Numerical simulation of electromagnetic flowmeter on GPU

Julia Michailova, Pavel Frolov
State Research Center "NIITeplopribor"
pr. Mira 95, Moscow, Russia, 129085
Phone: +7-495-6165762, FAX: +7-495-615-78-00, E-Mail: datchik@online.ru, pavel.frolov@gmail.com

Abstract: The role of numerical simulations in new device development and optimization sensor properties is steadily increasing. Last years we observe dramatic increase of computational power of GPU based chips with double point precision. With using CUDA architecture process of adapting old numeric code for modern parallel architecture is greatly simplified. In contrast to the modern universal CPU chips graphics accelerators are designed for parallel computing with a large number of arithmetic operations. Steadily increasing number of transistors on GPU works for its intended purpose - processing of data sets. As a result, the basis for the effective use of power of the GPU in scientific is the parallel algorithms for hundreds of execution units available in video chips. In addition, the use of the multiple GPUs greatly increases the processing power of the system at relatively low cost.
Authors developed a method of describing the induced electromagnetic field in the channel makes it possible to evaluate the influence of various factors: the extent and heterogeneity of the magnetic field, the spatial distribution of the field coil, the conductivity of the wall and fluid, the thickness of the pipe on the sensitivity of the flowmeter and the linearity of its properties. Previously, similar problems were solved analytically by setting the magnetic field distribution approximation of functions, for example, infinitely extended and homogeneous, or decaying exponentially, or a Fourier series with idealized coefficients, etc. Then obtained approximate theoretical results were refined through pilot studies breadboard and prototype devices. The increasing availability of high-performance resources can get more accurate results in less time. In this paper we use graphics cards for the parallel calculation of the magnetic field given by approximate functions. The accuracy of calculations depends on the amount of these approximate functions. They can be calculated independently of each other, which is very well suited for the parallel architecture of modern graphics accelerators. .

Keyword: Electromagnetic flowmeter, CUDA, GPU, numeric simulations

1. Introduction

An important competitive advantage in the development of modern devices is to reduce the time required to develop and test prototypes of complex physical structures. If we can calculate in advance the optimum design of the device, assess the limitations of its use without performing expensive and time-consuming experiments with a real device, we benefit from development time as well as reduce costs. Reduced time for numerical simulation of device prototypes requires the use of modern supercomputer technology. Architecture of modern graphics accelerators is increasingly adapting to the demands of modern supercomputer calculations. In the latest generation of graphics accelerators NVIDIA Corporation added support for double precision calculations, and with the new NVIDIA Fermi architecture performance of double point calculations is increased by 6-8 times, so the graphics accelerator of the last generation can ensure peak double precision floating point performance in the range of 2.1 - 2.5 TFlops [1]. The fast productivity increase of graphics accelerators greatly exceeds the performance of the CPU. In favor of using graphics accelerators is supercomputer calculations also says the fact that energy efficiency and the cost per GFlop in graphics accelerators are much lower. It is convenient for small research groups because it does not need to worry about complex cooling systems and requires less energy for a cluster installation of similar performance.

Architecture of graphic accelerators, originally designed for parallel processing of graphics primitives, proved to be very convenient solution for parallel computing. It is now being used for wide range of physical simulations, mathematical and biological problems. A number of CUDA applications was developed for CFD [2-4] and MHD [5] problems as well as for electromagnetic simulations. The usual way of solving these problems is the use of such algorithms as finite element method, finite volume method, finite difference method and particle methods. In our simulation we calculate volumetric weight function that has singularity in electrode locations. This reason can lead to numerical instabilities in the above methods. Therefore, we have developed an algorithm to calculate the volumetric weight function, which uses its representation as an infinite sum of Bessel functions combinations. Bessel functions are replaced by approximations and the resulting series summed. The accuracy of the method determined by the number of the series members taken into account. Such approach overcomes instabilities because sum operation is stable. Also, this approach can benefit from parallel calculation because each member of the series is independent of other series members.

In the State Research Center "NIITeplopribor" installed system for numerical simulation and visualization device prototypes and systems for energy generation facilities based on cluster computing technology with the GPU accelerators NVIDIA Tesla and virtual environments. The system is designed to address the following challenges: mathematical modeling of measurements and monitoring, including flow sensors liquid metal coolant, the visualization of models and systems to display monitoring information using virtual environment technologies. The uniqueness of the installation is that the first time in practice we have in one place a supercomputer cluster, built on GPU, and a virtual environment system, designed to display the results of modeling in real time. We used these cluster system in our electromagnetic flowmeter simulations.

2. Electromagnetic flowmeter simulation.

The electromagnetic flow measurement plays important role in many technological processes. The work described here involves the application of our supercomputer installation to improve accuracy of device prototypes.

First electromagnetic flowmeter was introduced by Faraday in 1831 measuring current from the flow of the Thames in the Earth's magnetic field. The principle of electromagnetic flowmeter is based on the Lorentz force deflection of moving charges in a magnetic field. As shown in Shercliff's book [6] distribution of potential in the electromagnetic flowmeter can be described by the following equation:

$$\Delta\varphi = \mathrm{div}[\mathbf{v}\times\mathbf{B}], \qquad (1)$$

where **v** is the flow velocity and **B** is the magnetic flux density vector. This is a Poisson equation with given right-hand side. This equation can be solved for analytically for simple cases, if you set of corresponding boundary conditions. Shercliff suggested using a weighting function W, which describes the contribution of different parts of the flow in the output signal. Bevir showed[7] that the electromagnetic flowmeter signal U can be expressed through the weighting function as follows:

$$U = \int_{\tau} \mathbf{v} \cdot \mathbf{W} \, d\tau \qquad (2)$$

where τ is channel volume, **W** is volumetric weight function and **v** is the flow velocity. Volumetric weight function can be represented as a function that characterizes the contribution to the signal a flow part of the stream flowing through the elementary volumes at different points in the working channel volume.

In this paper we solve the following problem. We have infinite tube with liquid with isolating walls. So that, boundary condition for equation (1) follows from the vanishing of the normal current through the inner surface of the wall and has the form $\left.\frac{\partial \varphi}{\partial r}\right|_{r=R} = 0$. In addition, it is natural to assume that at an infinite distance from the source of the magnetic field $\varphi \to 0$, which allows you to safely use further the Fourier transform for the electrostatic potential. We use a polar coordinate system, so $x = r\cos\theta, y = r\sin\theta$, z axis is directed along the tube axis, and r is the distance from the center of the pipe. As the result, the potential difference across the electrodes can be provided in the following form

$$\mathrm{U} = \varphi(\mathrm{r}=\mathrm{R}, \theta = \pi/2) - \varphi(\mathrm{r}=\mathrm{R}, \theta = -\pi/2) = \int_{-\infty}^{\infty} \mathrm{dz} \int_{0}^{\mathrm{R}} \mathrm{rdr} \int_{0}^{2\pi} \mathrm{d}\theta \, \mathrm{g} \, \mathrm{div}[\mathbf{v} \times \mathbf{B}] = \int_{\tau} \mathbf{v} \cdot [\mathbf{B} \times \frac{\partial \mathrm{G}}{\partial \mathbf{r}}] \mathrm{d}\tau \qquad (3)$$

where $G = \int_{-\infty}^{\infty} \cos kz \, dk \sum_{\substack{m=0,\\ n=2m+1}}^{\infty} (-1)^m \frac{\mathrm{I}_n(kr)}{r \, \mathrm{I}_n'(kR)} \sin n\theta$ is the difference of the Green function for Laplace equation derived from (1) and we can denote $\mathbf{w} = \frac{\partial G}{\partial \mathbf{r}}$ as volumetric weighted function, and $\mathrm{I}_n(x)$ is modified Bessel function. By using asymptotic representation of Bessel functions from [8]

$$I_n(x) \cong \exp\left[s/n - arcth\left(\frac{n}{s}\right) \right] \frac{G(s)}{\sqrt{2\pi s}} \qquad (4)$$

where $s = \sqrt{n^2 + x^2}$, and $G(s) = 1 + \frac{1}{8}\left(\frac{1}{s} - \frac{5n^2}{3s^3}\right) + \frac{1}{8^2}\frac{3}{s^3}\left(\frac{3}{2} - \frac{77n^2}{9s^2} + \frac{285n^4}{54s^4}\right) + \ldots$ is the Debye series, one can deduce the following sum for volumetric weighted function:

$$w_x = \sum_{\substack{p=0,\\ n=2p+1}}^{\infty} (-1)^p \frac{\left(\frac{r}{R}\right)^n n^{2/3} e^{-\frac{nz}{R}} \cos(n\theta - \theta)}{r}. \qquad (5)$$

This expression represents approximation of volumetric weighted function as infinite series with independent members. Such representation allows us to use parallel computing algorithms and parallel hardware, in particular CUDA, for faster and more precise estimation of the volumetric weighting function.

3. CUDA Technology

The main API used for parallel computing on NVidia Tesla is CUDA (Compute Unified Device Architecture). CUDA parallel architecture constructed as follows. The structure of GPU comes with a few clusters of texture blocks (Texture Processing Cluster). Each cluster consists of a consolidated block of texture samples and two or three streaming multiprocessors, which in turn include eight computing devices and two functional blocks. For instruction execution SIMD principle is used, only one instruction is executed once for all threads in the warp (a group of 32 threads - the minimum amount of data processed by multiprocessors). This method of execution was called SIMT (single instruction multiple threads - one statement for multiple threads). In CUDA, there are several types of memory: shared, constant, global and texture. Each multiprocessor has access to the shared memory size 16Kbyte. This memory is not cached, but provides the fastest access to resources. Each multiprocessor has 8Kbate cache memory for constant and texture data. Flows of one block are always executed on a single multiprocessor. Flows of different blocks can not communicate with each other, because they can be different multiprocessors. Multiprocessors impose a large number of registers (16384 for GT2xx architecture) common to all threads of all blocks that are executed on the multiprocessor. The architecture of the GT200 could run up to 1024 threads on multiprocessor, which is organized in 10 clusters of three multiprocessors, processing up to 30720 streams. Thus, in the architecture Tesla has 30 multiprocessors.

CUDA programming is based on the creation of cores (kernel), which run on multiprocessors. Accelerator gets the kernel and creates a copy of each data element inside kernel to run thread. Thread also contains counters, registers and states. Threads are executed in groups of 32 pieces, called warp. Warp run on certain multiprocessors, each of which consists of eight cores of stream processors that can execute one MAD instruction per clock. Multiprocessor perfectly suited for multi-threading. With every clock cycle scheduler chooses warp to process, and switch between them without loss of time. If we talk in terms of CPU, this is similar to the simultaneous execution of 32 programs and run them without losing context switch. NVIDIA also provides nvcc compiler with special extensions of C++ language to smoothly incorporate CUDA kernels in the existing code. This greatly reduces time to develop numerical applications.

4. Algorithm implementation

The implementation of the algorithm on CUDA is straight forward. For the selected volume of flowmeter tube around the electrodes we created a 256x256x256 grid, where volumetric weighted function was estimated. As each series member in (5) is independent of other members we done each series member calculations per one CUDA thread. Next, sum of the values from each thread in each block was calculated. And the resulting sum is calculated on CPU with loop over all blocks.

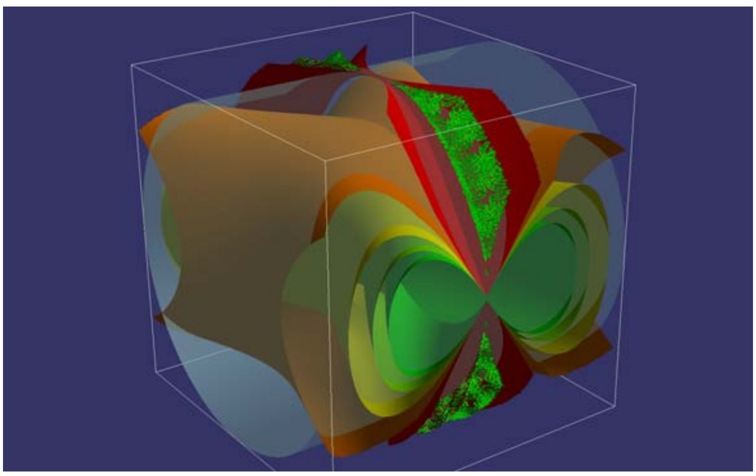

Fig. 1 The volumetric weighted function visualization

We have used OpenSceneGraph scene engine[9] with additional module for isosurface reconstruction with Marching Cubes algorithm for results visualization on VR system. Figure 1 shows the visualization of w_x calculated by the algorithm.

5. Discussions

Simulation of the electromagnetic flowmeter with the use of CUDA can provide two main advantages either we significantly reduce time or we improve accuracy for volumetric weighted function calculations. The middle point of such extreme cases could be trade off between speed and accuracy, we get maximum accuracy in appropriate time.

The performance comparison tests are conducted in three configurations on the same machine Core i7-940 with NVIDIA Quadro FX 5800 board. The three configuration are: one threaded application on CPU, four threaded application on CPU and GPU application. One threaded case implemented on C and complied with optimization flags. We used OpenMP API for forur threaded application and CUDA API for GPU application. According to results shown in Table 1, the GPU application gain over 10x performance increase in comparison with one threaded application on CPU.

Table 1 Performance results

	One threaded on CPU	Four threaded on CPU	GPU
Time, min	183	54	14

This results show that our approach is well suited for parallel computations. It is also scalable for supercomputer clusters. Next step in research will be implementation of our algorithm in cluster system with GPU accelerators. It is possible to use parallel CPU and GPU calculations on such system that will further accelerate numerical simulation of electromagnetic flowmeter.

6. Conclusion

We have presented the algorithm for parallel simulation of electromagnetic flowmeter based on analytical representation of volumetric weighted function as infinite series of approximation-function. Using such method it is possible to evaluate the influence of various factors: the extent and heterogeneity of the magnetic field, spatial distribution of the coil field, the conductivity of the wall and fluid and so on. Our method is well suited for parallel computations and gives good acceleration on GPU chips. By using parallel computations we reduce time required to design new device prototypes that gives competitive advantage in the development of modern devices. Also using supercomputer cluster installation we can improve accuracy of device modeling by taking into account more approximate functions in the series.

Acknowledgment

The authors wishes to thank Ogurtsov A.I. the head of IT department for their support and help in conducting this research. The authors are also indebted to the Russian State Research Center "NIITeplopribor" for his presence at this Conference.

References

[1] Nvidia Tesla S2050/S2070 specification,
http://www.nvidia.com/object/product_tesla_S2050_ S2070_us.html

[2] D. Göddeke, S. H.M. Buijssen, H. Wobker, and S. Turek., GPU acceleration of an unmodified parallel finite element Navier-Stokes solver," High Performance Computing & Simulation 2009 (Waleed W. Smari and John P. McIntire, eds.), pp. 12-21.

[3] Andrew Corrigan, Fernando Camelli, Rainald Loehner, John Wallin, Running Unstructured Grid CFD Solvers on Modern Graphics Hardware, 19th AIAA Computational Fluid Dynamics Conference, June 2009, paper no: AIAA 2009-4001

[4] Julien C. Thibault and Inanc Senocak, CUDAImplementation of a Navier-Stokes Solver on Multi-GPUDesktop Platforms for Incompressible Flows, 47th AIAAAerospace Sciences Meeting, Orlando FL, paper no:AIAA-2009-758.

[5] Wong, H.-C., Wong, U.-H., Feng, X., Tang, Z., Efficient magnetohydrodynamic simulations on graphics processing units with CUDA, 2009, arXiv:0908.4362

[6] Shercliff J.A., The theory of electromagnetic flowmeasurement, Cambridge University Press, 1962.

[7] Bevir M.K., The theory of induced voltage electromagnetic flowmeters, J. Fluid Mech. 43, №3, 1970, p. 577-590

[8] Jahnke, E.; Emde, F.; Lösch, F., Tafeln höherer Funktionen, Teubner, Stuttgart 1960

[9] OpenSceneGraph engine, http://www.openscenegraph.org