DOI: 10.31401/WS.2020.proc.13

Using of Parallel Computing on Gpus for MHD Modeling of Solar Flare Behavior in Real Time Scale

Borisenko A.V.[1], Podgorny I.M.[2], Podgorny A.I[1]

[1]Lebedev Physical Institute of the RAS, Moscow, Russia
[2]Institute of Astronomy of the RAS, Moscow, Russia
E-mail: sunw77@mail.ru

Abstract.

The forecast of solar flares and the appearance of solar cosmic rays (SCR) and other solar flare manifestations is an important scientific and technical task. Magnetohydrodynamic (MHD) modeling in corona above the active region allows us to find the appearance of current sheets in the solar atmosphere which are sources of magnetic energy of flares. In the conditions of complicated configuration of magnetic field in the solar corona the appeared current sheets can be found using specially developed graphical system of search basing of the founding of local current density maximums. MHD simulation in strongly reduced time scale showed the appearance of numerical instability near the photospheric boundary due to unnaturally fast magnetic field increase on the photosphere. To avoid the large number of current density maximums caused by instability near the photospheric boundary, it is necessary to perform MHD simulation in corona in the real scale of time. MHD simulation in real time scale on the ordinary computer took several years. The results of MHD simulation in real scale of time obtained using parallel computing on Nvidia Tesla GPUs showed acceleration of calculations more 100x speed up. The time of calculation of was strongly reduced due to the correct choice of computing equipment and software and using the capabilities of the selected equipment for parallelizing the calculation. The first results of the GPU MHD simulation of flare situation in real time scale for the AR 10365 showed the appearance of current density maximums with X-type configurations of magnetic field and with a plasma flow which cause to the formation of a current sheet. The simulation in real time scale confirmed the solar flare mechanism based on release of energy accumulated in magnetic field of current sheet.

Introduction. The need to apply parallel computations for MHD simulation in the solar corona.

In view of the appearance of solar flares high in the corona, as well as the impossibility of obtaining from observations the configuration of the magnetic field in the corona, in order to study the mechanism of flares and their prognosis, MHD simulation in the corona above the active region (AR) is necessary [*Podgorny et al.*, 2020]. Flares appear as a result of the release of magnetic energy accumulated in the field of a current sheet, which is formed in the vicinity of a X-type singular line of magnetic field [*Podgorny and Podgorny, 2012*]. For the magnetic field found as a result of MHD simulation, in a complex configuration above the AR, the position of the current sheet can be determined only with the help of a specially developed graphical search system [*Podgorny et al., 2017; Podgorny et al., 2018*]. The search system is based on determining the positions of local maxima of the current density, in which the configuration of the magnetic field and plasma flow is analyzed. MHD simulation on the usual computer (Intel (R) Core (TM) i7 CPU 920 @ 2.67GHz) was only carried out in a 10^4 times reduced time scale. Due to the appearance of instability at the photospheric boundary, caused by an unnaturally rapid change in the magnetic field, a large number of fictitious current density peaks appeared, masking the current sheets, which made it difficult to use the graphical search system. To get rid of a large number of current density maxima masking current sheets, as well as to establish the evolution of processes in time, it is necessary to perform MHD simulation in real time.

Calculations have shown that using a specially developed difference scheme [*Podgorny and Podgorny 2008, 2004; Podgorny et al*., 2020], MHD simulation in real time, requires that the time step does not exceed 4×10^{-7} days in the main part of the calculated interval. Such simulation cannot be carried out on the usual computer - it would require 8 years of calculation. It becomes necessary to use a 'home supercomputer' with many computational threads, for parallelizing the numerical solution of the used finite-difference scheme.

Selection of computing equipment and software and the use of their capabilities to optimize parallelization of computations

To use parallel computations, the optimal parameters of the finite-difference scheme were selected and a number of optimizations of the parallel computation algorithm were carried out, primarily related to the exchange of values specified in the computational domain between the main memory of the computer and the memory of the graphics card (GPU), [*Podgorny et al*., 2020].

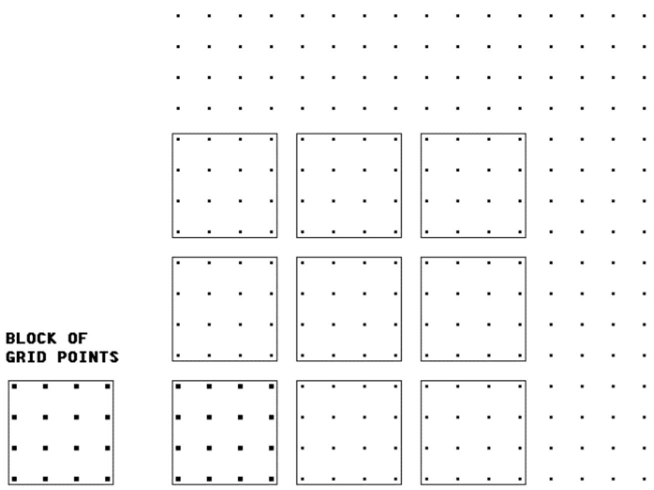

Fig. 1. Coverage of a two-dimensional grid with two-dimensional blocks of points, each of which corresponds to a computational thread.

Parallel computations were carried out using CUDA technology on graphic cards, the computational threads of which carry out parallel computations. The problem was set on 'home supercomputers' rented in the clouds with modern graphics cards. For the calculation, we chose graphic cards with a high computation speed for each thread. The installed software provided additional opportunities for optimizing the parallelization algorithm.

To parallelize computations, modern graphics cards (GPU) Titan-V (Volta), V100 (Volta-100), P100 (Pascal-100) were used. The use of the modern language Fortran PGI (Portland Group - Fortran, created specifically for parallelization using graphics cards GPU) made it possible to carry out parallel computation at points of a spatial grid in sequentially selected three-dimensional blocks of threads. To calculate the boundary conditions in the planes bounding the computational domain, two-dimensional blocks of threads were used. An example of covering a two-dimensional grid with two-dimensional flow blocks is shown in Fig. 1, covering the 3D mesh with 3D blocks is similar.

In the majority of calculations 3D blocks were choses $NX\times NY\times NZ = 4\times4\times4 = 64$ points in block (64 parallel calculating threads, each find values it its point). We also took blocks of

8×4×4 = 128, 8×8×4 = 256 and more, but this did not lead to an increase in the calculation speed (even led to a slight slowdown in the calculation; a very large number of points in a block led to a significant deceleration). So, so far, we have not managed to increase the calculation speed by 1024 or even 512 times, as we hoped, since so many computational threads physically exist on the graphics card. Now the calculation speed has been increased by parallelizing calculations by 120 times. Experts explain the reason in the waste of time for exchanging data between memory registers.

Also, the use of the selected computing equipment and software made it possible to transfer a large number (~ 100) of parameters to subroutines that carry out parallelization on graphics cards.

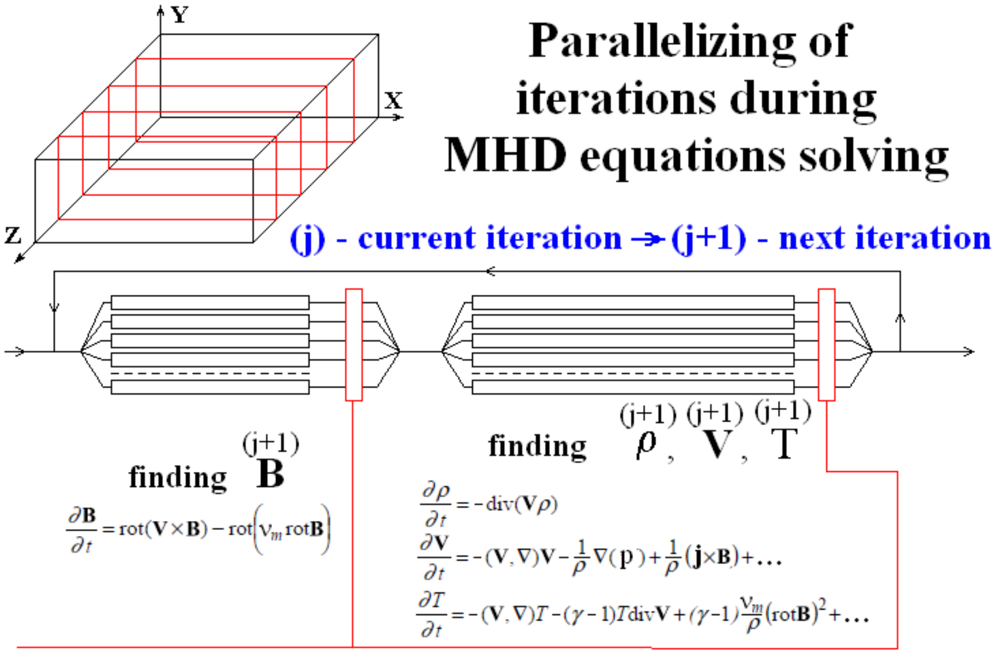

Fig. 2. Parallelization by OpenMP method. Each thread calculates in its own subregion.

The earlier use of the M2050 graphics card made it possible to understand the procedure for parallelizing computations for such difference schemes as the developed scheme for the numerical solution of MHD equations solved by the iteration method using the example of solving the Poisson equation (iterations according to the same principle as for MHD equations, but very much simplified). However, parallel computing using 3D blocks of threads and passing a large number of parameters to the parallelization routines is not possible for the M2050 graphics card. To solve the three-dimensional Poisson equation, parallelization was carried out using one-dimensional blocks, for this purpose a single numbering of all points of the computational grid was introduced. The use of this method did not make it possible to numerically solve a complex system of MHD equations on the M2050 graphics card, and on the Titan-V, V100 and P100 graphics cards it slows down the solution by ~ 20%.

Parallelization of the numerical solution of MHD equations by the OpenMP method using the computational threads of the main processor is carried out. Intel Visual Fortran was used. The computational domain was divided into subdomains, so that each thread calculates in its subdomain (Fig. 2). Computations on multiprocessor clusters in the clouds showed a slowdown in computation by a factor of several compared to parallel computations on a GPU using CUDA technology. On graphics cards, parallelization is also carried out, similar to the OpenMP method, when each GPU thread computes in its own subdomain (Fig. 2). As a result of such parallelization on a graphics card, the calculation speed becomes ~ 15 times slower than the calculation speed, according to a program that performs parallel calculations at the points of sequentially selected blocks.

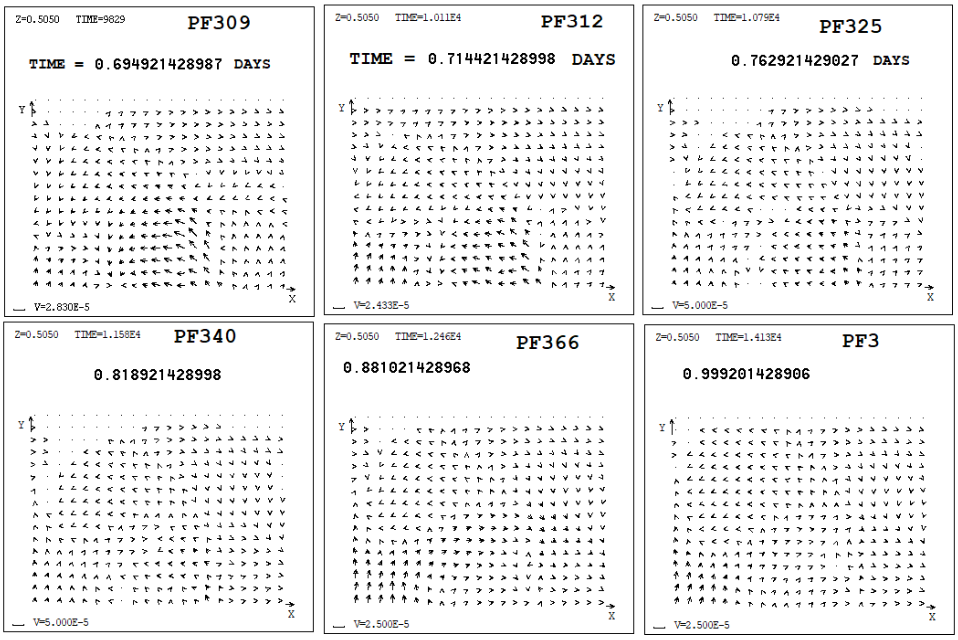

Fig. 4. Evolution of field of plasma velocity in the central plane of computational domain.

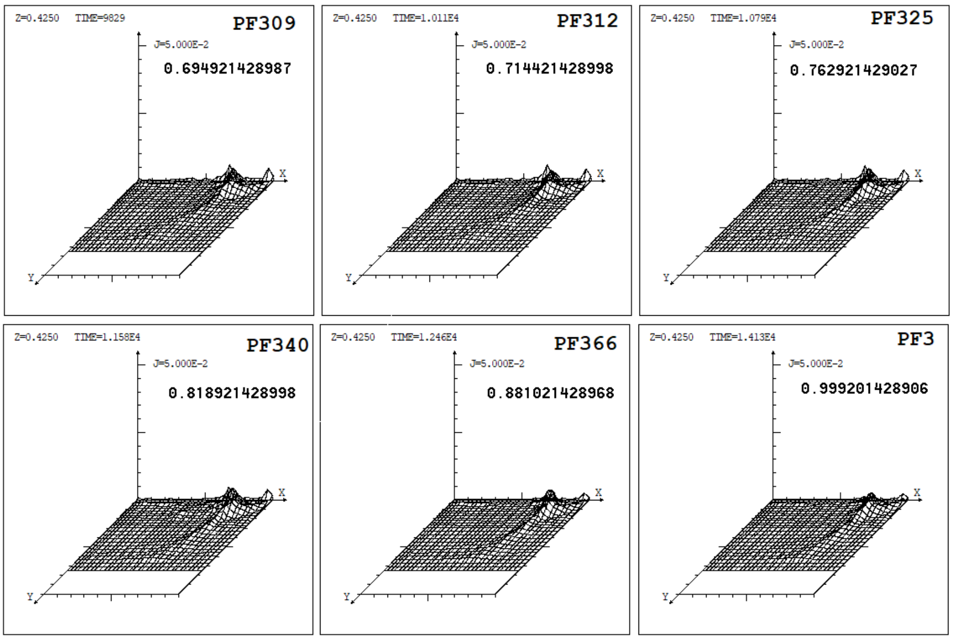

Fig. 5. Evolution of current density distribution.

The presented parallelization methods were used for:

- subroutines that calculate the values at the given iteration at the points of the three-dimensional grid of the computational domain according to the difference scheme used
- calculation at the given iteration (or immediately the final calculation, when possible) of values at the points of the boundary according to the set boundary conditions
- calculating arrays and working with the calculated arrays, necessary to calculate the calculation error at a given iteration in order to determine the end of the iterative process

- additional calculations using arrays of values in the computational domain (in the corners and on the edges of the computational domain and other additional computations) that do not require parallelization and in previous were calculated by the main computer processor (in order not to send arrays on the graphics card to the main memory, calculations are performed by one computational thread on the graphic card).

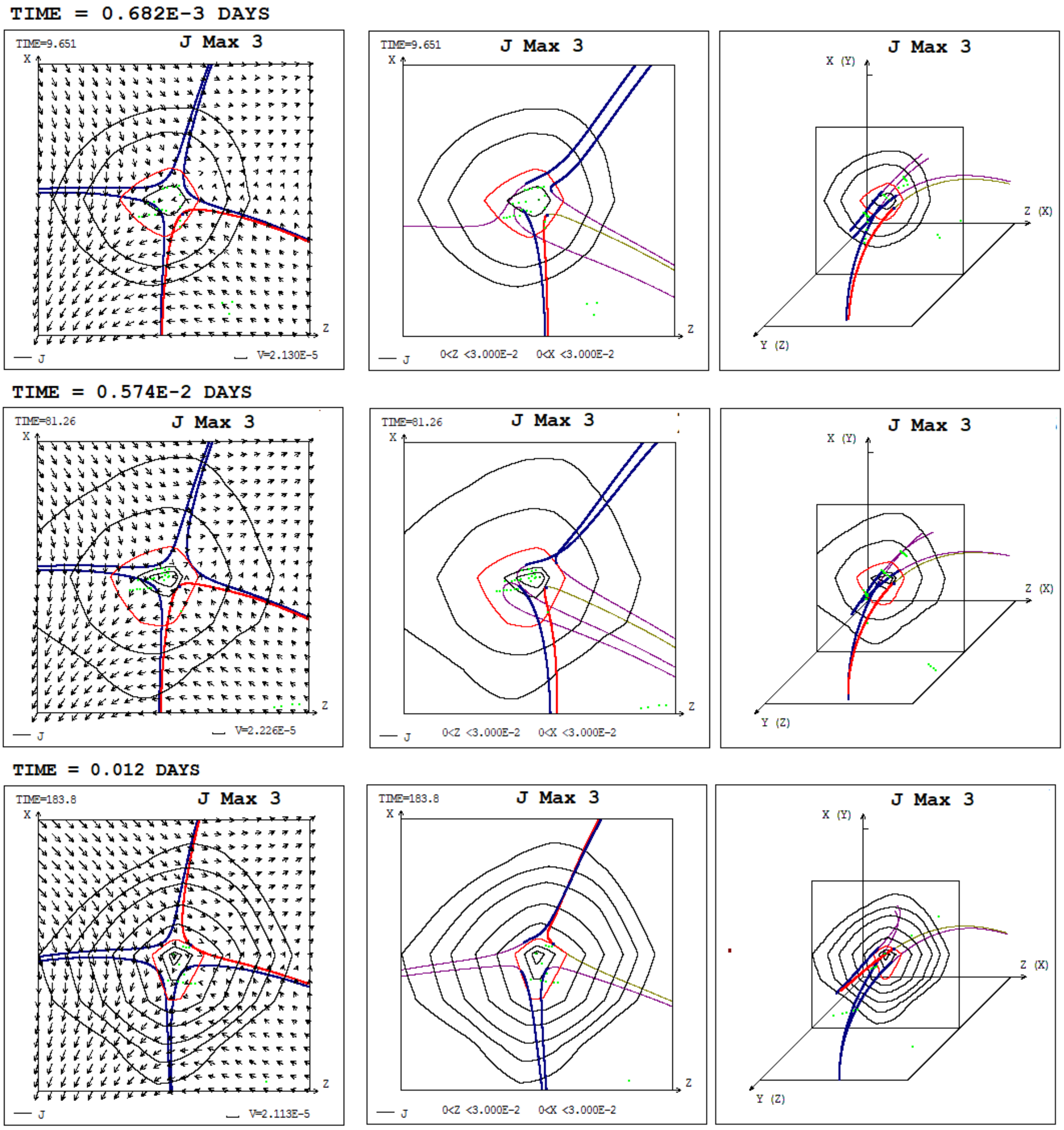

Fig. 6. 2D and 3D magnetic field configurations and plasma flow in current density maximums.

Using the capabilities of the selected computing equipment and software, as well as minimizing the exchange between the GPU memory and the main computer memory [*Podgorny et al*., 2020], reduced the calculation time of iteration to 0.0282 - 0.0302 s. Under these conditions, the optimal choice of the parameters of the difference scheme [*Podgorny et al*., 2020] made it possible to calculate the evolution during one day in ~ 21 days. For the prognosis, this time should be less than a day, it is necessary to work on further optimization, the reserves are existing.

Some results of MHD simulation in the real scale of time above AR 10365 for evolution during the day

The evolution of the plasma velocity and current density in the central plane of computational domain for MHD simulation in the corona above AR 10365 during the first 1.2 days of evolution, when there were no flares yet, is presented in Fig. 4, 5. In some current density maximums, founded by the graphical system of search the magnetic field have X-type configuration and plasma flow should cause the current sheet appearance (Fig. 6).

Conclusion

1. MHD simulation in the corona above the AR in the real scale of time to study the physical mechanism of a solar flare is possible only using parallel computations.
2. The graphics cards and software which perform fast parallel computing, in particular through the use of three-dimensional blocks of computational threads, have been selected.
3. As a result of the application of parallelization methods, which are feasible for the selected computing equipment and software, and optimization of the parallelization algorithm [*Podgorny et al*., 2020], the GPU calculation speed exceeded more 120 times the calculation speed on a serial CPU computer (Intel(R) Core(TM) i7 CPU 920 @ 2.67GHz). Calculation in the real scale of time of the evolution of the field and plasma in the corona within one day became possible in 21 days.
4. MHD simulation in the corona above AR 10365 during the first 1.2 days of evolution showed the appearance of X-type configurations with plasma flow, which should cause the formation of a current sheet.
5. Trial calculations showed the possibility of further increasing the speed of calculations using the latest equipment.

Acknowledgments

The authors are grateful to the SOHO/MDI team for the scientific data provided at http://soi.stanford.edu/magnetic/index5.html, as well as to the many professional cloud service specialists, which made it relatively easy for us to configurate rented remote computers for GPU computing.

References

Podgorny A.I., Podgorny I.M. (2004) MHD Simulation of Phenomena in the Solar Corona by Using an Absolutely Implicit Scheme. Computational Mathematics and Mathematical Physics. 44, 1784-1806.

Podgorny A.I., I.M. Podgorny I.M. (2008) Formation of Several Current Sheets Preceding a Series of Flares above the Active Region AR 0365. Astronomy Reports. 52, 666–675.

Podgorny A.I, I. M. Podgorny I.M. (2012). Magnetohydrodynamic Simulation of a Solar Flare: 1. Current Sheet in the Corona. Geomagnetism and Aeronomy, 52, 150-161.

Podgorny A.I., Podgorny I.M., Meshalkina N.S. (2017) Magnetic field configuration in corona and X-ray sources for the flare from May 27, 2003 at 02:53. Sun and Geosphere. 12, 85-92.

Podgorny A.I., Podgorny I.M., Meshalkina N. S. (2018) Current sheets in corona and X-ray sources for flares above the active region 10365. Journal of Atmospheric and Solar-Terrestrial Physics. 180, 16-25.

Podgorny A.I., Podgorny I.M., Borisenko A.V. (2020). Study of the solar flare mechanism by MHD simulation in the corona above the active region in real time scale. Proc. 12-th Workshop “Solar Influences on the Magnetosphere, Ionosphere and Atmosphere”, Primorsko, Bulgaria, June, 2020. This Issue.