Numerical Modeling of Space Plasma Flows: ASTRONUM-2012
ASP Conference Series, Vol. 474
N. V. Pogorelov, E. Audit, and G. P. Zank, eds.

GPU Computing in Space Weather Modeling

Xueshang Feng, DingKun Zhong, Changqing Xiang, and Yao Zhang

SIGMA Weather Group, State Key Lab of Space Weather, Center for Space Science and Applied Research, Chinese Academy of Sciences, PO Box 8701, Beijing 100190, China

Abstract. Space weather refers to conditions on the Sun and in the solar wind, magnetosphere, ionosphere, and thermosphere that can influence the performance and reliability of space-borne and ground-based technological systems and that affect human life or health. In order to make the real- or faster than real-time numerical prediction of adverse space weather events and their influence on the geospace environment, high-performance computational models are required. The main objective in this article is to explore the application of programmable graphic processing units (GPUs) to the numerical space weather modeling for the study of solar wind background that is a crucial part in the numerical space weather modeling. GPU programming is realized for our Solar-Interplanetary-CESE MHD model (SIP-CESE MHD model) by numerically studying the solar corona/interplanetary solar wind. The global solar wind structures is obtained by the established GPU model with the magnetic field synoptic data as input. The simulated global structures for Carrington rotation 2060 agrees well with solar observations and solar wind measurements from spacecraft near the Earth. The model's implementation of the adaptive-mesh-refinement (AMR) and message passing interface (MPI) enables the full exploitation of the computing power in a heterogeneous CPU/GPU cluster and significantly improves the overall performance. Our initial tests with available hardware show speedups of roughly 5x compared to traditional software implementation. This work presents a novel application of GPU to the space weather study.

1. Introduction and Motivation

Space weather is as variable as terrestrial weather. Like atmospheric science, space science now has an application branch, that fulfills a societal role in serving human activities and mitigating the loss caused by destructive space weather. Like in terrestrial weather forecasting, numerical models are a critical part of any forecasting system.

Corona-interplanetary space, as a linking medium of fully understanding space weather process in the Sun-Earth system, is the unique channel for solar storms to propagate from the Sun to Earth. It involves a large extent, and contains many critical features, such as discontinuities and heliospheric current sheet, that have spatial scales many orders of magnitude smaller than the system size. In particular, the plasma density, the Alfvén velocity, interplanetary magnetic fields and the plasma β vary over many orders of magnitude from the Sun to Earth. This also implies a large variation of the CFL stability limit from corona to interplanetary space. Usually, time scales range from a few seconds near the Sun to the expansion time of the solar wind from the Sun to the Earth ($\sim 10^5$ s). The numerical grids are either adapted to the small

scale features in the system, or a brute force approach is used with as high numerical resolution as possible while fighting the limits of available computational power. For a typical system size of $215 \times 215 \times 215 R_S^3$ at a resolution of $\sim 0.057 R_S$ (comparable to the thickness of heliospheric current sheets since the existence of a wide range in HCS width at 1 AU, is typically agreed to be from 40,000 to 100,000 km (Behannon et al. 1981)), one would need times of 10^{10} grid cells. The increase of grid points in 3D simulations costs both memory and computing time, e.g., increasing the resolution by a factor of 10 in all directions requires that the time stepping is also made 10 times more frequent. In total this means a factor of 10000, which is the difference between 1 s and 3 hr in computing time. A simple Cartesian mesh would grossly under-resolve much of the problem, while over-resolving relatively uninteresting regions. These problems need solution-adaptive schemes, which enable us to define a different grid spacing in different parts of the Sun-to-Earth space, achieving the best resolution where mostly needed (Feng et al. 2010, 2011, 2012a). This adaptation should also be made dynamic; e.g., to follow the position of helioshpheric current sheet.

Such highly parallelized computations of solar wind modeling requiring high arithmetic intensity is extremely well suited for running on the graphics processing units (GPUs). Specifically, the GPU addresses problems that can be expressed as data-parallel computations-the same program is executed on many data elements in parallel-with high arithmetic intensity-the ratio of arithmetic operations to memory operations. Because the same program is executed for each data element, there is a lower requirement for sophisticated flow control, and because it is executed on many data elements and has high arithmetic intensity, the memory access latency can be hidden with calculations instead of big data caches. The main difference between GPUs and central processing units (CPUs) are that GPUs devote proportionally more transistors to arithmetic logic units and less to caches and flow control in comparison to CPUs. GPUs also typically have higher memory bandwidth that substantially outpaces its CPU counterpart. As a result, GPUs have become an alternative parallel computing device for high performance scientific computing (Owens et al. 2007; Che et al. 2008; Schenk et al. 2008). There is an increasing number of research on using GPUs for scientific applications to both fluid and MHD simulations (Belleman et al. 2008; Gaburov et al. 2009; Schive et al. 2010; Wang et al. 2010; Wong et al. 2011; DeLeon and Senocak 2012; Lin et al. 2012). These studies achieved an overall speedup compared to a single core on the host computer. However, to date, there are few studies on the application of GPU for solving the full MHD equations for space weather modeling in solar-terrestrial space. The objective of the present paper is to transfer our SIP-AMR-CESE MHD model (Feng et al. 2012a) to GPU platform.

2. The 3D SIP-AMR-CESE MHD Model

Feng et al. (2012a) and references therein presented the details of the AMR implementation of the SIP–CESE MHD model on the six-component grid system of the spherical-shell domain in solar-terrestrial space. AMR implementation is carried out in the associated reference component space (ξ, η, ζ) of a rectangular box in a logically Cartesian coordinate, and the PARAMESH package available at `http://sourceforge.net/projects/paramesh/` decomposes every reference component (*i.e.*, our computational space) into many blocks of the same size, and organizes all the blocks in the whole computational space (patched by the six reference components) into an oct-tree

structure. Both the solution variables and their first-order derivatives ($\hat{\mathbf{U}}, \hat{\mathbf{U}}_\xi, \hat{\mathbf{U}}_\eta, \hat{\mathbf{U}}_\zeta$) are stored at each solution point, which have the number of variables $\texttt{nvf} = 4 \times 8 = 32$.

Initially, the computational domain in every reference component is divided into 224 blocks with each block consisting of $8 \times 8 \times 8$ cells with one layer of guard cells included. Totally we have 224 blocks for each component. That is, 6×224 blocks for the whole computational domain. With the help of PARAMESH according to the refinement strategy of the curl of the magnetic field (e.g., Feng et al. (2012a) and reference therein), three levels of grid refinement are used to obtain a grid cell size of 0.012Rs on the solar surface. The grid throughout the simulation is refined to obtain a grid cell size of about 0.16 Rs with three levels of grid refinement near the current sheet within 20 Rs and it is about 0.55 Rs near 1 AU with five levels of grid refinement. The maximum grid cell size is about 1.31 Rs in the corona and about 7 Rs in the inner heliosphere. And the most densely refined component can have about 8800 blocks.

To start the computation, the Parker's well-known hydrodynamic solar wind solution is used to give the initial values of the plasma density, ρ, gas pressure p, and the plasma velocity $\mathbf{u}$. Here, the initial temperature and number density on the solar surface are prescribed to be 1.3×10^6 K and 1.5×10^8 cm^{-3}, respectively.

3. OpenCL Programming Model and Performance

Besides the Compute Unified Device Architecture (CUDA) (NVIDIA CUDA Programming Guide 2.2.1, May 2009, `http://www.nvidia.com/object/cuda_home_new.html`), Open Computing Language (OpenCL) is another open royalty-free standard for general purpose parallel programming across CPUs, GPUs and other processors, giving software developers portable and efficient access to the power of these heterogeneous processing platforms (`http://www.khronos.org/opencl/`). This section describes OpenCL programming model and performance of our SIP-AMR-CESE model.

Figure 1a shows the whole procedure implementation of 3D SIP-CESE MODEL with block-structured adaptive mesh refinement on multi-GPU clusters using OpenCL. Noticeably, we also combine OpenCL with MPI to run the code. In the framework, since the CESE solver is computationally dominant, which takes up 94% time during one time-step, in this work we consider the mapping of CESE solver onto GPU while leaving the creation and refinement of the grid hierarchy, OpenCL initialization, the disposal of boundary conditions and MPI communication on CPU. In our six-component mesh grid system, the parallel implementation in the whole computational domain of the spherical shell from the Sun to Earth is realized by domain decomposition of six-component decomposition of the spherical surface and PARAMESH application program interface (API). Correspondingly, we will have a similar grid structure of six-component overlapping patches in logically Cartesian space. In the present work, we employ 24 processes on CPUs to deal with the whole computational domain.

Within the same component or in the intra-component, we use PARAMESH guardcell API to manage guardcells that are responsible for the neighboring blocks. While, the boundary or internal border value in the overlapping area of inter-components need to be determined by an interpolation from its neighbor component grids according to the related geometrical positions of component grids (Feng et al. 2010). To improve speed-up ratio, we take advantage of non-blocking communication to overlap the time of computation. That is, during the data transferring of overlapping areas among CPU

processes, the computation on the inner portion of the component can start, and operates in parallel with the MPI non-blocking communication.

The computational domain is first covered by root blocks with the lowest spatial resolution. Initially, 56 blocks are assigned to each process that corresponds to one GPU, and therefore a single GPU kernel can be applied to all the 56 blocks. Since the amount of computation workload of each block is the same, there will be no synchronization overhead when multiple blocks are evolved in parallel by GPU. As time advances, cell refines and coarsens to different levels on different components and thus number of blocks will change correspondingly. When mapping all blocks to GPUs as shown in Figure 1b, the mapped workspace on 24 GPUs must guarantee the coverage of the whole computational nodes and each process must have the capability of copying with additionally new-born blocks after refinement within its allowed maximum number of initial 56 blocks. In practice, in GPU workspace, we can set the size of a work group as 64×8 work items for one block with $8 \times 8 \times 8$ cells. That is, each work item in its work group computes one cell of a block. The most heavily tasked process holds about 2500 blocks. In this case, 2500 blocks have to be divided into seven sequential calculations on GPU since our GPU card is out of memory after the block size exceeds 400. Here, we can set the number of work group as 360 in order to include the maximum number of all the permitted cells in the same step even at the finest refinement levels (at most 5 levels in practice).

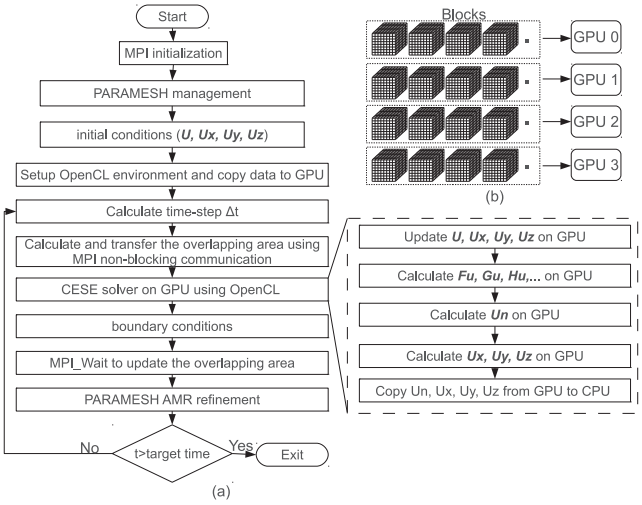

Figure 1. (a) The flowchart of GPU implementation of SIP-CESE MHD Model. (b) Mapping blocks to GPUs.

The test cases are carried out on a 12-node GPU/CPU cluster with two quad-core Intel Xeon E5620 processors operating as 2.4 GHz. Each node consists of two GPU cards, and the cluster, in total, has 10 Tesla C1060 cards and 14 Tesla C2050 cards, which are connected by Gigabit Ethernet. The CPU/GPU cluster is equipped with OpenCL 1.1 supported by CUDA 4.0, and MPI Compiler supplied with Open MPI 1.3 (`http://www.open-mpi.org/`), which is compiled by GCC 4.1 (GNU Compiler Collection, `http://gcc.gnu.org/`) with the optimization option "-O3". Table 1 shows preliminary performance comparison between the CPU and GPU of major calculations

Block Size	Operations (include copying data between GPU and CPU)	CPU (Intel® Xeon® CPU E5620@2.40GHz)	GPU (C2050)	Ratio
192	Flux ($\mathbf{U}_t$, F_u,...)	0.56	0.075	7.42
	$\mathbf{U}_n$	1.29	0.17	7.68
	$\mathbf{U}_x$, $\mathbf{U}_y$, $\mathbf{U}_n$	0.68	0.078	8.66
	all	2.53	0.33	7.76
384	Flux ($\mathbf{U}_t$, F_u,...)	1.12	0.14	8.10
	$\mathbf{U}_n$	2.38	0.31	7.58
	$\mathbf{U}_x$, $\mathbf{U}_y$, $\mathbf{U}_n$	1.45	0.16	8.92
	all	4.95	0.61	8.11

Table 1. Preliminary performance comparison (s/step) between the CPU and GPU of CESE solver calculations in double precision with different block sizes

within CESE solver in double precision with different block sizes. From this table we can see that the speedup ratio increases with the block size. Performance comparison of one time-step between the CPU and GPU in double precision in global computation domain shows that using 24 MPI processes without GPU takes us 16 minutes, while 24 MPI processes with GPU 3.5 minutes, which gives us about 5x speedup. In our test, the present OpenCL code using 24 MPI processes on CPU/GPU cluster finishes with wall time 30 hrs to reach a steady-state at the physical time 180 hrs for one CR. However, on the same platform with the use of 24 MPI processes for the original Fortran code it takes 156 hrs wallclock time.

4. Numerical Results

In this section, we concisely present the model's results for the steady solar corona and interplanetary space in CR 2060 and compare them with the observations from multiple spacecraft to validate the performance of the SIP-CESE MHD model with AMR of hexahedral meshes developed formerly. In fact, the solar wind background in the inner heliosphere during CR 2060 has been investigated by some researchers (Riley et al. 2012; Pahud et al. 2012; Feng et al. 2012a; Yang et al. 2012), and here we select it as our studied interval in order to better assess the model.

In Figure 2, we present the synoptic maps of white-light polarized brightness (pB) at the east (left) and west (right) limbs observed by the Large Angle Spectrometric Coronagraph (LASCO) C2 on board SOHO and the synoptic maps of the proton number density and the radial velocity on the surface of 2.5 radii. The bright areas in pB images often indicate that there are high-density structures near the sky plane along the line of sight through these points. This figure shows that both the MHD and PFSS models give almost the same magnetic neutral line (MNL) characterized by two peaks roughly at $\phi = 150°$ and $\phi = 310°$ and a trough at $\phi = 240°$, which indicates there exists the presence of the tilting and warping of the MNL and also the curving of the streamer belt during this period. The peak of the MNL is located just next to the southern coronal hole and the trough next to the isolated equatorial coronal hole. Figure 2 also shows that the MNL is surrounded by regions of high density and low speed plasma

flow, which is consistent with the distribution of the pB enhanced regions observed by LASCO/C2 and the simulation conducted by Feng et al. (2012a). It should be noted that the high-density structures are also present in the regions far away from the MNL shown in the upper panels of Figure 2 and are associated to uniploar streamers (Riley and Luhmann 2012).

In order to consider the solar wind structure in interplanetary space and the in-situ measurements, we present the model solutions at the surfaces of 20 Rs and 215 Rs in Figure 3a and compare the modeled temporal profiles of the solar wind parameter with the in-situ measurements in Figure 3b. Figure 3a shows that the first peak of HCS roughly centered around at $\phi = 150^\circ$ at 2.5 Rs shifts to $\phi = 140^\circ$ at 20 Rs and $\phi = 80^\circ$ at 215 Rs due to the solar rotation, while the trough at $\phi = 240^\circ$ shifts to $\phi = 230^\circ$ at 20 Rs and $\phi = 165^\circ$ at 215 Rs. What's more, the highest mass-flux regions are coincident with the locations of HCS and the low-speed solar wind ($v_r < 550$ km/s) covers about 50°-65° in latitude, unlike the low-speed solar wind only spanned 40° in the previous solar minimum. These are consistent with the studies on the unusual solar minimum caused by the weaker polar photospheric magnetic field (de Toma 2011; Yang et al. 2011; Feng et al. 2012a). In addition, the upper right panel in Figure 3a shows that the finer AMR grids resolve the interfaces between high-speed and low-speed streams very well. Seen from Figure 3b, the model matches the stream structure reasonably well. The model solution shows that the initial long-duration of slow wind is followed by a strong and steep high-speed stream on Day 12, August 26, 2008, which is one day earlier than the observation. The second modeled high-speed stream rises on almost the same date as observed, but declines so slowly that the solution misses the low-speed stream between the second and third high-speed streams. Both the modeled and the measured polarities of the radial magnetic field during this interval switch from inward to outward around August 30, 2008 and are in reasonable agreement except on the first 4 days and the last two days. The smaller-scale reversals missed by the model's results probably represent waves and/or turbulence, which are not included in the MHD model. Like many other numerical MHD models, the radial field strength obtained from the model solution here is only one-third to one-half of the observation, which cannot be improved only by means of high-resolution grids as other researchers noted Riley et al. (2012).

5. Conclusions

The performance of the modified 3D MHD simulation code using CPU/GPU cluster in double precision was evaluated. By exploiting the feature of non-blocking communication for parallel execution between CPU and GPU, the data transferring of the overlapping areas among different components can be covered with the execution of the GPU solvers, and yields considerable performance enhancement. Although we used a simple parallelization approach, the computational speed of the modified code was improved significantly under the condition of the same processes. The performance of the present OpenCL code using CPU/GPU cluster in double precision was evaluated to give a speedup of 5x (at least) without any optimization. A direct improvement at hand is to future consider the following aspects: 1) load balancing and memory management in MPI will be left for next consideration and 2) spatial resolution needs to be further refined to identify the HCS.

The validated simulations for the steady corona and interplanetary solar wind reveals that the SIP-CESE AMR MHD model run on CPU/GPU clusters in double pre-

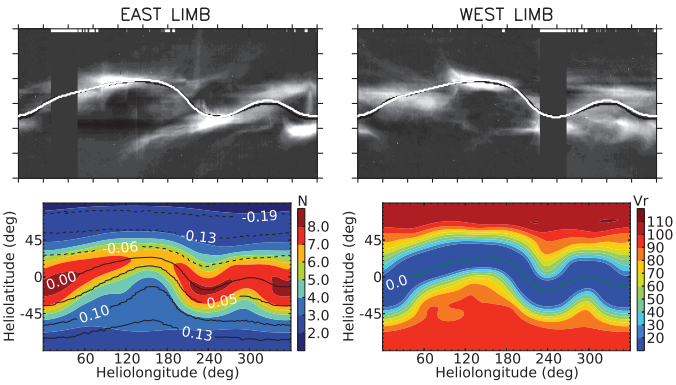

Figure 2. Synoptic maps at 2.5 R_S for CR 2060. The first row are the white-light polarized brightness at the east (*left*) and west (*right*) limbs from SOHO/LASCO-C2. In the first row, the black lines denote the magnetic neutral lines from the MHD model and the white lines from the PFSS model. In the second row, the left panel overlays the isolines of the radial magnetic field [Gauss] on the contour map of the simulated number density N [10^5 cm^{-3}] and the right one is the pseudo-color image of the radial speed v_r [km s^{-1}]. In the left panel, the dashed lines stand for inward radial magnetic field and the solid ones outward radial magnetic field. The dashed lines in the right panel denote the magnetic neutral lines.

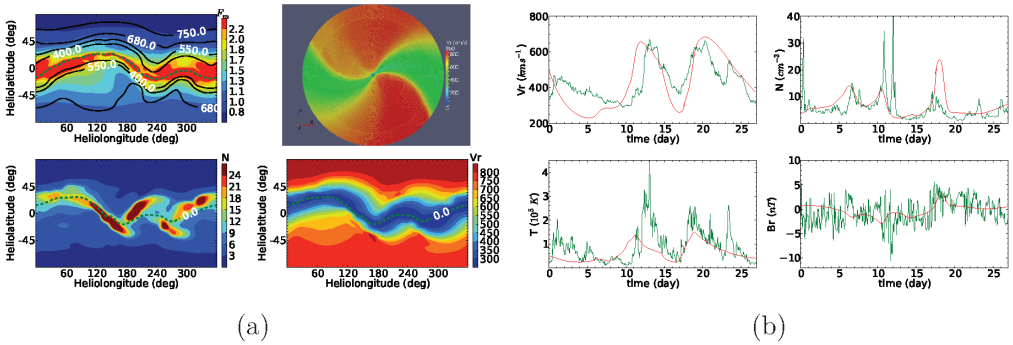

Figure 3. (a) The simulated steady solution in interplanetary space from the MHD model for CR 2060. The top row displays the isolines of the radial speed [v_r: km s^{-1}] superimposed on the synoptic pseudo-color image of the mass flux density [F_m: 10^8 km s^{-1} cm$^{-3}R_S^2$] at 20 R_S (*left*), and the pseudo-color image of the radial velocity in the solar equatorial plane (*right*). The bottom row shows the synoptic contours of the simulated proton number density [N: cm^{-3}] (*left*) and v_r [km s^{-1}] (*right*) at 215 R_S. Here, the dashed lines denote the magnetic neutral lines. (b) The calculated MHD steady state at 1 AU for CR 2060 with comparisons of the one-hour averaged OMNI data near 1 AU for radial solar-wind speed [v_r], number density [N], temperature [T], and radial magnetic field [B_r]. Here, the green lines denote the observations and the red lines represent the numerical results.

cision can achieve the basically consistent temporal profiles of solar wind parameter at 1 AU with the in-situ measurements. These features are essentially determined by the photospheric magnetic fields, especially their dipole and quadrupole components (Sanderson et al. 2003) and the polar open fields (Wang et al. 2009), and they are also in agreement with the previous studies (Riley and Luhmann 2012; Yang et al. 2012; Feng et al. 2012a). On the other hand, we should also note that there exist some differences between the numerical results and observations, such as the magnitude of radial magnetic fields and arrival times of the high-speed stream at the Earth, which conventionally appear in CPU MHD codes. For the purpose of improving the numerical results to better capture the structures of the heliosphere during specific time periods, methods suggested by Feng et al. (2012b) can be followed.

Acknowledgments. The work is jointly supported by the National Basic Research Program (973 program) under grant 2012CB825601, the Knowledge Innovation Program of the Chinese Academy of Sciences (KZZD-EW-01-4), the National Natural Science Foundation of China (41031066, 41231068, 41274192, 41074121, and 41074122), and the Specialized Research Fund for State Key Laboratories.

References

Behannon, K. W., Neubauer, F. M., and Barnstorf, H., 1981 J. Geophys. Res., 86, 3273.
Belleman, R. G., Bédorf, J., and Portegies Zwart, S., 2008 New Astronomy, 13, 103.
Che, S., Boyer, M., Meng, J. Y., Tarjan, D., Sheaffer, J. W., and Skadron, K., 2008 J. Parallel Distrib. Comput., 68, 1370.
DeLeon, R., and Senocak, I., 2012 AIAA 2012-0722.
de Toma, G., 2011 Solar Phys., 274, 195.
Feng, X. S., Yang, L. P., Xiang C. Q., Wu S. T., Zhou Y. F., and Zhong D. K., 2010 ApJ, 723, 30.
Feng, X. S., Zhang, S. H., Xiang, C. Q., Yang, L. P., Jiang, C. W., and Wu, S. T. , 2011 ApJ, 734, 50.
Feng, X. S., Yang, L. P., Xiang, C. Q., Jiang, C. W., Ma, X. P., Wu, S. T., Zhong, D. K., and Zhou, Y. F., 2012a Solar Phys., 279, 207.
Feng, X. S., Jiang, C. W., Xiang, C. Q., Zhao, X. P., and Wu, S. T., 2012b ApJ, 758, 62.
Gaburov, E., Harfst, S., and Portegies Zwart, S., 2009 New Astronomy, 14, 630.
Lin, L., Ng, C. S., and Bhattacharjee, A., 2012 Numerical Modeling of Space Plasma Slows (ASTRONUM 2011), 459, 222.
Owens, J. D., Luebke, D., Govindaraju, N., Harris, M., Krüger, J., Lefohn, A. E., and Purcell, T. J., 2007 Computer Graphics Forum, 26, 80.
Pahud, D. M., Merkin, V. G., Arge, C. N., Hughes, W. J., and McGregor, S. M., 2012 Journal of Atmospheric and Solar-Terrestrial Physics, 83, 32.
Riley, P., Linker, J. A., Lionello, R., and Mikic, Z., 2012 Journal of Atmospheric and Solar-Terrestrial Physics, 83, 1.
Riley, P., and Luhmann, J. G., 2012 Solar Phys., 277, 355.
Sanderson, T. R., Appourchaux, T., Hoeksema, J. T., and Harvey, K. L., 2003 J. Geophys. Res., 108(A1), 1035.
Schenk, O., Christen, M., and Burkhart, H., 2008 J. Parallel Distrib. Comput., 68, 1360.
Schive, H. Y., Tsai, Y. C., and Chiueh, T., 2010 ApJS, 186, 457.
Wang, P., Abel, T., and Kaehler, R., 2010 New Astronomy, 15, 581.
Wang, Y.-M., Robbrecht, E., and Sheeley, N. R., 2009 ApJ, 707, 1372.
Wong, H. C., Wong, U. H., Feng, X. S., and Tang, Z. S., 2011 Comput. Phys. Commun., 182, 2132.
Yang, L. P., Feng, X. S., Xiang, C. Q., Zhang, S. H., and Wu, S. T., 2011 Solar Phys., 217, 91.

Yang, L. P., Feng, X. S., Xiang, C. Q., Liu, Y., Zhao, X. P., and Wu, S. T., 2012 J. Geophys. Res., 117(A16), 8110.