GPU there are a few things to consider with respect to configuration:

- Use the latest NVIDIA driver supporting the considered GPU. Drivers can be found at <http://www.nvidia.com/Download/index.aspx>.
- Some GPUs offer the possibility to switch driver mode from WDDM (Windows Display Driver Model) to TCC (Tesla Compute Cluster), which can have a significant impact on performance. Note, this is only possible if the GPU does not have to drive a display. Please see the FAQ in Section 5.2 for more information.
- A few NVIDIA GeForce GPUs have a CUDA Double Precision switch, which can be enabled from the NVIDIA Control Panel. The default setting for the MIKE FM models capable of utilising GPUs is to perform the floating-point calculations taking place on the GPU in double precision. Hence, enabling the CUDA Double Precision switch will improve performance of the models significantly. Note that not all GPUs have this feature. Please see the FAQ in Section 5.3 for information on how to determine whether or not the GPU has this switch and how to enable it.
- The NVIDIA Tesla GPUs support ECC (Error Correcting Code) memory protection, which helps reducing data errors. However, by disabling this feature it is possible to increase performance. In the benchmarking reports mentioned in Section 5.4 ECC was disabled.

4 Model Setup

When setting up the model there are a few things to keep in mind regarding performance:

- To obtain good performance for the MIKE FM models using GPU parallelisation the number of wet elements in the domain should be sufficiently large. Here, the emphasis is on the wording “wet elements”, since it is not sufficient to have a huge number of elements representing the domain if there is almost no water in the domain.
For the test cases presented in the benchmarking reports (see section 5.4), the maximum performance of the tested GPUs is reached only when each subdomain consists of several hundreds of thousands of wet elements.
If the model subdomain for each GPU consists of only a small number of wet elements, then the MIKE FM models could in fact be slower when utilising the GPU than when not utilising it.
- Only output what is necessary, and only when it is necessary. Writing output from the GPU is time consuming due to transferring data from GPU memory to CPU memory.

5 Frequently Asked Questions

This section lists a few FAQs related to the usage of GPU. For the full list of FAQs for the MIKE software, please visit <http://faq.dhigroup.com/>.

5.1 Why does the MIKE software not find my GPU?

If the GPU is not recognised by the MIKE software it may be due to an outdated GPU driver. Please try updating the driver. Drivers are found at this webpage: <http://www.nvidia.com/Download/index.aspx>.

Alternatively, the GPU might not be supported by the MIKE software. Only NVIDIA GPUs of compute capability 6.0 or higher are supported. A list of GPUs and their compute capability is found at: <https://developer.nvidia.com/cuda-gpus>.

If the GPU has been successfully recognised by the MIKE software, then the “Parallelization” tab in the “Launch Settings” dialog in the respective MIKE FM editor will look similar to the left picture in Figure 5.1. Otherwise, it will look similar to the right picture in Figure 5.1.

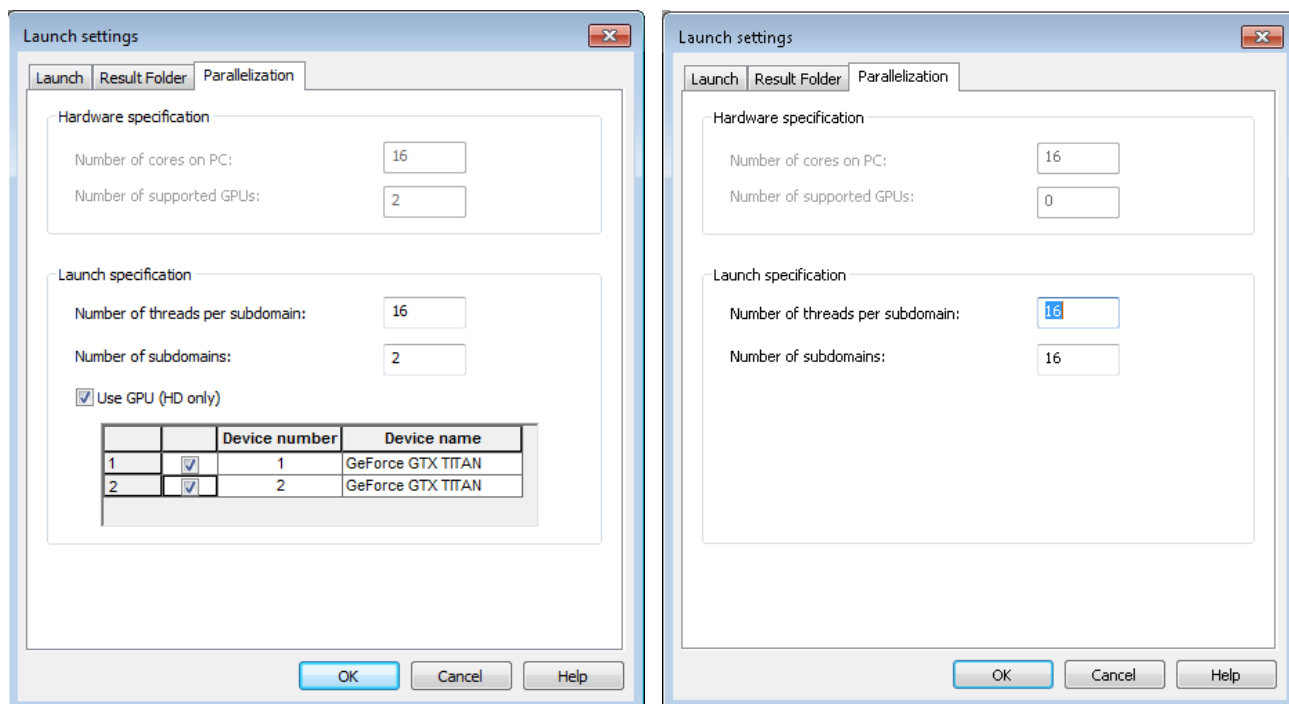


Figure 5.1 *Left:* Dialog where the GPUs have been recognised. *Right:* Dialog where the GPUs have not been recognised

5.2 How do I change the driver mode of the GPU?

Some GPUs offer the possibility to switch driver mode from WDDM (Windows Display Driver Model) to TCC (Tesla Compute Cluster). Note, this is only possible if the GPU does not have to drive a display.

From a performance perspective, the TCC mode is the preferred option, as it has a significant effect on the performance. Per default the GeForce and Quadro GPUs are set to WDDM and the Tesla GPUs are set to TCC.

Driver settings can be modified using the command line tool `nvidia-smi.exe`, which should be directly available from the command prompt after installing the graphics driver.

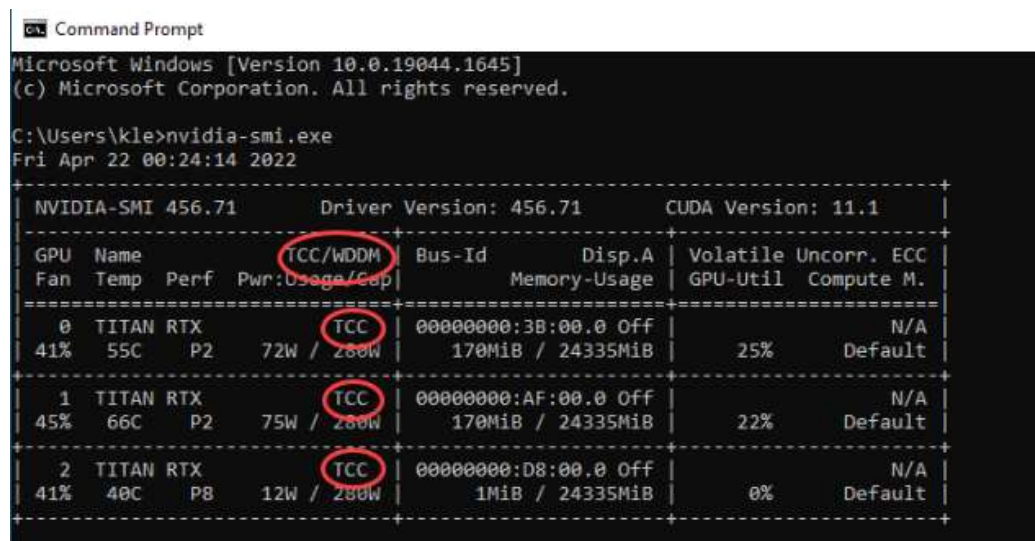
The driver mode has to be set for each GPU individually using the following call:

```
nvidia-smi.exe -i 0 -dm 1
```

where the number after `-i` specifies the id of the GPU and the number after `-dm` specifies the driver mode (WDDM:0 / TCC:1).

Changing this setting requires a reboot of the machine.

Calling the tool without any arguments will give an overview of the available GPUs and their settings:



```

Microsoft Windows [Version 10.0.19044.1645]
(c) Microsoft Corporation. All rights reserved.

C:\Users\kle>nvidia-smi.exe
Fri Apr 22 00:24:14 2022

+-----+
| NVIDIA-SMI 456.71      | Driver Version: 456.71      | CUDA Version: 11.1      |
+-----+-----+
| GPU  Name            | TCC/wDDM | Bus-Id      | Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf    | Pwr:Usage/Cap |      Memory-Usage | GPU-Util  Compute M. |
+-----+-----+-----+-----+-----+
| 0   TITAN RTX        | TCC      | 00000000:3B:00.0 Off |          | 25%                | N/A |
| 41%   55C    P2     | 72W / 200W | 170MiB / 24335MiB |          | Default            |
+-----+-----+-----+-----+-----+
| 1   TITAN RTX        | TCC      | 00000000:AF:00.0 Off |          | 22%                | N/A |
| 45%   66C    P2     | 75W / 200W | 170MiB / 24335MiB |          | Default            |
+-----+-----+-----+-----+-----+
| 2   TITAN RTX        | TCC      | 00000000:D8:00.0 Off |          | 0%                 | N/A |
| 41%   40C    P8     | 12W / 200W | 1MiB / 24335MiB  |          | Default            |
+-----+-----+-----+-----+-----+

```

5.3 Does my GPU have a CUDA Double Precision switch?

A few NVIDIA GeForce GPUs have a CUDA Double Precision switch, which can be enabled from the NVIDIA Control Panel. By enabling this switch more of the GPU's processing power is dedicated to performing double precision floating-point calculations.

Since the default setting for the MIKE software is to use double precision floating-point calculations on the GPU, enabling this switch will improve performance of the GPU simulations significantly.

Please note, that the NVIDIA Tesla GPUs does not have this switch, since these GPUs are designed explicitly for doing numerical calculations.

If the GPU has a CUDA Double Precision switch, then it can be enabled as illustrated in Figure 5.2. If the switch is not there, then the GPU does not support this feature.

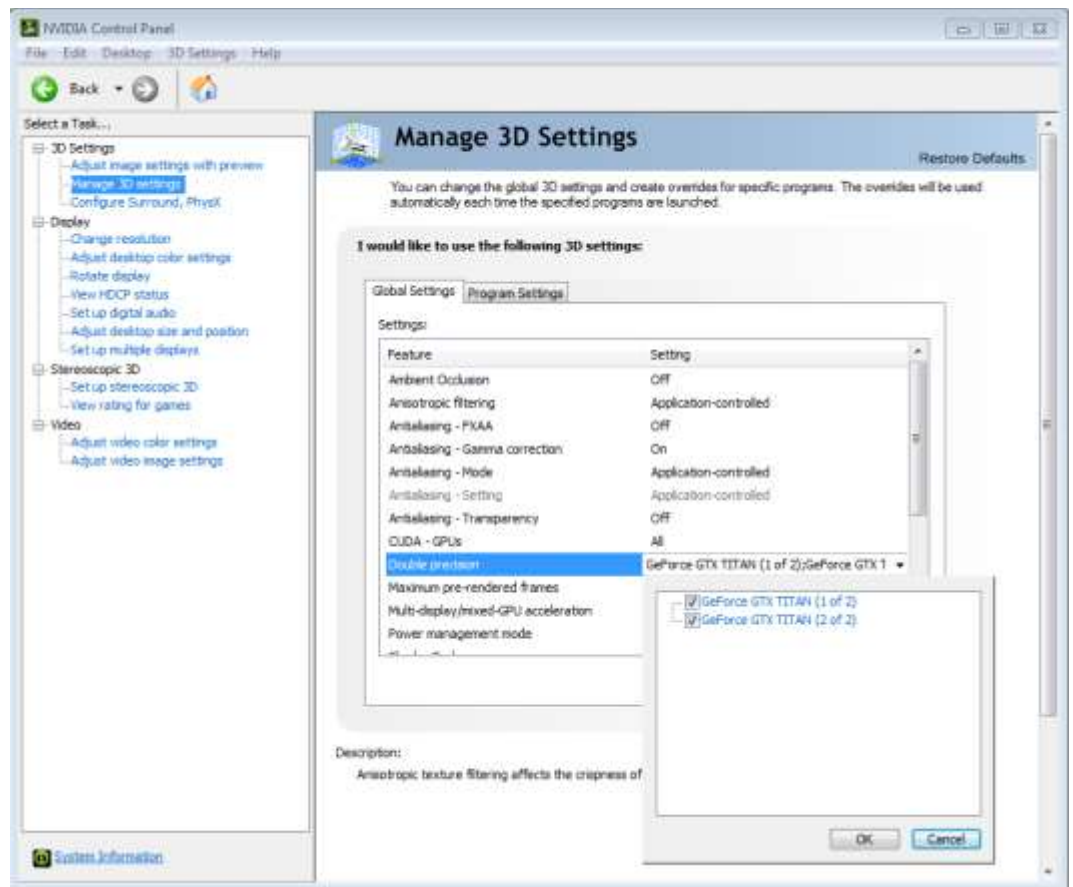


Figure 5.2 Screenshot of the NVIDIA Control panel showing how to enable the CUDA Double precision switch

5.4 Which GPUs have been tested by DHI?

Extensive benchmarking has been done for both MIKE 21 Flow Model FM utilising GPU and MIKE 3 Flow Model FM utilising GPU. The tests have been conducted using a number of different GPUs. The benchmarking results for MIKE 21 Flow Model FM and MIKE 3 Flow Model FM, respectively, are presented in the following two reports:

[MIKE 21 Graphical Processing Units \(GPU\) benchmarking report](#)

and

[MIKE 3 Graphical Processing Units \(GPU\) benchmarking report](#)

5.5 How many subdomains and threads should I use?

Launching a simulation from within the MIKE FM editors makes the “Launch settings” dialog appear. On the “Parallelization” tab in this dialog the user can specify which parallelisation technique to use for the simulation, please see Figure 5.3.

The number of subdomains specifies the number of domains to decompose the model domain into. Generally, this number should not exceed the number of CPU cores available on the machine. When utilising GPUs, this number should reflect the number of GPUs to use for the simulation. In rare occasions it might be beneficial, from a performance perspective, to use more subdomains than GPUs, but never less subdomains than GPUs.

The number of threads per subdomain specifies the number of OpenMP threads to use for each subdomain during the simulation. As a rule of thumb, this number times the number of subdomains should be less than or (preferably) equal to the total number of CPU cores available on the machine.

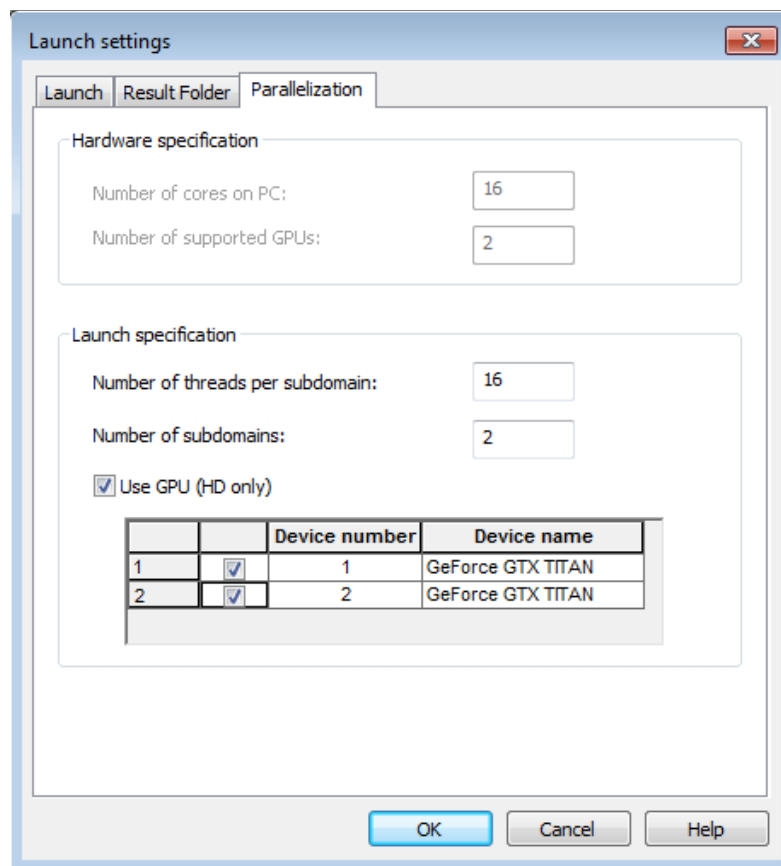


Figure 5.3 The “Launch Settings” dialog where parallelisation technique is specified

5.6 How do I run GPU simulations in batch?

There are a couple of possibilities on how to run GPU simulations in batch.

The first possibility is to use the MzLaunch.exe application, which is installed with the MIKE Zero installation, please see Figure 5.4.

In Figure 5.4, the first call to MzLaunch.exe is followed by the optional parameter -gpu 1, which instructs MzLaunch.exe to utilise the GPU using 1 subdomain. The second call instructs MzLaunch.exe to utilise the GPU using 2 subdomains. When using more than one subdomain, the application will automatically distribute the work over the same number of GPUs, if available.

```

RunGPUBatchMzLaunch.bat - Notepad
File Edit Format View Help
set Path=%Path%;%DHI_MIKE_2023%

start /w MzLaunch.exe Sim1.m21fm -gpu 1 -x

start /w MzLaunch.exe Sim2.m21fm -gpu 2 -x
    
```

Figure 5.4 Calling the MzLaunch.exe application with optional parameters for utilising the GPU

The second possibility for running GPU simulations in batch is to call the computational engine directly, without using the MzLaunch.exe application. The tricky part here is to figure out which executable to use for which setup-file. A job normally maintained by the MzLaunch.exe application.

The following table illustrates which executable being associated to which type of setup-file:

Table 5.1 Associating computational engines with file extensions.

Model	Executable	File extension
MIKE 21 Flow model FM	FemEngineHDGPU.exe	.m21fm
MIKE 3 Flow model FM	FemEngineHDGPU.exe	.m3fm
MIKE 21/3 Coupled Model FM	FemEngineGPU.exe	.mfm
MIKE FLOOD (M11)	FemEngineMFGPU.exe	.couple
MIKE FLOOD (M1D)	DHI.Couplings.Runner.exe	.couple

The commands given in Figure 5.5 illustrate three different ways of utilising the GPU by calling the computational engine directly. The command “set OMP_NUM_THREADS” specifies the number of threads used by each subdomain in the simulation.

The first call does not use the “mpiexec” command, meaning that only a single subdomain is used in the simulation. In this case the application automatically determines which GPU to use, if more are available.

The second command is used when more subdomains are used in the simulation. Again, the application automatically determines which GPU to use, if more are available.

The third command is used when more subdomains are used in the simulation and the user would like to specify which GPUs to use for the simulation. In this case the user has specified to use two GPUs (one for each subdomain), and the user would like to utilise GPU number 1 and GPU number 2. The syntax for specifying the optional parameters is:

```
-gpu NumberOfGPUs GPU1 GPU2
```

This means that if the user has 4 GPUs available on the machine, but only wants to use 3 of these, namely GPU1, GPU2 and GPU4, in the simulation, then the optional parameters would be:

-gpu 3 1 2 4

Of course, here the number of subdomains needs to be at least 3, since the user wants to use 3 GPUs.

The numbering of the GPUs refers to the numbers assigned to each GPU by the computational engine. This numbering is listed in the top of the simulation log file, please see Figure 5.6.

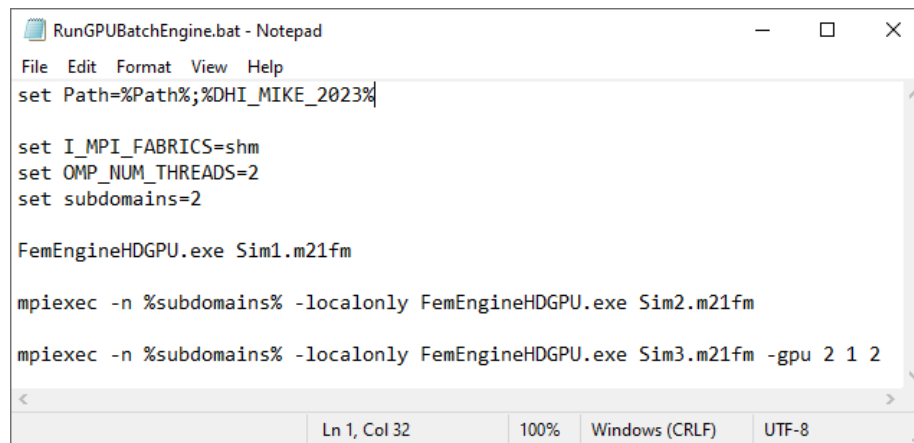


Figure 5.5 Calling the computational GPU engine directly for a MIKE 21 Flow Model FM simulation

```

===== CUDA GPU Device Information =====
Number of GPU devices      : 1

GPU device number         : 1
GPU device name           : Quadro M1000M
CUDA Compute Capability   : 5.0

Selected device number    : 1

number_of_threadsPerBlock : 128 (default)
=====

```

Figure 5.6 Section of the simulation log file showing GPU device information and the numbering assigned by the computational engine

Even though the above example only considers the FemEngineHDGPU.exe application and the .m21fm file format, the procedure is the same for the other executables and file formats listed in Table 5.1. The only exception to this is DHI.Couplings.Runner.exe, where it is not possible to specify which GPUs to use, meaning only the -gpu argument should be provided.

5.7 Why did my GPU simulation crash?

There can be a number of reasons why the simulation does not complete successfully ranging from wrong model input to hardware failure.

Most commonly, the reason for the simulation to stop is when a numerical blow-up happens in the model. This type of crash is reported in the log file, and it is most often caused by wrong or self-contradictory input to the model, which causes the model to be unstable.

Typically, this kind of crash is not related specifically to the utilisation of GPUs. This can be tested by running the simulation with GPU utilisation and then once more without utilising the GPU. If the blow-up happens in both simulations, then it is not related to utilisation of the GPU.

In other occasions, the progress bar in the model editor goes directly to 100% without the model creating a log file. Often, this crash happens because the user has insufficient write-privileges to the folder where the model setup is located. In such case, the crash would also happen when not utilising the GPU.

Alternatively, the reason might be an error in the GPU hardware. This type of error can be tricky to find, but one approach is to replace the GPU suspected of creating the error with a different GPU and check if the error still occurs.

In very rare occasions, the crash happens in the middle of an otherwise perfectly running simulation without leaving any message in the log file and the system is shut down and rebooted. Again, the reason can be faulty GPU hardware or perhaps a power supply which is not large enough to cover the total power consumption of the different hardware in the computer. However, it is also a possibility that the system was close to overheating and shut down as a safety measure. Indications of this might be found in the Windows Event Viewer. Since the GPU generates a lot of heat, it is important to have a good airflow inside the computer cabinet and a cooling system which ensures sufficient heat transportation.