

Dynamic 5 user manual
A GPU version of CMLAir

Tholfaqar (Dolf) Mardan and David B. Bogy

Computer Mechanics Lab.
Mechanical Engineering
University of California, Berkeley

April 26, 2016

Table of contents

1. List of symbols and acronyms.
2. Introduction to GPU Dynamic 5.0.
3. Features added.
4. Why GPU and Cuda FORTRAN?
5. Why Dynamic 5.0?
6. Computer system's requirements and installation.
 - a. Hardware requirements.
 - b. Software requirements.
7. Test results on example sliders.
8. Comparison between the two slider designs
9. Results discussion and conclusion.

List of symbols and acronyms

Vocabulary	Meaning
GPU	Graphical Processing Unit
DLL	Dynamic Link Library
NVIDIA	GPU manufacturing company located in San Jose
Cuda Fortran	Version of Fortran able to communicate with GPU
Kernel	Special subroutine in Cuda Fortran to link between Fortran 90 and GPU hardware
Cuda cores	Processors inside the GPU card
GDDR5 memory	Graphics Double Data Rate memory version 5
CC	A number refers to GPU Compute Compatibility
OpenAcc directives	A new method of rapping codes sent to GPU

Introduction to GPU Dynamic 5.0

Dynamic code version 5.0 is an upgrade from the Dynamic 4 code. It is more intelligent yet faster than Dynamic 4. It has its unique internal structure utilizing the computing power of a Graphical Processing Unit (GPU) efficiently for scientific heavily mathematical calculations. The main concept behind the dramatic speed improvement (2.6x in some complex slider designs) is the division of the task into a smaller portions and computing them in a synchronized and parallel manner rather than computing serially. Parallel computing has proven its superiority and speed efficiency over sequential computing.

GPU Dynamic 5.0 has new features that are not used in GPU Quick 5 called OpenAcc directives. It's a new method that is easier to use and more accurate with fewer errors to rap portion of the code intended to be sent to run on GPU hardware. This method shows more efficiency over the traditional GPU kernels. The Dynamic code implemented both methods to enhance both speed and accuracy.

Dynamic 5.0 has the same input files, and same output files as Dynamic 4. Since it runs on a different hardware structure than the conventional hardware such as computer CPU and RAM it needs a special DLL (dynamic link library) file to run successfully. This DLL is already included in the latest release of CMLAir8.4. The new Dynamic code has been tested successfully on the NVIDIA GeForce 960 GPU card. It can also run on any NVIDIA GPU card with CC (Compute Compatibility) of 3.5 or higher. We will talk about compatible GPU's in more detail in a later section of this manual.

The new CMLAir8.4 includes both the Dynamic solvers and the required DLL to run Dynamic 5.0. There is no need to download them separately; however, when you install CMLAir, the default Dynamic solver is Dynamic 4. If you have the proper NVIDIA hardware and software installed in your system and want to use Quick 5.0 instead, you need to go to the solver section in CMLAir and manually select GPU Dynamic 5.0 solver.

Features added

The following features and improvements have been added to the Dynamic code to create the newest and fastest Dynamic code version 5.0:

- 500 lines have been modified using the OpenACC directives method.
- 2 new OpenACC modules have been added to enhance portability of data handled by the code and gain optimum memory utilization.
- 2 files have been migrated from Fortran 77 to Fortran 90 to support Cuda FORTRAN.

Why GPU and Cuda FORTRAN?

- GPU supports parallel processing.
- It has more processors than the latest CPU.
- It has a more reliable, robust, easy to use and control multi-processing module than CPU.
- It works separately from CPU since it has its own hardware.
- It supports the implementation of multi GPU's.
- It is available and more affordable than additional CPU.
- Cuda C (by NVIDIA) and Cuda FORTRAN (by Portland group, 2007) are the two languages provided to communicate with NVIDIA GPU's.

Why Dynamic 5.0?

The essential part of the new Dynamic code version 5.0 runs on the GPU architecture. Furthermore, depending on the shape and initial conditions of the air bearing slider, one can realize different speed improvement characteristics compared to Dynamic code version 4. This can vary with the slider design, as shown in the examples and figures in a later section: 23% for a traditional and simple or moderate air bearing slider designs to as much as 60% (2.5X faster) for more complex designs.

Computer system's requirements:

a. Hardware requirements

During the testing and implementation on the new GPU Dynamic 5.0, we used a Dell desktop computer model: XPS 8500. It has powerful core i7-3770 processors, 3.4 GHz (two processors on a chip), 16 GB of DDR3 RAM and a PCI express slot version 3.0 where you attach the GPU to the motherboard and get maximum speed transferring data in and out of the GPU. A decent power supply on board is required since the GPU, during calculations, consumes a relatively large amount of DC power. Each GPU has its rated max power that needs to be noted and accommodated. Inside the Dell XPS 8500 we installed the following NVIDIA graphic card model: GeForce 960, which is a medium level GPU with powerful features and compelling price that is used by many video game enthusiasts. The only exception in this case, in CML and for the first time, we will be using it purely for scientific problem solving. A picture of the GPU is shown in Fig. 1. It has the interesting technical specifications given in Table 1.



Figure 1. NVIDIA GeForce 960 GPU card

Cuda Cores	1024
Base Clock (MHz)	1127
Boost Clock (MHz)	1178
Memory Size (GB)	2
Memory interface width	128-bit GDDR5
Memory bandwidth (GB/sec)	112
Interface Bus support	PCI Express 3.0
Power dissipation (Watts)	20 idle, 120 maximum
Minimum power supply size (Watts)	400
Dimensions (Inch)	9.5 length x 4.38 height x 1.5 width
Price (\$US)	220
Compute Compatibility (CC)	5.2

Table 1. NVIDIA GeForce 960 specifications chart.

As seen in Table 1 above, the GeForce GPU card has quite interesting characteristics that could be utilized for CML math calculations. It has 1024 cores running at 1.1 GHz speed. In addition it has 2 GB of fast GDDR5 memory with 128-bit attached and divided among the 1024 processors. This unique memory reduces the need for transferring data back and forth between the GPU and CPU and ultimately reduces the heap (memory reserved for intermediate calculations) and obviously speeds up problem solving. Having a PCI express version 3.0 bus takes data transfer to a new level compared to older GPUs. That explains the ultra-fast transfer rate of 112 GB/second between the GPU and CPU and its peripherals. On the other hand, such a GPU needs substantial power to perform its operations when running simulations. The reasons why we chose the Dell XPS 8500 are: first, it has the minimum required power supply by the GeForce 960, second, it has a PCI express 3.0 bus interface on its motherboard and ultimately, its case and the inside space is more than enough to accommodate comfortably the GeForce 960. It is recommended not to use the DVD/CD for playing or burning while using GPU Dynamic 5.0 to protect the power supply from overheating. No further action is needed in regards to the heat generated by the card when working at its maximum capacity since the GPU has a large centralized fan on top of the main chip specially designed for this task, as illustrated in Figure 1. Price wise, the GeForce 960 priced at \$220 is in the medium level between the high end, super expensive Tesla K40 cards (priced at \$2,600 US) and low end older GeForce models such as GeForce 460 (priced at \$120) with fewer and slower processors and smaller memory.

b. Software requirements

After obtaining the right hardware, a medium form factor desktop computer such as Dell XPS 8500 or equivalent computer, a GPU such as the GeForce 960 or above, you can directly and easily install the GPU and replace the already installed generic graphic card. Install the hardware into the desktop computer with the proper precautions such as discharging any electrostatic charges off your hands and trying not to apply too much pressure while installing the GPU into the proper slot, which could lead to damage to the motherboard and off course a malfunctioning GPU.

The GPU card comes with an installation DVD to install the card drivers, however, it is recommended to go to the www.nvidia.com website and download the latest GPU driver software. This way you are sure of getting the best compatibility and bug free driver software. As of the time of writing this manual, the latest GeForce driver is version 361.91. The installation of the latest GPU driver is a straight forward process, just follow the installation instructions and it will take no longer than a few minutes.

Regarding the installation of the GPU Dynamic 5.0 solver executable, there are two ways to do it: directly by installing the latest CMLAir8.4 which includes GPU Dynamic 5.0, or via downloading the solver from the CML website along with one required run time Cuda FORTRAN DLL file.

If you would like to try a different GPU than GeForce 960, please let Dolf know. He will make sure the correct compiled GPU Dynamic 5.0 fits the CC requirements for your GPU.

Test results on example sliders

In this section we discuss the example sliders used and the results and corresponding speed improvements obtained by using the GPU Dynamic 5.0 solver.

Testing the new GPU Dynamic code version 5.0 was a substantial task. So we tried to use multiple sliders, both simple and complex slider designs, to ensure variety and accuracy of results. All the sliders have been tested with high complexity GPUs. In this section we show 2 sliders as examples for the test. They are either sliders borrowed from CML students or from CMLAir users in the hard disk industry. All sliders under discussion have been tested on a GeForce 960. For each example slider, we show the rail design in addition to the corresponding speed performance.

Example Sliders: Slider 1

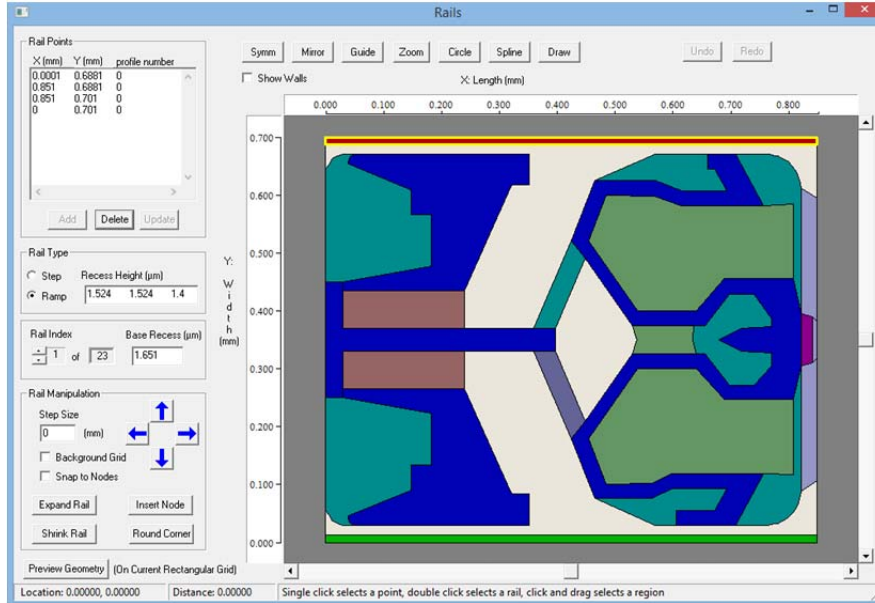


Table 2. GeForce GPU solver vs Dynamic 4 solver

Dynamic Solver	Dynamic 5	Dynamic 4	Speed increase
Execution time	20 minutes	29 minutes	31%

Figure 2a. CMLAir example slider 1 rails

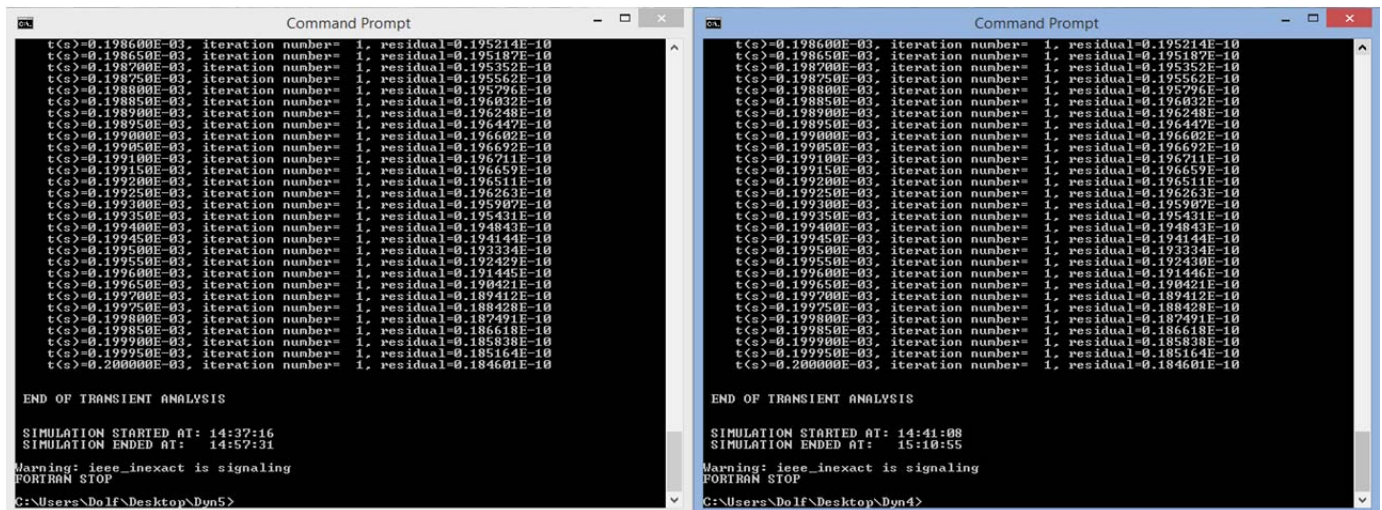


Figure 2b. Dynamic 5 output vs Dynamic 4 output, slider 1 case.

Slider 2

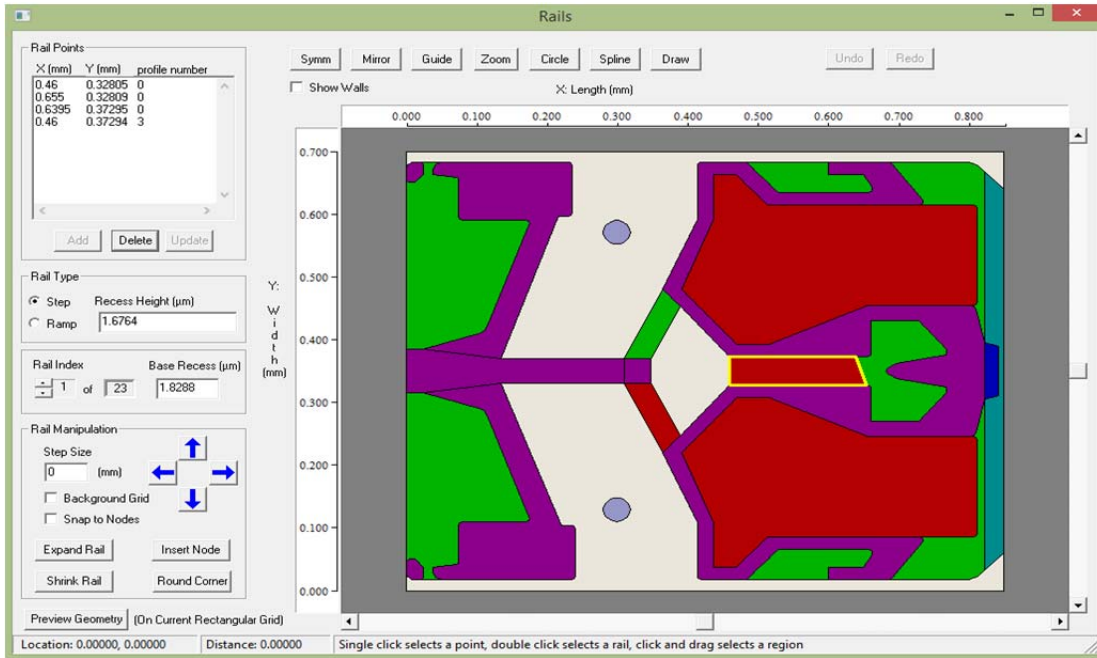


Figure 3a. CMLAir Slider 2 rails

Dynamic Solver	GeForce 960 GPU	Dynamic 4	Speed increase
Execution time	26 minutes	65 minutes	60 % (2.5X)

Table 3. GeForce 960 GPU solver vs Dynamic 4 solver

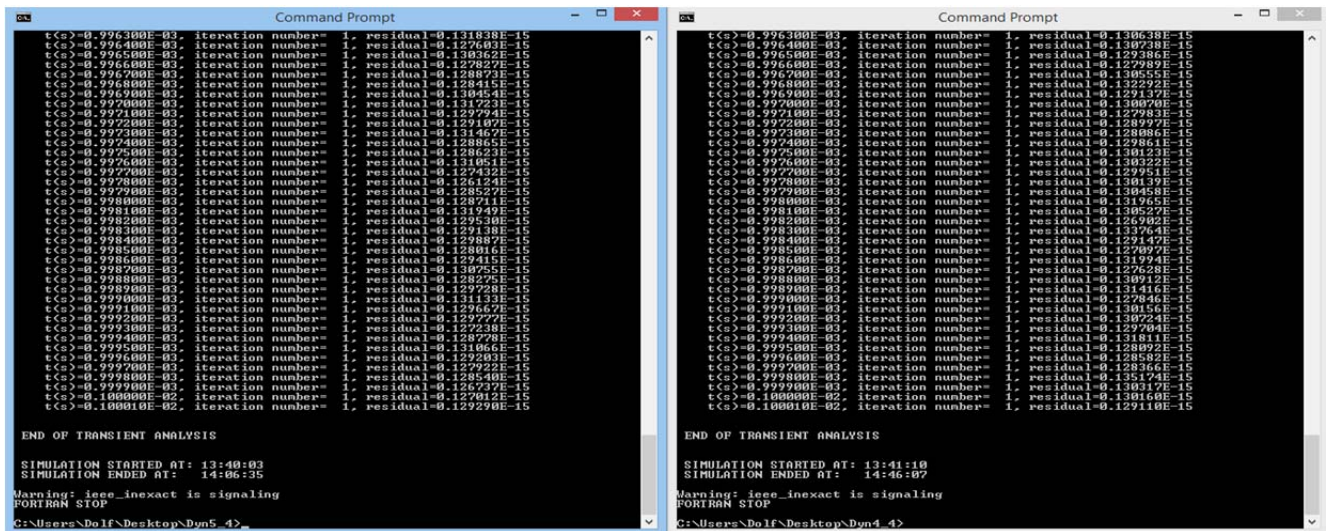


Figure 3b. Dynamic 5 output vs Dynamic 4 output, slider 2 case.

Comparison between the two slider designs

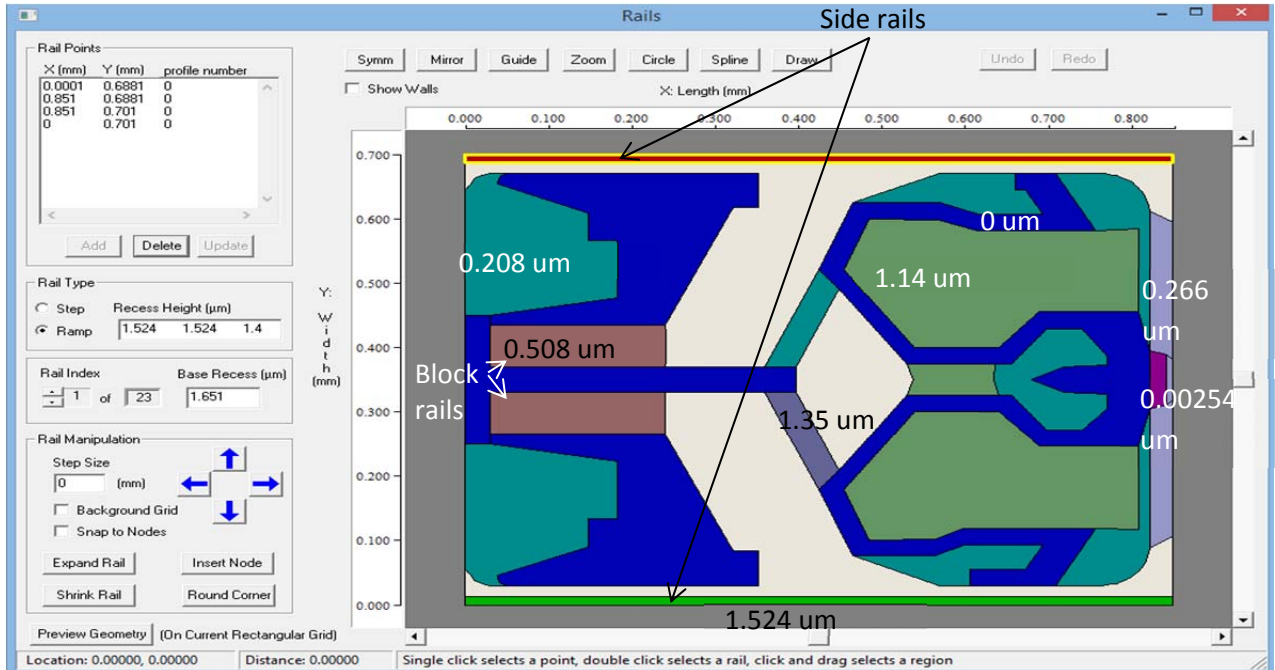


Figure 4a. CMLAir Slider 1 rails with etching depths

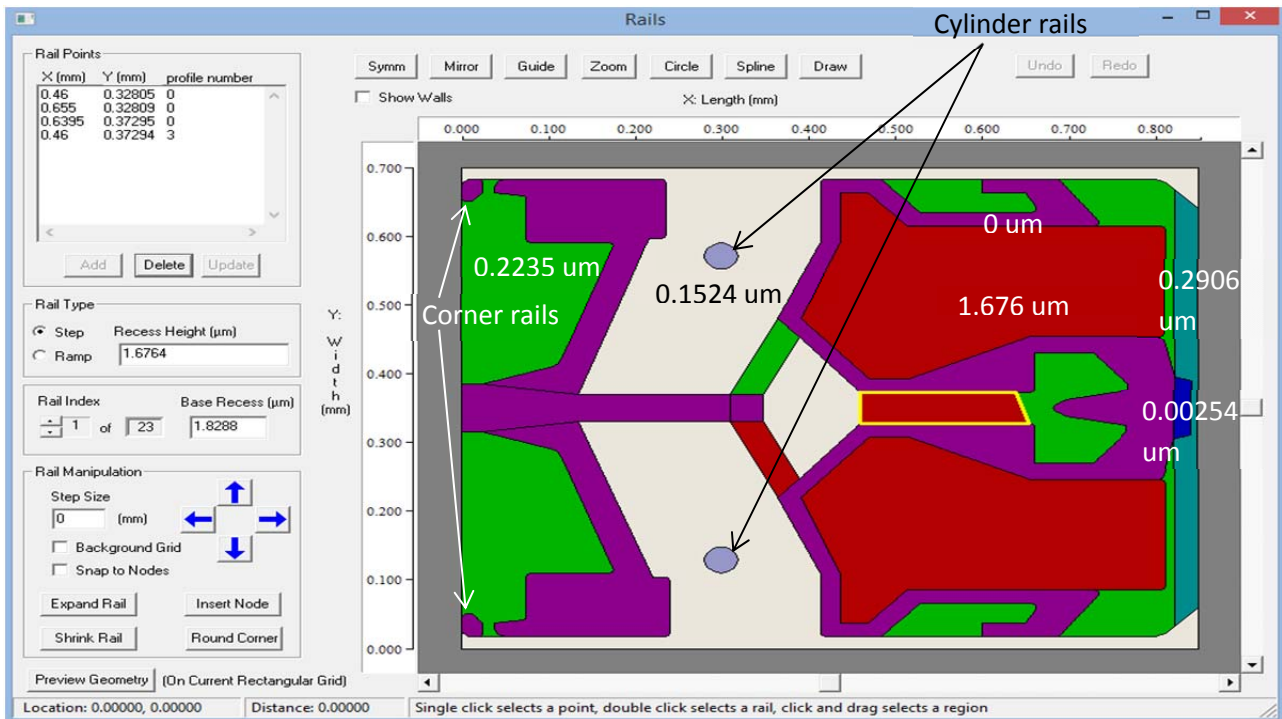


Figure 4b. CMLAir Slider 2 rails with etching depths

Looking at tables 2 and 3, we can see clearly that using GPU Dynamic code gives a much faster speed of convergence. We can also see that there is big difference between speed improvement between table 2 (slider 1 design) compared to table 3 (slider 2 design). In the second case (table 3, slider 2) we have achieved 2.5X speed increase over the traditional Dynamic 4. In order to better understand the reason behind the difference in performance, we decided to examine the two sliders in more detail. We show each slider design along with its recess depth. Figures 4a and 4b illustrate those rail designs.

Slider 1 has two rails that look like a bar along the x axis between the trailing edge and leading edge as shown in figure 4a (side rails). Those two rails are ramp type rails. Slider 2 does not have such rails. Moreover, slider 2 has two rails in a shape of a cylinder as shown in figure 4b (cylinder rails). Slider 1 has no such cylinder type rails. Additionally, slider 1 has block rails as shown in figure 4a near the leading edge, while slider 2 does not have those. Finally, slider 2 has corner rails near the leading edge as shown, but slider 1 does not have such rails.

Each color in each slider corresponds to a rail/rails profile and a certain etching depth as shown. There are differences between etching depths between the two cases. For example, Figure 4a shows the base recess depth of slider 1 to be 1.651 μm , while figure 4b shows a base recess of 1.8288 μm for slider 2. There is a 0.2 μm difference in base recess between the two cases in study. Additionally, the slider 1 design has a rails with an etching depth of 1.14 μm , while slider 2 has equivalent rails but with etching depth of 1.676 μm . There is a 0.536 μm etching depth differences for these rails. Moreover, slider 1 design has a rail with depth of 1.35 μm , while slider 2 has a rail in the same location but with etching depth of 1.676 μm .

The differences in the two discussed slider designs and etching depths above could explain why the speed improvement is not the same in these two sliders. All the above factors participated in making slider two converge much faster than slider 1.

Results discussion and conclusion:

The figures and results are self-explanatory. As can be seen in the first slider, the speed increase over Dynamic 4 is only 33%. Slider 1 design is not as complex as slider 2 design, as we can see in table 2 and table 3. While for other complex sliders, such as slider 2 which has different rails designs and etching depths, the speed increase was 2.5X faster compared with running same slider design using Dynamic 4. For that reason, GPU is recommended for complex slider designs, which are the common case in the new hard disk drives.

Accuracy is an essential factor when looking at the results. Figures 2b and 3b show that the accuracy of GPU Dynamic 5 was as close and comparable to results of Dynamic 4 as possible. There is almost a negligible difference in residuals between the two slider cases under study.

If you are not decided to buy a relatively expensive \$2,200 Tesla K40 GPU along with minimum of \$1,500 high power tower desktop with multiple PCI express slots, our recommendation is to buy the more affordable, \$220 GeForce 960 GPU. It does the job well without the higher price of Tesla K40.

In conclusion, as we see in the example cases along with their output results, the GPU Dynamic 5.0 is a significantly faster solution for complex modern slider designs.