SCIENCE CHINA
Earth Sciences

Progress of Projects Supported by NSFC
• RESEARCH PAPER •

November 2013 Vol.56 No.11: 1864–1880
doi: 10.1007/s11430-013-4661-y

GPU-accelerated computing of three-dimensional solar wind background

FENG XueShang*, ZHONG DingKun, XIANG ChangQing & ZHANG Yao

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

Received March 27, 2013; accepted May 22, 2013; published online July 3, 2013

High-performance computational models are required to make the real-time or faster than real-time numerical prediction of adverse space weather events and their influence on the geospace environment. 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 are obtained by the established GPU model with the magnetic field synoptic data as input. Meanwhile, the time-dependent solar surface boundary conditions derived from the method of characteristics and the mass flux limit are incorporated to couple the observation and the three-dimensional (3D) MHD model. The simulated evolution of the global structures for two Carrington rotations 2058 and 2062 is compared with solar observations and solar wind measurements from spacecraft near the Earth. The MHD model is also validated by comparison with the standard potential field source surface (PFSS) model. Comparisons show that the MHD results are in good overall agreement with coronal and interplanetary structures, including the size and distribution of coronal holes, the position and shape of the streamer belts, and the transition of the solar wind speeds and magnetic field polarities.

space weather modeling, SIP-CESE MHD model, GPU computing

Citation: Feng X S, Zhong D K, Xiang C Q, et al. GPU-accelerated computing of three-dimensional solar wind background. Science China: Earth Sciences, 2013, 56: 1864–1880, doi: 10.1007/s11430-013-4661-y

Space weather, with scientific research and applications as its two focal points, 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 (Bothmer et al., 2007; Baker, 2002). 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.

Global computational models based on first principles represent a very important component of efforts to understand the intricate processes coupling the Sun to the geospace environment. The hope for such models is that they will eventually fill the gaps left by measurements, extending the spatially and temporarily limited observational database into a self-consistent global understanding of our space environment. Presently, and in the foreseeable future, magnetohydrodynamic (MHD) models are the only models that can span the enormous distances present in the corona-interplanetary space, although even generalized MHD equa-

*Corresponding author (email: fengx@spaceweather.ac.cn)

tions are only a relatively low-order approximation to more complete physics by providing only a simplified description of natural phenomena in space plasmas. Promisingly, scientists have started to use a three-dimensional MHD-based solar-magnetosphere-ionosphere coupling simulation for operative space weather forecasting purposes (Dryer, 2007; Feng et al., 2010, 2011a, b, 2012a; Hu et al., 2007; Janhunen et al., 1996; Jin et al., 2012; Lugaz et al., 2011; Lyon et al., 2004; Lyon, 2000; Ogino, 1986, 2002; Powell et al., 1999; Raeder et al., 2001; Riley et al., 2011, 2012a; Tanaka, 1994, 1995; Tóth et al., 2012; van der Holst et al., 2010; Watermann et al., 2009). On one hand, especially for such applications, the robustness of the code is very important since the program is expected at least to converge for any solar wind input, if not necessarily to produce a correct forecast in every situation. On the other hand, forecasting in advance of the arrival and impact effect of severe space weather at the Earth is another crucial factor, since large solar eruptive events, such as coronal mass ejections (CMEs), arrive at the Earth within one to three days. CMEs can inject large quantities of mass and magnetic flux into the heliosphere and can drive interplanetary shocks, which are a key source of solar energetic particles and are known to be the major contributor to severe space weather at the Earth. Studies over the past decade using the data sets from SOHO, TRACE, Wind, ACE, STEREO, and SDO spacecraft, along with ground-based instruments, and CME modeling (Feng et al., 2011a; Kleimann, 2012) have enriched our knowledge of the origins and development of CMEs at the Sun, and their evolution in interplanetary space and their contribution to space weather at the Earth (Webb et al., 2012). For numerical space weather forecasts, the three most crucial quantities to be addressed by a model are the CME's trajectory, travel time, and geoeffectiveness.

In quantitative studies of the solar wind structures, 3D global MHD models have been developed (e.g., Feng et al., 2010, 2007; Lugaz et al., 2011; Tanaka, 1995; Tóth et al., 2012; Lionello et al., 2009; Mikić et al., 1999; Nakamizo et al., 2009; Cohen et al., 2008; Riley et al., 2006; Roussev et al., 2003; Taktakishvili et al., 2011; Tóth et al., 2005; Usmanov et al., 2006). As pointed out by space weather scientists (e.g., Aschwanden et al., 2008; Dryer, 2007; Feng et al., 2011a; Watermann et al., 2009), high performance computational models require further improvements in order to make the real-time or faster than real-time numerical prediction of adverse space weather events and their influence on the geospace environment. Computationally speaking, such speedup model improvement involves two issues. On one hand, numerical solutions to the governing magnetohydrodynamic (MHD) equations currently used for the numerical space weather modeling from the Sun to Earth or beyond are typically feasible only on massively parallel computers for the sake of computational resources. On the other hand, one numerical challenge is due to the presence of different temporal and spatial scales on which solar wind plasma occurs throughout the vast solar-interplanetary space of these problems.

Corona-interplanetary space involves a large extent, and contains many critical features, such as discontinuities and heliospheric current sheet, which 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 Courant-Friedrichs-Lewy (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 (~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 × 215 × 215 R_S^3 (R_S = 6.955 × 10^5 km, the radius of the Sun) at a resolution of ~0.057 R_S (comparable to the thickness of heliospheric current sheets since the existence of a wide range in HCS width at 1 astronomical unit (AU) (1 AU is equal to 215 R_S), is typically agreed to be from 40000 to 100000 km (Blanco et al., 2006; Behannon et al., 1981) or even smaller (Podgorny et al., 2005), one would need times of 1010 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 h 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, 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 (i.e., 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) is 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 (http://developer.download.nvidia.com/compute/DevZone/docs/html/C/doc/CUDA_C_Programming_Guide.pdf).

Due to the great computational power of the GPU, the GPU computing method has proven valuable in various areas of science and technology.

1 GPU computing

Modern many-core processor or multi-core processor, e.g., current generation NVIDIA and AMD GPU, is a highly parallel, highly multithreaded multiprocessor and has a unified graphics and computing architecture that serves as both a programmable graphics processor and a scalable parallel computing platform.

The advent of the Compute Unified Device Architecture (CUDA) (NVIDIA Corporation, NVIDIA CUDA Compute Unified Device Architecture Programming Guide 2.2.1, May 2009, http://www.nvidia.com/object/cuda home new.html) hardware and software releases computational scientists from working with graphics-specific shading languages. The CUDA programming model has an SPMD (single-program multiple data) software style, in which a programmer writes a program for one thread that is instanced and executed by many threads in parallel on the multiple processors of the GPU. In fact, CUDA is an environment for writing parallel programs for the entire heterogeneous computer system.

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. OpenCL supports a wide range of applications, ranging from embedded and consumer software to high performance computing solutions, through a low-level, high-performance, portable abstraction. By creating an efficient, close-to-the-metal programming interface, OpenCL will form the foundation layer of a parallel computing ecosystem of platform-independent tools, middle ware and applications. Meanwhile, OpenCL consists of an API (Application Program Interface) for coordinating parallel computation across heterogeneous processors and a cross-platform programming language with a well-specified computation environment (http://www.khronos.org/registry/cl/sdk/1.1/docs/man/xhtml/).

As a result, GPUs have become an alternative parallel computing device for high-performance scientific computing (Che et al., 2008; Owens et al., 2007, 2008; Schenk et al., 2008). There are an increasing number of researches on using GPUs for scientific applications to both fluid and astrophysical simulations (Gaburov et al., 2009; Belleman et al., 2008; Schive et al., 2010; Wang et al., 2010).

Michalakes et al. (2008) explored the performance of GPU by selecting a computationally intensive module from the Weather Research and Forecast (WRF) model and demon-started a 20× speedup for a computationally intensive portion of the WRF model on an NVIDIA GTX8800 GPU and an overall 1.3× speedup.

Brandvik et al. (2008) solved 2D and 3D Euler equations with uniform grids by a speedup factor of 29 (2D) and 16 (3D). Elsen et al. (2008) solved 3D Euler equations on multi-block meshes, where speed-ups of over 40× were demonstrated for simple test geometries and 20× for complex geometries. Using a quad-GPU platform, Thibault et al. (2009) observed two orders of magnitude speedup relative to a serial CPU implementation of the Navier-Stokes (NS) equations. Jacobsen et al. (2010) presented a dual-level parallel implementation of the Navier-Stokes (NS) equations to simulate buoyancy-driven incompressible fluid flows on multi-GPU clusters with heterogeneous architectures, and found that performance on a fixed problem size using 128 GPUs on 64 compute-nodes resulted in a speedup of 130× over the CPU solution using Pthreads on two quad-core 2.33 GHz Intel Xeon processors. Then, DeLeon et al. (2012) extended earlier work (Elsen et al., 2008; Thibault et al., 2009) on multi-GPU acceleration of an incompressible NS solver to include a large-eddy simulation (LES) capability. Kestener et al. (2010) solved the 2D compressible Euler equations on GPU with two different second order numerical schemes of Godunov and Kurganov-Tadmor types. Their tests showed that these two numerical schemes can achieve 30× to 70× speed-up compared to a mono-thread CPU reference implementation.

Belleman et al. (2008) re-implemented the direct gravitational N-body simulations on GPUs using CUDA, and for $N \geqslant 10^5$, they reported a speedup of about 100 compared to the host CPU and about the same speed as the GRAPE-6Af. Gaburov et al. (2009) presented Sapporo, a library for performing high-precision gravitational N-body simulations on an NVIDIA GT200 GPU to get a noticeable speedup 2× in peak performance. Schive et al. (2010) implemented the GPU adaptive mesh refinement for astrophysics problems and obtained speedup factors of 12.19 and 10.47 on 1 GPU with 4096^3 resolution and 16 GPUs with 8192^3 resolution, respectively. Wang et al. (2010) described an implementation of compressible inviscid fluid Harten-Lax-van Leer solvers for pure hydrodynamic and magnetohydrodynamic cases with block-structured adaptive mesh refinement on GPUs using NVIDIA's CUDA, and achieved an overall speedup of approximately 10 times faster execution on one graphics card as compared to a single core on the host computer, and this speedup in uniform grid runs as well as in problems with deep AMR hierarchies. Wong et al. (2011)

have implemented a TVD (total variation diminishing) MHD code on GPUs and achieved speedup factors of 106 with 1024^2 grids (2D problem) and 43 with 128^3 grids (3D problem), respectively.

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. Thus, it is undoubtful that the porting of three-dimensional numerical solvers from a conventional CPU implementation to the novel target platform of the Graphics Processing Unit (GPU) is promising in establishing speedup code. The objective of the present paper is to transfer our SIP-AMR-CESE MHD model (Feng et al., 2012a) to GPU platform.

2 Description of AMR SIP-CESE MHD model

This section is devoted to the introduction of governing equations, grid partition in the computational domain of the Sun-to-Earth spherical shell and the AMR Implementation of SIP-CESE MHD Model.

2.1 Governing equations

The three-dimensional equations governing solar-wind plasma are the set of the MHD equations in the conservative form in the frame corotating with the Sun, which reads as follows:

$$\frac{\partial \mathbf{U}}{\partial t}+\frac{\partial \mathbf{F}}{\partial x}+\frac{\partial \mathbf{G}}{\partial y}+\frac{\partial \mathbf{H}}{\partial z}-\frac{\partial \mathbf{F}_\upsilon}{\partial x}-\frac{\partial \mathbf{G}_\upsilon}{\partial y}-\frac{\partial \mathbf{H}_\upsilon}{\partial z}=\mathbf{S}, \qquad (1)$$

where

$$\mathbf{U}=(\rho,\rho v_x,\rho v_y,\rho v_z,e,B_x,B_y,B_z)^{\mathrm{T}},$$

$$\mathbf{F}=\begin{pmatrix}\rho v_x\\ \rho v_x^2+p_0-B_x^2\\ \rho v_x v_y-B_xB_y\\ \rho v_x v_z-B_xB_z\\ (e+p_0)v_x-(v\cdot B)B_x\\ 0\\ v_xB_y-B_xv_y\\ v_xB_z-B_xv_z\end{pmatrix},\quad \mathbf{G}=\begin{pmatrix}\rho v_y\\ \rho v_y v_x-B_yB_x\\ \rho v_y^2+p_0-B_y^2\\ \rho v_y v_z-B_yB_z\\ (e+p_0)v_y-(v\cdot B)B_y\\ v_yB_x-B_yv_x\\ 0\\ v_yB_z-B_yv_z\end{pmatrix},$$

$$\mathbf{H}=\begin{pmatrix}\rho v_z\\ \rho v_z v_x-B_zB_x\\ \rho v_z v_y-B_zB_y\\ \rho v_z^2+p_0-B_z^2\\ (e+p_0)v_z-(v\cdot B)B_z\\ v_zB_x-B_zv_x\\ v_zB_y-B_zv_y\\ 0\end{pmatrix},\quad \mathbf{F}_\upsilon=\begin{pmatrix}0\\0\\0\\0\\0\\ \upsilon\nabla\cdot\mathbf{B}\\0\\0\end{pmatrix},$$

$$\mathbf{G}_\upsilon=\begin{pmatrix}0\\0\\0\\0\\0\\0\\ \upsilon\nabla\cdot\mathbf{B}\\0\end{pmatrix},\quad \mathbf{H}_\upsilon=\begin{pmatrix}0\\0\\0\\0\\0\\0\\0\\ \upsilon\nabla\cdot\mathbf{B}\end{pmatrix},$$

$$\mathbf{S}=(0,\rho\mathbf{F}_0,\rho\mathbf{v}\cdot\mathbf{F}_0,0)^{\mathrm{T}}-\nabla\cdot\mathbf{B}(0,\mathbf{B},\mathbf{v}\cdot\mathbf{B},\mathbf{v})^{\mathrm{T}}+(0,\mathbf{0},Q_e,\mathbf{0})^{\mathrm{T}}$$

with the total pressure $p_0=p+\frac{1}{2}\mathbf{B}^2$. As usual, $e=\frac{p}{\gamma-1}+\frac{1}{2}\rho\mathbf{v}\cdot\mathbf{v}+\frac{1}{2}\mathbf{B}^2$, ρ, $\mathbf{v}$, p, and $\mathbf{B}$ are the total energy density, the mass density, plasma velocity, gas pressure, and magnetic field. In eq. (1), the external force $\mathbf{F}_0=-\frac{GM_s}{r^3}\mathbf{r}-\boldsymbol{\Omega}\times(\boldsymbol{\Omega}\times\mathbf{r})-2\boldsymbol{\Omega}\times\mathbf{v}$ is the sum of solar-gravity force and inertial force due to the corotating frame with the Sun. G, M_s, and $\boldsymbol{\Omega}$ are the gravity constant, solar mass, and solar angular speed. γ is the specific-heat ratio and is taken to be 1.5 here. Solar rotation is considered in the present study with angular velocity $|\boldsymbol{\Omega}|$ = 2π/27.2753 radian day^{-1}. In eq. (1), the symmetrizable source term $-\nabla\cdot\mathbf{B}(0,\mathbf{B},\mathbf{v}\cdot\mathbf{B},\mathbf{v})^{\mathrm{T}}$ (Powell et al., 1999) and the diffusive control term $\nabla(\upsilon\nabla\cdot\mathbf{B})$ (Feng et al., 2011b, 2012a; Marder, 1987; van der Holst, 2007) have been added in the MHD equations to deal with the divergence of the magnetic field. Here, following Feng et al. (2010, 2011b), $\upsilon=1.3\left(\frac{1}{\Delta x^2}+\frac{1}{\Delta y^2}+\frac{1}{\Delta z^2}\right)^{-1}$, where Δx, Δy, and Δz are grid spacings in Cartesian coordinates.

In order to achieve the observed pattern of fast and slow solar winds through MHD simulations, the energy-source term Q_e in the energy equation adopts the volumetric heating method as done by Feng et al. (2012a) with the help of the Wang-Sheeley-Arge (WSA) model (Wang et al., 1990; Arge et al., 2000, 2003). This energy-source term takes into account the flux-tube expansion factor f_S and the minimum angular separation (at the photosphere) between an open-field foot point and its nearest coronal hole boundary (θ_b, measured in degrees), which have been used for the solar-wind study (Nakamizo et al., 2009; Taktakishvili et al., 2011).

Other details such as the normalization and the splitting of the full magnetic-field vector $\mathbf{B}$ into the sum of a time-independent potential magnetic field [$\mathbf{B}_0$] and a time-dependent deviation $\mathbf{B}_1$, i.e., $\mathbf{B}=\mathbf{B}_0+\mathbf{B}_1$, in the governing equations, have been given in Feng et al. (2012a).

The computational domain is set to be the domain [(r, θ, ϕ): 1 $R_S \leqslant r \leqslant 240$ R_S; $0\leqslant\theta\leqslant\pi$; $0\leqslant\phi\leqslant 2\pi$]. The initial

solar-surface temperature and number density are 1.3×10^6 K and 2.0×10^8 cm^{-3}. Then, the code is initialized by using Parker solar-wind solution and the potential magnetic field based on the line-of-sight measurements of the photospheric magnetic field from Wilcox Solar Observatory (WSO) at Stanford University for the Carrington Rotation (CR) of interest. The same projected-normal characteristic boundary conditions and initial conditions as those given in Section 5 of Feng et al. (2010) are employed in the present article. Finally, our model is run in time-relaxation method until a quasi-steady state is achieved.

The SIP-CESE MHD model has been developed by us in a series of articles (Feng et al., 2007, 2010, 2012a). For the present article to be self-contained, in what follows we present the main points of six-component grid and curvilinear coordinate transformation from physical to reference coordinates from the context of solar-wind modeling.

2.2 Six-component grid and curvilinear coordinate transformation

In the six-component grid introduced by Feng et al. (2010) for the solar-wind study, the spherical-shell computational domain is decomposed into six identical components with partial overlapping regions (Figure 1), and each component is identically defined by a low-latitude spherical domain

$$\left(\frac{\pi}{4}-\delta\leqslant\theta\leqslant\frac{3\pi}{4}+\delta\right)\cap\left(\frac{3\pi}{4}-\delta\leqslant\varphi\leqslant\frac{5\pi}{4}+\delta\right).$$

The parameter $\delta=\Delta\theta$ is determined by the grid spacing and layers of guard cells required for the minimum overlapping area of two grid sizes. In the six-component grid partitions, the same metric can be used and the component grids can be transformed into each other by coordinate transformation such that the basic equations, numerical grid distribution, and all numerical tasks are identical in each component. Hence we do not have to distinguish them and we only need to describe the grid partition and the associated coordinate transform. The grid points in both θ and ϕ directions are evenly spaced such that $\Delta\theta=\Delta\phi$. In the r direction, a new variable ξ is introduced as a reference coordinate, which is exponentially related with r by $r=e^{\xi}$. In this way, the grid spacing $[\Delta r]$ in the r direction is always around $r\Delta\theta$ and $r\sin\theta\Delta\phi$ by choosing $\Delta\xi=\ln(1+\Delta\theta)$, so that each grid cell is always approximately a cube. As pointed out by Feng et al. (2012a), the reference coordinates (ξ, η, ζ) used in our CESE solver refer to (ξ, θ, ϕ) here, and with this setting, the grid cell in the reference space (ξ, η, ζ) is a rectangular box. In principle, any parameter $a>1$ can be chosen instead of e and a representative case $a>1.481$ is just presented for the validation (Feng et al., 2012a).

Initially, the computational domain in every reference component is divided into $14\times4\times4$ blocks with each block consisting of $6\times6\times6$ cells. These correspond to $N_\theta=N_\phi=25$ and $\Delta\theta=\pi/48$ by defining gird points on each component in physical space as $\theta_j=\theta_{\min}+j\Delta\theta, j=0, 1,\cdots, N_\theta+1$, $\phi_k=\phi_{\min}+k\Delta\phi, k=0, 1,\cdots, N_\phi+1$ and $\Delta\theta=(\theta_{\max}-\theta_{\min})/(N_\theta-1)$, $\Delta\phi=(\phi_{\max}-\phi_{\min})/(N_\phi-1)$, where N_θ and N_ϕ are the mesh numbers of the latitude and longitude, respectively. $\theta_{\min}=\pi/4$, $\theta_{\max}=3\pi/4$, $\phi_{\min}=3\pi/4$, $\phi_{\max}=5\pi/4$. The innermost region is set on the solar surface at 1 R_S and the outermost region on the sphere at 240 R_S, such that each component of the physical grids is equivalent to $0\leqslant\xi\leqslant5.4807$, $\pi/4-\delta\leqslant\eta\leqslant3\pi/4+\delta$, and $3\pi/4-\delta\leqslant\zeta\leqslant5\pi/4+\delta$ in the computational domain or reference space, with $\Delta\xi=\ln(1+\Delta\theta)$, and $\Delta\eta=\Delta\zeta=\delta=\Delta\theta=\pi/48$. Corresponding to the computational nodes (N_r, N_θ, N_ϕ) in the physical space, assume (N_ξ, N_η, N_ζ) be the associated computational nodes in the ξ, η and ζ directions for a component computational domain in the reference coordinate space (ξ, η, ζ) or directly denote it by (N_x, N_y, N_z) for convenience.

With the transformation [**J**] between the coordinates of the reference space and the physical space for one component,

$$\begin{cases}x=e^{\xi}\sin\eta\cos\zeta\\ y=e^{\xi}\sin\eta\sin\zeta\\ z=e^{\xi}\cos\eta\end{cases}\qquad\begin{cases}\xi=\ln\sqrt{x^2+y^2+z^2}\\ \eta=\arccos(z/\sqrt{x^2+y^2+z^2})\\ \zeta=\arctan(y/x)\end{cases}\quad(2)$$

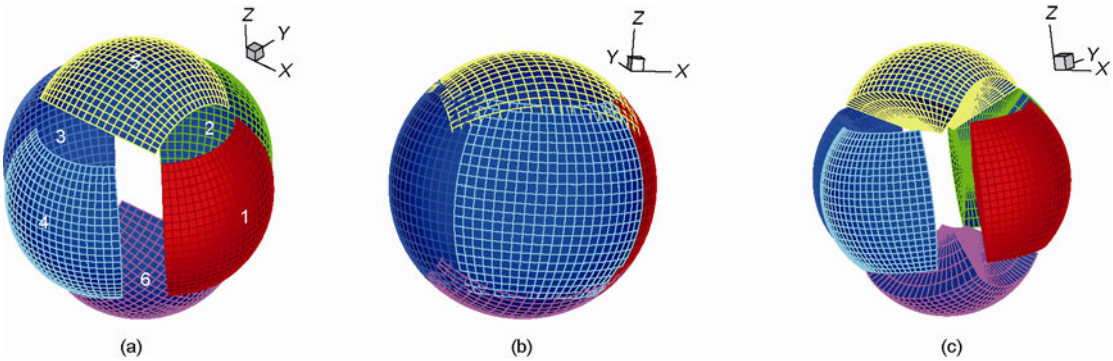

Figure 1 Six-component grid. (a) A spherical overset grid consisting of six identical components; (b) dividing a sphere into six identical components with partial overlap; (c) stacking the spherical meshes of each component up in the radial direction.

the 3D MHD equations of solar-wind plasma (1) in the reference coordinates (ξ, η, ζ) can read below:

$$\frac{\partial \hat{\mathbf{U}}}{\partial t}+\frac{\partial \hat{\mathbf{F}}}{\partial \xi}+\frac{\partial \hat{\mathbf{G}}}{\partial \eta}+\frac{\partial \hat{\mathbf{H}}}{\partial \zeta}-\frac{\partial \hat{\mathbf{F}}_{\upsilon}}{\partial \xi}-\frac{\partial \hat{\mathbf{G}}_{\upsilon}}{\partial \eta}-\frac{\partial \hat{\mathbf{H}}_{\upsilon}}{\partial \zeta}=\hat{\mathbf{S}}, \qquad (3)$$

where the corresponding transformation of coordinates between any two components or transformation of solution variables between $(\hat{\mathbf{U}},\hat{\mathbf{F}},\hat{\mathbf{G}},\hat{\mathbf{H}},\hat{\mathbf{F}}_{\upsilon},\hat{\mathbf{G}}_{\upsilon},\hat{\mathbf{H}}_{\upsilon},\hat{\mathbf{S}})$ in the reference space and $(\mathbf{U},\mathbf{F},\mathbf{G},\mathbf{H},\mathbf{F}_{\upsilon},\mathbf{G}_{\upsilon},\mathbf{H}_{\upsilon},\mathbf{S})$ in the physical space on an arbitrary component have been provided by Feng et al. (2010) and Jiang et al. (2010).

Note that eqs. (3) and (1) share the same form, and thus the CESE solver for eq. (1) in the physical space can be transplanted directly to the CESE solver for eq. (3) in the reference coordinate space (ξ, η, ζ) with rectangular box cells in a logically Cartesian space.

With these in mind, we can easily take turns between the physical-solution variables and the reference-solution variables. That is, if the initial inputs of the physical solution variables are given, we first use **F**, **G**, **H**, **S** to calculate the fluxes $\hat{\mathbf{F}},\hat{\mathbf{G}},\hat{\mathbf{H}},\hat{\mathbf{S}}$ and apply the CESE solver to eq. (3) to obtain the reference-solution variables $\hat{\mathbf{U}}$ and their first-order derivatives $(\hat{\mathbf{U}}_{\xi},\hat{\mathbf{U}}_{\eta},\hat{\mathbf{U}}_{\zeta})$ at the new time step. Finally, we can recover the physical solution variables **U**. Of course, we can reverse the above process from the reference-solution variables to the physical-solution variables when needed.

2.3 AMR implementation of SIP-CESE MHD model

Feng et al. (2012a) presented the details of the AMR implementation of the SIP-CESE MHD model on the six-component grid system (Figure 1) of the spherical-shell domain in solar-terrestrial space. To accomplish this, all the ranks are classified into six groups, each of which corresponds to one component grid system and deals with the same grid, basic equations, and numerical task. For a spherical surface grid (θ, ϕ) with piling up the radial direction, a simple radial decomposition of the spherical shell computational domain can be managed to yield very good load balancing since the processor workload and the communication load are very similar for each processor. However, this is not necessarily true for other grids, in particular AMR in Cartesian coordinates. With the help of the PARAMESH package at http://sourceforge.net/projects/paramesh/, 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 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 nvf = 4 × 8 = 32.

Initially, the computational domain in every reference component is divided into 224 blocks with each block consisting of 8 × 8 × 8 cells with one layer of guard cells included. Totally we have 224 blocks for each component. That is, 6 × 4 × 4 × 14 blocks for the whole computational domain. The SIP-AMR-CESE MHD model carries out the AMR implementation with the help of PARAMESH according to the refinement strategy of the curl of the magnetic field (e.g., Feng et al. (2012a) and references therein).

In solar-wind modeling, the topology of helioshperic current sheet is an important structure and thus only the curl of the magnetic field is used as our refinement strategy to capture the current sheet. The standard deviation about zero $\left[\tau_e=\sqrt{\frac{\sum_{i=1}^{N}\chi_{ei}^2}{N}}\right]$ is computed for $\chi_e=V^{0.5}\frac{|\nabla\times B|}{|B|+\varepsilon\sqrt{p}}$ and $\varepsilon=10^{-10}$ to set the thresholds of refining and coarsening grid (De Zeeuw, 1993), with V being the spatial cell volume. When the maximum of this criterion in a block is greater than $\kappa\tau_e$, this block is flagged to be refined, while if the maximum in a block is less than $\varsigma\tau_e$, this block is flagged to be coarsened. κ and ς are selected according to different physical time (t_H in units of hours) intervals during the code's running. That is, when t_H < 20, κ = 3 and ς = 0.1; when $20\leqslant t_H<50$, κ = 6 and ς = 0.2; when $50\leqslant t_H<100$, κ = 10 and ς = 0.3; when $100\leqslant t_H<160$, κ = 15 and ς = 0.4; when $t_H\geqslant 160$, κ = 6 and ς = 0.1. With these settings, three levels of grid refinement are used to obtain a grid cell size of 0.012 R_S on the solar surface. The grid throughout the simulation is refined to obtain a grid cell size of about 0.16 R_S with three levels of grid refinement near the current sheet within 20 R_S and it is about 0.55 R_S near 1 AU with five levels of grid refinement. The maximum grid cell size is about 1.31 R_S in the corona and about 7 R_S in the inner heliosphere. And the most densely refined component can have about 8800 blocks.

3 OpenCL programming model and performance

Figure 2 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 timestep, 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

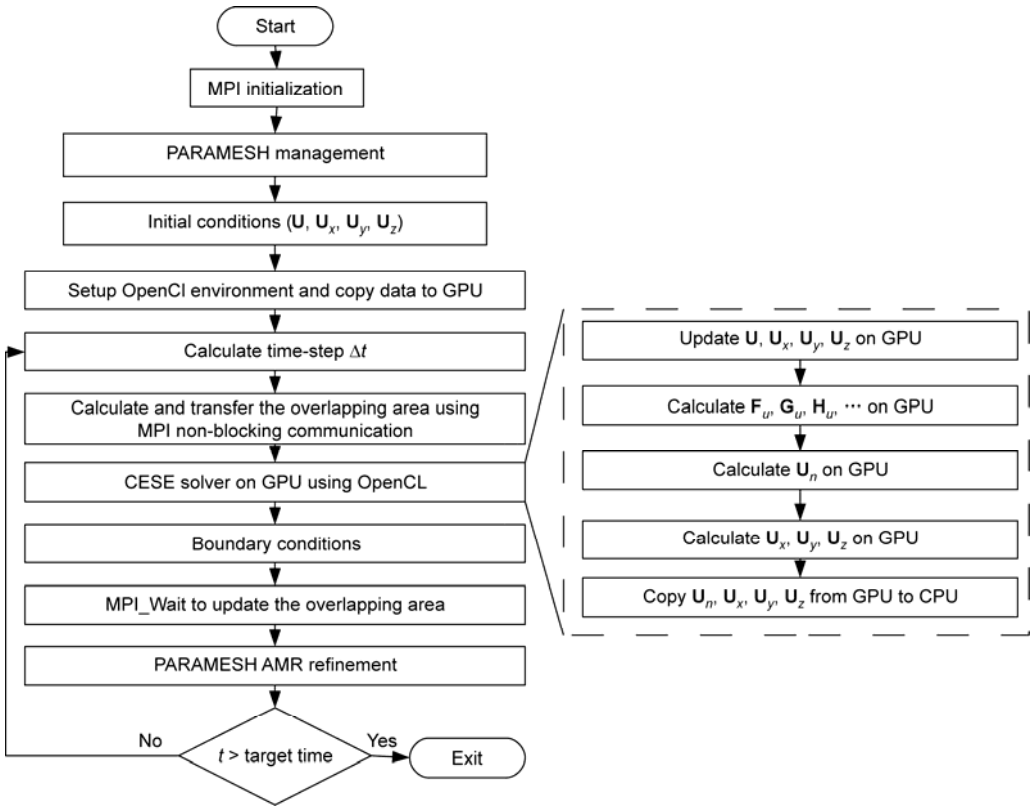

Figure 2 The flowchart of GPU implementation of SIP-CESE MHD model.

conditions and MPI communication on CPU.

As mentioned above, 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. In the meanwhile, the boundary or internal border value in the overlapping area of inter-components needs to be determined by an interpolation from its neighbor component grids according to the related geometrical positions of component grids (Feng et al., 2010). 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 operate in parallel with the MPI non-blocking communication. First, the CPUs compute the values in the overlapping area of inter-components, and then send or receive data using MPI non-blocking communication, at the same time the GPUs compute ($\mathbf{U}_n$, $\mathbf{U}_{xn}$, $\mathbf{U}_{yn}$, $\mathbf{U}_{zn}$). Afterwards, using MPI-Wait synchronizes all processes. Last, update ($\mathbf{U}$, $\mathbf{U}_x$, $\mathbf{U}_y$, $\mathbf{U}_z$) in the whole domain and compare the current time with the target time. With this implementation of overlapping the computation of the inner portion with the memory transfer of the inter-components, speed-up ratio can be improved. Figure 3 shows a partial host-side code that implements the SIP-CESE MHD model on the six-component grid.

The computational domain is first covered by root blocks with the lowest spatial resolution. Then, according to the user-defined refinement criteria as done by Feng et al. (2012a), each root block may be refined into eight child blocks with a spatial resolution twice that of their parent block. The same refinement operation may further be applied to all blocks in different refinement levels. As usual, each block has a fixed number of $8 \times 8 \times 8$ cells. (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 4, the mapped workspace on 24 GPUs must guarantee the coverage of the whole computational nodes and each process must have the

```
Set up OpenCL environment and copy data to GPU
while (time < target time)
{
    Calculate time-step Δt according to the Courant-Friedrichs-Lewy stability condition
    //for each time step
    for (step = 1; step <= maxstep; step++)
    {
        Calculate U_n, U_xn, U_yn and U_zn of overlapping area on CPU
        MPI_Isend(U_n, U_xn, U_yn, U_zn, ···);
        MPI_Irecv(U_n, U_xn, U_yn, U_zn, ···);
        //launch kernel to compute the flux (F, G, H, ···), source term S,
        //and Jacobi matrix (F_U, G_U, H_U, S_U, ···) in the CESE solver.
        status = clEnqueueNDRangeKernel(commandQueue, kernel_Flux, 3,
                 Null, globalworksize, localworksize, 0, Null, Null);
        //launch kernel to compute U_n
        status = clEnqueueNDRangeKernel(commandQueue, kernel_Un, 3,
                 Null, globalworksize, localworksize, 0, Null, Null);
        //launch kernel to compute U_xn, U_yn, U_zn
        status = clEnqueueNDRangeKernel(commandQueue, kernel_U_xyz, 3,
                 Null, globalworksize, localworksize, 0, Null, Null);

        Copy U_n, U_xn, U_yn, U_zn from GPU to CPU
        PARAMESH amr_guardcell;
        MPI_Wait(&req[i]; &status);
    }
    for all U = U_n, U_x = U_xn, U_y = U_yn, U_z = U_zn;
    Boundary Conditions;
    PARAMESH AMR refinement;
}
```

Figure 3 Partial host-side code that implements the SIP-CESE MHD model on the six-component grid. The outer loop is used for time stepping while the inner loop is for the solution of the CESE solvers. CPU process is created for each available GPU and executes the code above.

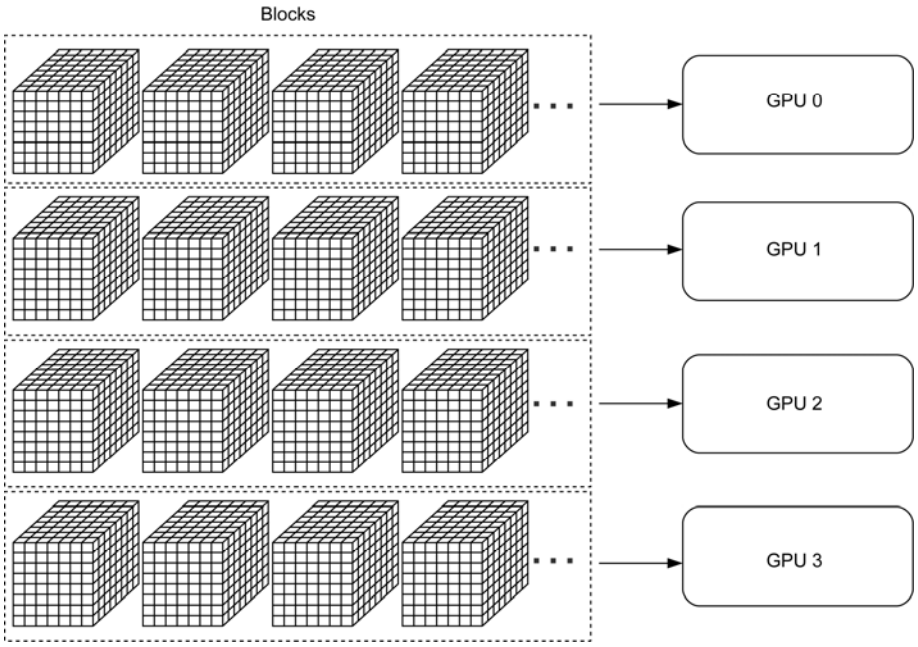

Figure 4 Mapping of blocks to GPUs.

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 × 8 × 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).

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 options "-O3". Table 1 shows a preliminary performance comparison between the CPU and GPU of major calculations 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 timestep between the CPU and GPU in double precision in global computation domain shows that using 24 MPI processes without GPU takes us 85.88 s, while 24 MPI processes with GPU 17.54 s give us about 5× speedup. The present OpenCL code using 24 MPI processes on CPU/GPU cluster finishes with wall time 30 h to reach a steady-state at the physical time 180 h for one CR. However, with the use of 96 MPI CPU processes for one CR it takes 36 h wallclock time.

4 Numerical results

In this section, we present test simulations of the solar wind background in CRs 2058 and 2062 to demonstrate that the SIP-CESE AMR MHD model run on the GPU/CPU cluster in double precision can greatly reduce the computing hours without any loss of accuracy.

CR 2058 lasted from June 21 to July 17, 2007 and CR 2062 from October 8 to November 4, 2007. Both CRs were near the most recent solar minimum, which had many unusual properties. The sunspot numbers were at their lowest in 75 years (Gibson et al., 2009) and the arithmetic mean polar field strength during the 2008 minimum was only two thirds of that during the 1976, 1986, and 1996 sunspot minima. The solar minimum also observed 15 percent smaller polar hole areas than those at the beginning of SC 23 and larger and more long-lived mid-latitude and low-latitude coronal holes (MLCHs) between ±40° latitudes around the solar equator. Other distinct features include multiple white-light coronal streamers, peculiar heliospheric current sheets (HCSs) of significantly high inclination (de Toma et al., 2010), the sparser and cooler fast solar wind (McComas et al., 2008) and the smaller average radial component of IMF (Smith et al., 2008). As we see below, our model run on the GPU/CPU cluster can reproduce most of these peculiarities.

4.1 Steady Corona

Figure 5 displays the synoptic maps of coronal observations from the SOHO/EIT 195 Å observations (upper panels) and from the model solution (lower panels) for CRs 2058 and 2062. The modeled coronal holes (the open-field regions) in the lower panels are shaded white and the other regions (the closed-field regions) green. The open-field and closed-field regions are determined by tracing the magnetic field lines from 6 R_S back to the photosphere. In the observation maps,

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

Block size	Operations (including data copying between GPU and CPU)	CPU (Intel® Xeon® CPU E5620@2.40 GHz)	GPU (C2050)	Ratio
48	Flux ($\mathbf{U}_t$, $\mathbf{F}_U$,...)	0.14	0.029	4.81
	$\mathbf{U}_n$	0.32	0.039	8.08
	$\mathbf{U}_x$, $\mathbf{U}_y$, $\mathbf{U}_z$	0.078	0.013	5.99
	all	0.54	0.081	6.66
192	Flux ($\mathbf{U}_t$, $\mathbf{F}_U$,...)	0.56	0.075	7.42
	$\mathbf{U}_n$	1.29	0.17	7.68
	$\mathbf{U}_x$, $\mathbf{U}_y$, $\mathbf{U}_z$	0.68	0.078	8.66
	all	2.53	0.33	7.76
384	Flux ($\mathbf{U}_t$, $\mathbf{F}_U$,...)	1.12	0.14	8.10
	$\mathbf{U}_n$	2.38	0.31	7.58
	$\mathbf{U}_x$, $\mathbf{U}_y$, $\mathbf{U}_z$	1.45	0.16	8.92
	All	4.95	0.61	8.11

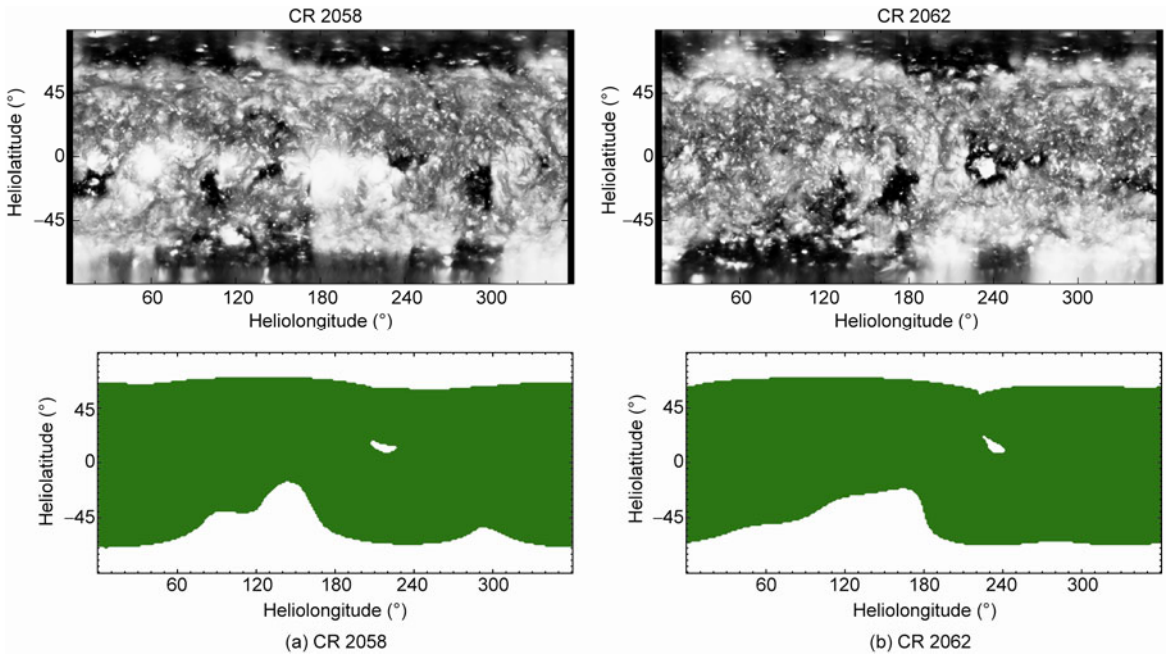

Figure 5 Synoptic maps of the coronal holes at 1 R_S for CRs 2058 (left) and 2062 (right). The first row is the SOHO/EIT 195 Å observation, in which the black vertical stripes represent data gaps. The second row presents the open-field regions and closed-field ones from the SIP-AMR-CESE MHD model, where the green shaded areas are of closed magnetic field and the white regions are of open field.

coronal holes are characteristic of low emission represented by dark color due to the continuous expansion of the solar wind. We can see that the equatorward boundaries of the northern polar coronal holes ranges between 60°–73°N for CRs 2058 and 2062 from both the observations and the model results. As for the southern polar holes, there is one extending coronal hole around Longitude 160° for each CR seen from the simulated results and the observations as shown in Figure 5.

Mid-latitude or equatorial holes were a special observational feature during the 2008 solar minimum and attracted the interests of many solar physicists and heliophysicists (e.g., Riley et al. (2012a); Gibson et al., 2009; de Toma et al., 2010; McComas et al., 2008; Lee et al., 2009; Yang et al., 2011; Abramenko et al., 2010). Obviously, the model captures the isolated equatorial hole (IEH) centered at (θ, ϕ) = (10°, 225°) for both CRs, but the size of the modeled IEH is smaller than that observed. Moreover, the model result misses the IEHs centered at (θ, ϕ) = (−20°, 0°) and (θ, ϕ) = (−20°, 290°) for CR 2058. These discrepancies result from the inherent deficiency of the MHD model with the potential field solution based on the photospheric magnetograms as the bottom boundary conditions, which can be confirmed by the previous study on the steady solar wind background of CR 2055 (Feng et al., 2012b) and Year 2007 (Yang et al., 2012), where the MHD simulation was driven by daily-updated magnetic field synoptic data. Feng et al. (2012c) constructed a data-driven model for the study of dynamic evolution of the global corona from September 4, 1996 to October 29, 1996 that can respond continuously to the changing of the photospheric magnetic field, where the global time-varying and self-consistent synchronic snapshots of the photospheric magnetic field produced by a surface flux transport model is used to drive the 3D numerical global coronal AMR-CESE-MHD model (Feng et al., 2012a) on an overset grid of Yin-Yang overlapping structure. They found that the data-driven MHD model achieved better results than the model with bottom radial magnetic field fixed in the frame corotating with the Sun, which is promising but there is still a long way to go in view of practical prediction. To sum up, our simulation has roughly captured the main coronal observations.

In Figure 6, we present the synoptic maps of the observations and simulated results for CRs 2058 (left column) and 2062 (right column). The maps of white-light polarized brightness (pB) at the east and west limbs observed by SOHO/LASCO-C2 are shown in Rows 1 and 2, where 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. Rows 3 and 4 in Figure 6 display the synoptic maps of the proton number density overlaying the isolines of the radial magnetic field and the radial velocity on the surface of 2.5 radii. The solid lines in Rows 1 and 2 and the dashed lines in Row 4 represent the magnetic neutral line, where the radial component of the magnetic field is null.

Figure 6 shows that both the MHD and PFSS models gave almost the same magnetic neutral line (MNL) characterized by a high peak roughly at ϕ = 150°, a low one at ϕ = 310°, and a trough centered at ϕ = 240°. Figure 6 reveals

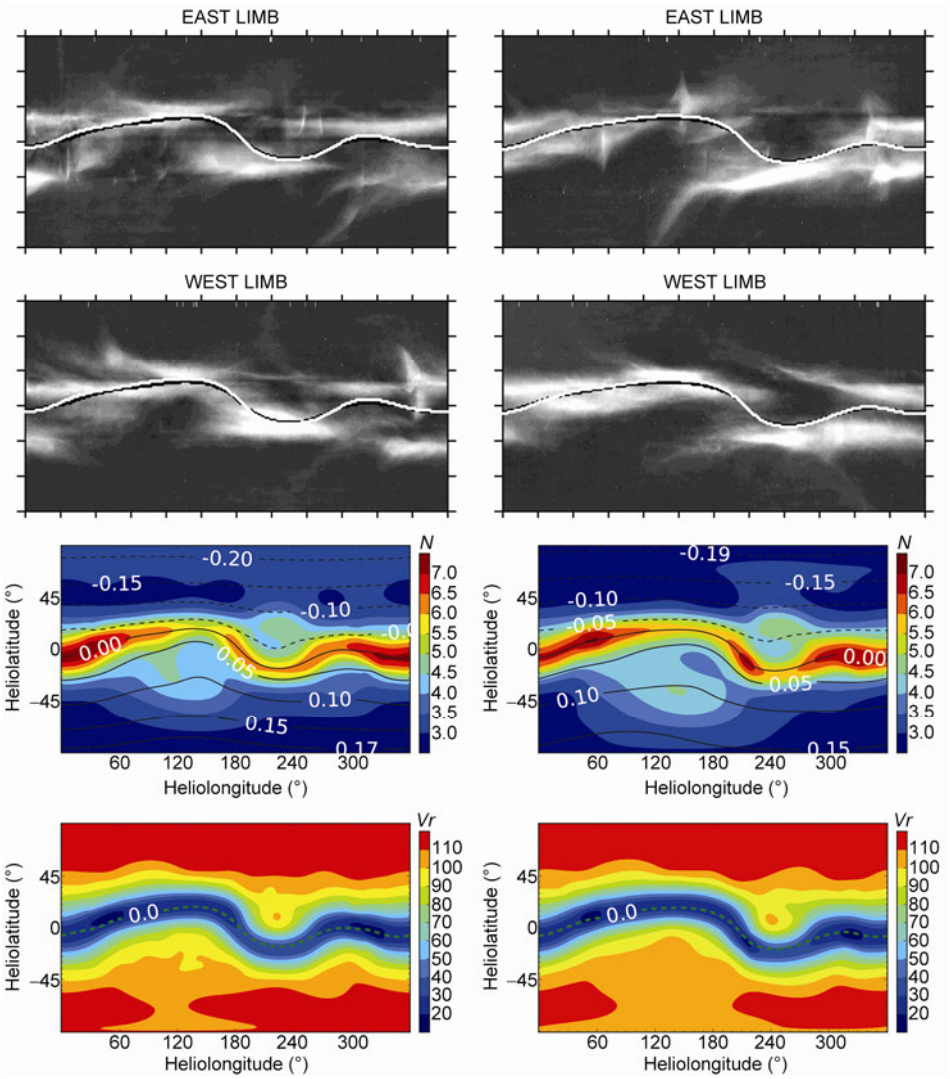

Figure 6 Synoptic maps at 2.5 R_S for CRs 2058 (left) and 2062 (right). The first and second rows are the white-light polarized brightness at the east and west limbs from SOHO/LASCO-C2. In the first and second rows, the black lines denote the magnetic neutral lines from the MHD model and the white lines from the PFSS model. The third row 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 fourth rows are the pseudo-color images of the radial speed v_r (km/s). In Row 3, the dashed lines stand for inward radial magnetic field and the solid ones outward radial magnetic field. The dashed lines in the last row denote the magnetic neutral lines.

that the MNL is coincident with the locations of the bright structures in SOHO/LASCO-C2 pB observations and surrounded by the regions of high-density and low-speed plasma flow. It should be noted that pB-enhanced regions are also present near (θ, ϕ) = (–20°, 160°) in CR 2058 and (θ, ϕ) = (–20°, 220°) in CR 2062, which are away from MNL. This phenomenon is believed to be associated with uniploar streamers (Yang et al., 2012; Riley et al., 2012b).

Figure 7 presents the white-light pB images from 2.3 to 6 R_S recorded by SOHO/LASCO-C2 (upper panels) and synthesized from the modeled results (lower panels) at ϕ = 0°–180° for CRs 2058 and 2062. The simulated magnetic field topologies projected on the meridional planes from 1 to 6 R_S are exhibited in Row 1 of Figure 8 and the colors of the points on the force line are coded by the magnitude of the radial speed. Rows 2 and 3 in Figure 8 present the simulated radial solar-wind speed from 1 to 13 R_S on the meridional plane and the simulated current sheet from 1 to 6 R_S.

Combining Figures 7 and 8, we can find that most observed bright structures on both limbs of Figure 7 can be explained to be the effect of high-density regions around the MNL or there are high-density regions near the sky plane. However, the south bright structures on the west limb for CR 2062 are probably associated with the abrupt change of the MNL and the edges of the unipolar streamers or pseudo-streamers roughly centered at Longitude 220°. Unipolar streamers separate holes of the same polarity, and hence

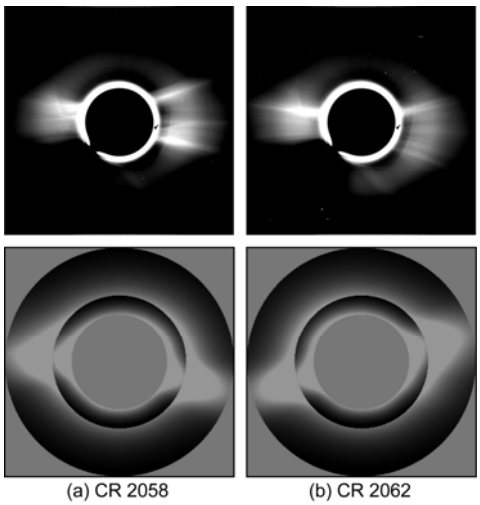

Figure 7 The coronal observations and the simulation results on the meridional plane at $\phi = 0°–180°$ for CRs 2058 (left) and 2062 (right). The first and second rows are the white-light pB images from 2.3 to 6 R_S recorded by SOHO/LASCO-C2 and computed from the simulation, respectively.

contain a double loop structure and emanate low-speed solar wind in interplanetary space (Riley et al., 2012b).

4.2 Interplanetary solar wind

Now we focus our attention on the solar wind structures in interplanetary space and the in-situ measurements. We present the model solutions on the meridional plane and at the spheres of 20 R_S and 215 R_S in Figure 9 and compare the modeled temporal profiles of the solar wind parameters with the *in-situ* measurements in Figure 10. Figure 9 shows that the HCSs for both CRs have a peak and a trough and are surrounded by plasma flow of low speed and high mass flux. For CR 2058, the peak is roughly centered on at $\phi = 150°$ at 2.5 Rs shifts to $\phi = 140°$ at 20 R_S and $\phi = 90°$ at 215 R_S due to the solar rotation, while the trough at $\phi = 230°$ shifts to $\phi = 220°$ at 20 R_S and $\phi = 180°$ at 215 R_S. The structure of the HCS for CR 2062 is similar to that for CR 2058, but the longitudes of the peak and the trough increase by about 15°–20°. Additionally, the highest mass-flux regions are coincident with the locations of HCS and the latitudinal width of the low-speed solar wind ($Vr < 550$ km/s) is about 50°–65° for both CRs, which is 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 et al., 2010; Yang et al., 2011). In

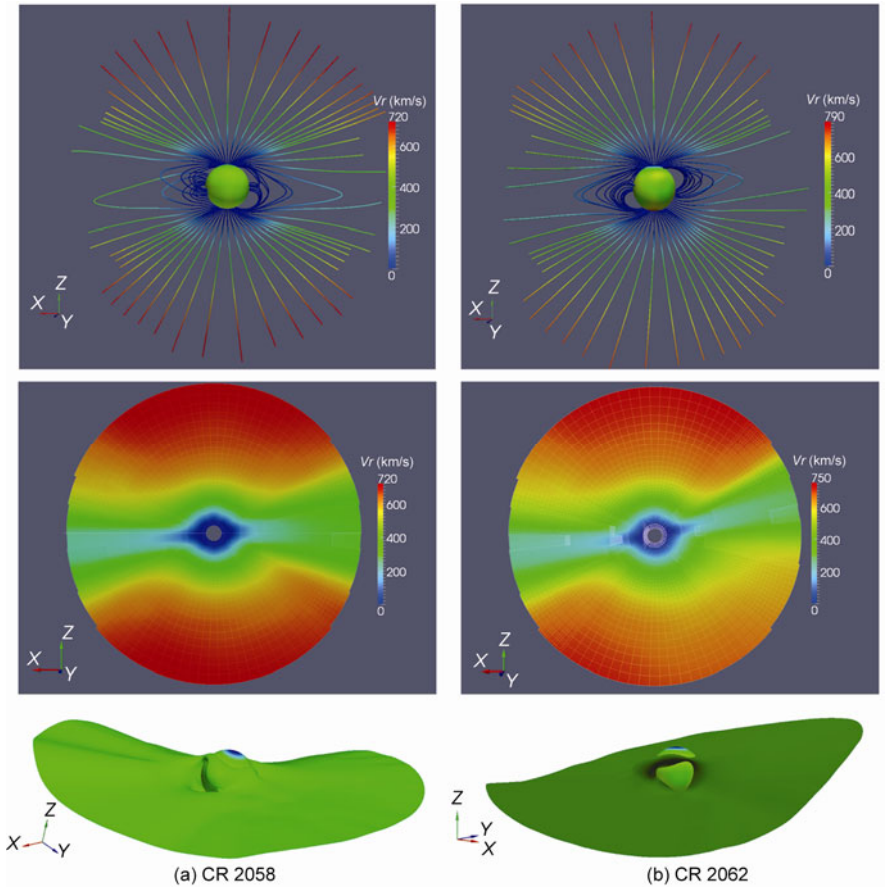

Figure 8 The coronal observations and the simulation results on the meridional plane at $\phi = 0°–180°$ for CRs 2058 (left) and 2062 (right). The first row is the simulated magnetic field topology projected on the meridional planes from 1 to 6 R_S. The color bars on the magnetic field lines stand for the magnitude of radial speed. The second row is the simulated radial solar-wind speed on the meridional planes from from 1 to 6 R_S, where the white quadrilaterals denote the grid blocks. The last row exhibits the simulated current sheets from 1 to 6 R_S.

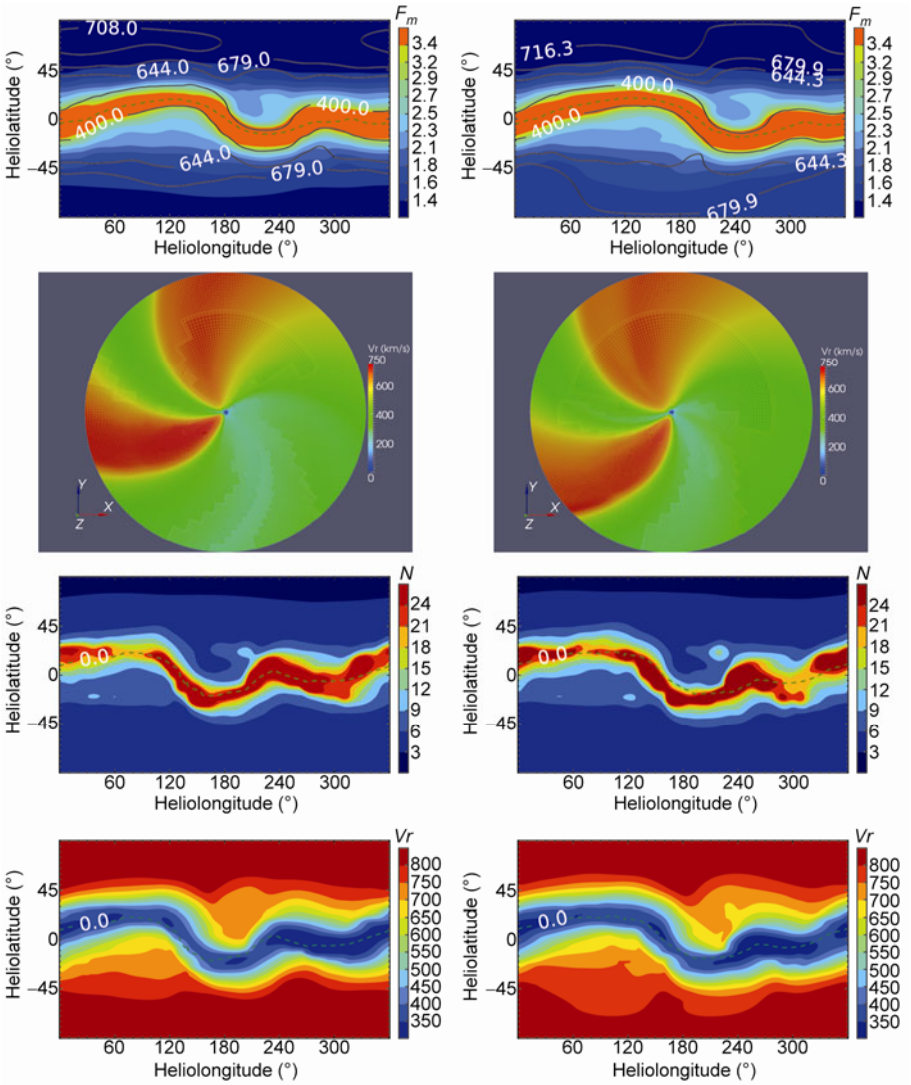

Figure 9 The simulated steady solution in interplanetary space from the MHD model for CRs 2058 (left) and 2062 (right). The isolines of the radial speed *Vr* (km/s) superimposed on the synoptic pseudo-color image of the mass flux density *Fm* (10^8 km s^{-1} cm^{-3} R_S) at 20 R_S (Row 1), the pseudo-color images of the radial velocity in the solar equatorial plane (Row 2), the synoptic contours of the simulated proton number density *N* (cm^{-3}) (Row 3) and *Vr* (km/s) (Row 4) at 215 R_S. In Rows 1, 3 and 4, the dashed lines denote the magnetic neutral lines.

addition, the solar winds in the equatorial plane for both CRs consist of two low-speed streams and two high-speed streams.

Figure 10 shows that the model solutions have basically captured the stream structures for the two CRs. Both the observed and modeled temporal profiles of the solar wind speed exhibit an initial long-duration of slow solar wind followed by a strong and steep high-speed stream on Day 12 of CR 2058 and Day 9 of CR 2062. The second modeled high-speed stream rises on Day 18 of CR 2058 and Day 17 of CR 2062, but declines so slowly that the solution misses the low-speed stream between the second and third high-speed streams for both CRs. In addition, the modeled polarity of the radial magnetic field for CR 2058 switches from inward to outward on Day 14, two days earlier than the *in-situ* measurements, while for CR 2062, it changes from inward to outward on Day 17, one day later than the observation. Generally speaking, the model reproduces the observed polarity of the radial magnetic field with a reasonable accuracy.

5 Conclusions

The SIP-CESE MHD 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 5× compared to traditional software implementation. This work presents a novel application of GPU to the space weather study.

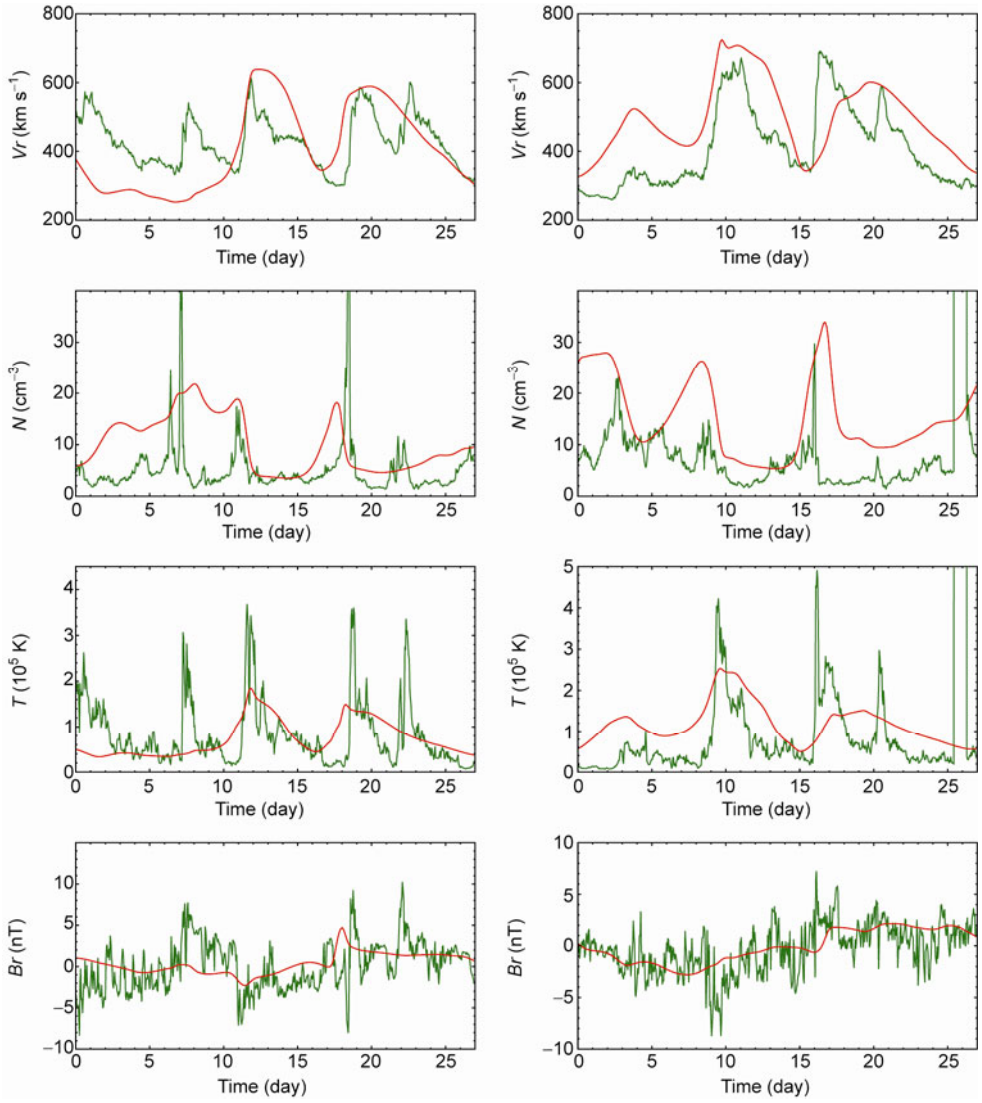

Figure 10 The calculated MHD steady state at 1 AU for CRs 2058 (left) and 2062 (right). The first, second, third, fourth rows are the comparisons between the MHD results and the one-hour averaged OMNI data near 1 AU for radial solar-wind speed *Vr*, number density *N*, temperature *T*, and radial magnetic field *Br*, respectively, where the green lines denote the observations and the red lines represent the numerical results.

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 5× (at least) without any optimization. However, the data transfer between the CPU and the GPU or between computers in the multi GPU technique incurs large costs. For such simulation run with a large amount of data input and output (I/O), the wall clock time spent on I/O operations can easily exceed the time needed for the calculations. In our code, the time spent transferring data using MPI among GPUs, guardcell filling and mesh refinement is 30% more of magnitude than that of calculations using GPUs, since we have at least 32 variables ($\mathbf{U}$, $\mathbf{U}_x$, $\mathbf{U}_y$, $\mathbf{U}_z$) stored and transferred.

A direct improvement at hand is to further consider the following aspects: (1) load balance and memory management in MPI will be left for next consideration, and (2) spatial resolution needs to be further refined. Our grid resolution far from the Sun is still low, since the existence of a wide range in HCS width at 1 AU is typically agreed to be from 40000 to 100000 km (Blanco et al., 2006; Behannon et al., 1981), a typical size of grid cell at 1 AU with a resolution of 0.057 R_S is required, in order to be comparable to the

thickness of HCS (heliospheric current sheets).

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 precision can reproduce the shapes and distributions of the polar coronal holes and the presences of the equatorial holes observed by SOHO/EIT and LASCO C2. It can also achieve the basically consistent temporal profiles of solar wind parameter at 1AU with the in-situ measurements. On the other hand, we should also note that there are some discrepancies between the numerical results and observations, such as the areas of the equatorial holes, the magnitudes of radial magnetic fields and arrival times of the high-speed streams at the Earth. In order to achieve better numerical results, which can capture the structures of the heliosphere during specific time periods more accurately, we should employ high time-cadence photospheric magnetograms to drive the model just as done by Riley et al. (2012a), Feng et al. (2012b, c), and Yang et al. (2012). Other considerations may include using synoptic maps from different observatories and choosing the solution that best matches the observations, and driving the numerical model by using the synoptic maps from the Air Force Data Assimilative Photospheric flux Transport model (Henney et al., 2012; Lee et al., 2012; Arge et al., 2010, 2011), which can assimilate different observations into surface flux model and thus provide more instantaneous snapshots of the global photospheric field distribution than traditional methods. The ADAPT model, together with high-quality observations from SOHO/MDI and SDO/HMI (Liu et al., 2012), will improve the input and results of the MHD simulations.

The work was jointly supported by the National Natural Science Foundation of China (Grant Nos. 41031066, 41231068, 41274192, 41074121 & 41074122), the National Basic Research Program of China (Grant No. 2012CB825601), the Knowledge Innovation Program of the Chinese Academy of Sciences (Grant No. KZZD-EW-01-4), and the Specialized Research Fund for State Key Laboratories. The numerical calculation has been completed on our SIGMA Cluster computing system. The PARAMESH software used in this work was developed at the NASA Goddard Space Flight Center and Drexel University under NASA's HPCC and ESTO/CT projects and under grant NNG04GP79G from the NASA/AISR project. Wilcox Solar Observatory data used in this study were obtained via http://wso.stanford.edu. The Wilcox Solar Observatory is currently supported by NASA. SOHO is a project of international cooperation between ESA and NASA. The OMNI data are obtained from the GSFC/SPDF OMNIWeb interface http://omniweb.gsfc.nasa.gov.

Abramenko V, Yurchyshyn V, Linker J, et al. 2010. Low-latitude coronal holes at the minimum of the 23rd solar cycle. Astrophys J, 712: 813–818

Arge C N, Pizzo V J. 2000. Improvement in the prediction of solar wind conditions using near-real time solar magnetic field updates. J Geophys Res, 105: 10465–10479

Arge C N, Odstrcil D, Pizzo V J, et al. 2003. Improved method for specifying solar wind speed near the sun. In: Velli M, Bruno R, Malara F, et al., eds. Solar Wind Ten, American Institute of Physics Conference Series, 679: 190–193

Arge C N, Henney C J, Koller J, et al. 2010. Air Force data assimilative photospheric flux transport (ADAPT) model. Twelfth International Solar Wind Conference, AIP Conference Proceedings, 1216: 343–346

Arge C N, Henney C J, Koller J, et al. 2011. Improving data drivers for coronal and solar wind models. In: Pogorelov N V, Audit E, Zank G P, eds. 5th International Conference of Numerical Modeling of Space Plasma Flows (ASTRONUM 2010). Astron Soc Pacific Conf Ser, 444: 99–104

Aschwanden M, Burlaga L F, Kaiser M L, et al. 2008. Theoretical modeling for the STEREO mission. Space Sci Rev, 136: 565–604

Baker D N. 2002. How to cope with space weather. Science, 297: 1486–1487

Baker D N, Wiltberger M J, Weigel R S, et al. 2007. Present status and future challenges of modeling the Sun Earth end-to-end system. J Atmos Sol-Terr Phys, 69: 3–17

Behannon K W, Neubauer F M, Barnstorf H. 1981. Fine-scale characteristics of interplanetary sector boundaries. J Geophys Res, 86: 3273–3287

Belleman R G, Bdorf J, Zwart S F P. 2008. High performance direct gravitational n-body simulations on graphics processing units II: An implementation in CUDA. New Astron, 13: 103–112

Blanco J J, Rodríguez-Pacheco J, Hidalgo M A, et al. 2006. Analysis of the heliospheric current sheet fine structure: Single or multiple current sheets. J Atmos Sol-Terr Phys, 68: 2173–2181

Bothmer V, Daglis I A. 2007. Space Weather: Physics and Effects. Chichester: Praxis Publishing. 1

Brandvik T, Pullan G. 2008. Acceleration of a 3D Euler solver using commodity graphics hardware. AIAA 2008-607, 46th AIAA Aerospace Sciences Meeting and Exhibit. January 2008, Reno, Nevada

Che S, Boyer M, Meng J, et al. 2008. A performance study of general-purpose applications on graphics processors using CUDA. J Parallel Distrib Comput, 68: 1370–1380

Cohen O, Sokolov I V, Roussev I I, et al. 2008. Validation of a synoptic solar wind model. J Geophys Res, 113: A03104

De Zeeuw D L. 1993. A quadtree-based adaptively-refined Cartesian-grid algorithm for solution of the Euler equations. Doctoral Dissertation. Ann Arbor: The University of Michigan

DeLeon R, Senocak I. 2012. GPU-accelerated large-eddy simulation of turbulent channel flows. AIAA 2012-0722, 50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition 09–12 January 2012, Nashville, Tennessee

Dryer M. 2007. Space weather simulation in 3D MHD from the Sun to the Earth and beyond to 100 AU: A modeler's perspective of the present state of the art. Asian J Phys, 16: 97–121

Elsen E, Legresley P, Darve E. 2008. Large calculation of the flow over a hypersonic vehicle using a GPU. J Comp Phys, 227: 10148–10161

Feng X S, Zhou Y F, Wu S T. 2007. A novel numerical implementation for solar wind modeling by the modified conservation element/solution element method. Astrophys J, 655: 1110–1126

Feng X S, Yang L P, Xiang C Q, et al. 2010. Three-dimensional solar wind modeling from the Sun to Earth by a SIP-CESE MHD model with a six-component grid. Astrophys J, 723: 300

Feng X S, Xiang C Q, Zhong D K. 2011a. The state-of-art of three-dimensional numerical study for corona-interplanetary process of solar storms (in Chinese). Sci Sin-Terrae, 41: 1–28

Feng X S, Zhang S H, Xiang C Q, et al. 2011b. A hybrid solar wind model of the CESE+HLL method with a Yin-Yang overset grid and an AMR grid. Astrophys J, 734: 50

Feng X S, Yang L P, Xiang C Q, et al. 2012a. Validation of the 3D AMR SIP-CESE solar wind model for four Carrington rotations. Solar Phys, 279: 207–229

Feng X S, Yang L P, Xiang C Q, et al. 2012b. Numerical study of the global corona for CR 2055 driven by daily updated synoptic magnetic field. In: Pogorelov N V, Font J A, Audit E, et al., eds. Numerical Modeling of Space Plasma Slows (ASTRONUM 2011). Astron Soc Pacific Conf Ser, 459: 202–208

Feng X S, Jiang C W, Xiang C Q, et al. 2012c. A data-driven model for the global coronal evolution. Astrophys J, 758: 62

Gaburov E, Harfst S, Zwart S F P. 2009. SAPPORO: A way to turn your graphics cards into a GRAPE-6. New Astron, 14: 630–637

Gibson S E, Kozyra J U, de Toma G, et al. 2009. If the Sun is so quiet, why is the Earth ringing? A comparison of two solar minimum intervals. J Geophy Res, 114: A09105

Henney C J, Toussaint W A, White S M, et al. 2012. Forecasting F10.7 with solar magnetic flux transport modeling. Space Weather, 10: S02011

van der Holst B, Keppens R. 2007. Hybrid block-AMR in Cartesian and curvilinear coordinates: MHD applications. J Comput Phys, 226: 925–946

van der Holst B, Manchester W B, Frazin R A, et al. 2010. A data-driven, two-temperature solar wind model with Alfvén waves. Astrophys J, 25: 1373–1383

Hu Y Q, Guo X C, Wang C. 2007. On the ionospheric and reconnection potentials of the earth: Results from global MHD simulations. J Geophy Res, 112: A07215

Jacobsen D A, Thibault J C, Senocak I. 2010. An MPI-CUDA implementation for massively parallel incompressible flow computations on multi-GPU clusters. AIAA 2010–522, 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace 4–7 January 2010, Orlando, Florida

Janhunen P, Koskinen K E J, Pulkinen T I. 1996. A new global ionosphere-magnetosphere coupling simulation utilizing locally varying time step. In: Rolfe E J, Kaldeich B, eds. International Conference on Substorms. ESA Special Publication 389, 205–210

Jiang C W, Feng X S, Zhang J, et al. 2010. AMR simulations of magnetohydrodynamic problems by the CESE method in curvilinear coordinates. Solar Phys, 267: 463–491

Jin M, Manchester W B, van der Holst B, et al. 2012. A global two-temperature corona and inner heliosphere model: A comprehensive validation study. Astrophys J, 745: 6

Kestener P, Chateau F, Teyssier R. 2010. Accelerating Euler equations numerical solver on graphics processing units. In: Hsu C H, Yang L, Park J, et al, eds. Algorithms and Architectures for Parallel Processing Lecture Notes in Computer Science 6082. Berlin/Heidelberg: Springer, 281–288

Kleimann J. 2012. 4π models of CMEs and ICMEs (invited review). Solar Phys, 281: 253–267

Lee D, Deane A E. 2009. An unsplit staggered mesh scheme for multidimensional magnetohydrodynamics. J Comput Phys, 228: 952–975

Lee C O, Arge C N, Odstrćil D, et al. 2012. Ensemble modeling of CME propagation. Solar Phys, doi: 10.1007/s11207-012-9980-1

Lionello R, Linker J A, Mikić Z. 2009. Multispectral emission of the Sun during the first whole Sun month: Magnetohydrodynamic simulations. Astrophys J, 690: 902–912

Liu Y, Hoeksema J T, Scherrer P H, et al. 2012. Comparison of line-of-sight magnetograms taken by the Solar Dynamics Observatory/Helioseismic and Magnetic Imager and Solar and Heliospheric Observatory/Michelson Doppler Imager. Solar Phys, 279: 295–316

Lugaz N, Downs C, Shibata K, et al. 2011. Numerical investigation of a coronal mass ejection from an Anemone active region: Reconnection and deflection of the 2005 August 22 eruption. Astrophys J, 738: 127–139

Lyon J, Fedder J, Mobarry C. 2004. The Lyon-Fedder-Mobarry (LFM) global MHD magnetospheric simulation code. J Atmos Sol-Terr Phys, 66: 1333–1350

Lyon J G. 2000. The solar wind-magnetosphere-ionosphere system. Science, 288: 1987–1991

Marder B. 1987. A method for incorporating Gauss' law into electromagnetic PIC codes. J Comput Phys, 68: 48–55

McComas D J, Ebert R W, Elliott H A, et al. 2008. Weaker solar wind from the polar coronal holes and the whole Sun. Geophys Res Lett, 35: L18103

Michalakes J, Vachharajani M. 2008. GPU acceleration of numerical weather prediction. Parallel Proc Lett, 18: 531–548

Mikić Z, Linker J A, Schnack D D, et al. 1999. Magnetohydrodynamic modeling of the global solar corona. Phys Plasmas, 6: 2217–2224

Nakamizo A, Tanaka T, Kubo Y, et al. 2009. Development of the 3-D MHD model of the solar corona-solar wind combining system. J Geophys Res, 114: A07109

Ogino T. 1986. A three-dimensional MHD simulation of the interaction of the solar wind with the Earth's magnetosphere—The generation of field-aligned currents. J Geophys Res, 91: 6791–6806

Ogino T. 2002. Three-dimensional global MHD simulation code for the Earth's magnetosphere using HPF/JA. Concurrency Comput: Pract Experience, 14: 631–646

Owens J D, Houston M, Luebke D, et al. 2008. GPU computing. Proc IEEE, 96: 879–899

Owens J D, Luebke D, Govindaraju N, et al. 2007. A survey of general-purpose computation on graphics hardware. Comp Graph Forum, 26: 80–113

Podgorny I M, Podgorny A I. 2005. Expansion of solar corona in the Sun's gravitational field and formation of the heliospheric current sheet. Inter J Geomagn Aeron, 6: GI1005

Powell K G, Roe P L, Linde T J, et al. 1999. A solution-adaptive upwind scheme for ideal magnetohydrodynamics. J Comput Phys, 154: 284–309

Raeder J, Wang Y, Fuller-Rowell T, et al. 2001. Global simulation of magnetospheric space weather effects of the Bastille Day storm. Solar Phys, 204: 323–337

Riley P, Linker J A, Mikić, Z, et al. 2006. A comparison between global solar magnetohydrodynamic and potential field source surface model results. Astrophys J, 653: 1510–1516

Riley P, Lionello R, Linker J A, et al. 2011. Global MHD modeling of the solar corona and inner heliosphere for the whole heliosphere interval. Solar Phys, 274: 361–377

Riley P, Linker J A, Lionello R, et al. 2012a. Corotating interaction regions during the recent solar minimum: The power and limitations of global MHD modeling. J Atmos Sol-Terr Phys, 83: 1–10

Riley P, Stevens M, Linker J A, et al. 2012b. Modeling the global structure of the heliosphere during the recent solar minimum: Model improvements and unipolar streamers. In: Heerikhuisen J, Li G, Pogorelov N, et al, eds. American Institute of Physics Conference Series. Amer Institute Phys Conf Ser, 1436: 337–343

Roussev I I, Gombosi T I, Sokolov I V, et al. 2003. A three-dimensional model of the solar wind incorporating solar magnetogram observations. Astrophys J Lett, 595: L57–L61

Schenk O, Christen M, Burkhart H. 2008. Algorithmic performance studies on graphics processing units. J Parallel Distrib Comput, 68: 1360–1369

Schive H Y, Tsai Y C, Chiueh T. 2010. Gamer: A graphic processing unit accelerated adaptive-mesh-refinement code for astrophysics. Astrophys J Suppl Ser, 186: 457–484

Smith E J, Balogh A. 2008. Decrease in heliospheric magnetic flux in this solar minimum: Recent Ulysses magnetic field observations. Geophys Res Lett, 35: L22103

Taktakishvili A, Pulkkinen A, MacNeice P, et al. 2011. Modeling of coronal mass ejections that caused particularly large geomagnetic storms using ENLIL heliosphere cone model. Space Weather, 9: S06002

Tanaka T. 1994. Finite volume TVD scheme on an unstructured grid system for three-dimensional MHD simulation of inhomogeneous systems including strong background potential fields. J Comput Phys, 111: 381

Tanaka T. 1995. Generation mechanisms for magnetosphere-ionosphere current systems deduced from a three-dimensional MHD simulation of the solar wind-magnetosphere-ionosphere coupling processes. J Geophys Res, 100: 12057–12074

Thibault J C, Senocak I. 2009. CUDA implementation of a navier-stokes solver on multi-GPU desktop platforms for incompressible flows. AIAA 2009-758, 47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition 5–8 January 2009, Orlando, Florida

de Toma G, Arge C N. 2010. The Sun's magnetic field during the past two minima. Twelfth International Solar Wind Conference, AIP Conference Proceedings, 1216: 679–681

Tóth G, Sokolov I V, Gombosi T I, et al. 2005. Space weather modeling framework: A new tool for the space science community. J Geophys Res, 110: A12226

Tóth G, van der Holst B, Sokolov I V, et al. 2012. Adaptive numerical algorithms in space weather modeling. J Comput Phys, 231: 870–903

Usmanov A V, Goldstein M L. A three-dimensional MHD solar wind model with pickup protons. J Geophys Res, 2006, 111: A07101
Wang Y M, Sheeley N R Jr. 1990. Solar wind speed and coronal flux-tube expansion. Astrophys J, 355: 726–732
Wang P, Abel T, Kaehler R. 2010. Adaptive mesh fluid simulations on GPU. New Astron, 15: 581–589
Watermann J, Wintoft P, Sanahuja B, et al. 2009. Models of solar wind structures and their interaction with the Earth's space environment. Space Sci Rev, 147: 233–270
Webb D F, Howard T A. 2012. Coronal mass ejections: Observations. Living Rev Solar Phys, 9: 3
Wong H C, Wong U H, Feng X S, et al. 2011. Efficient magnetohydrodynamic simulations on graphics processing units with CUDA. Comp Phys Commun, 182: 2132–2160
Yang L P, Feng X S, Xiang C Q, et al. 2011. Simulation of the unusual solar minimum with 3D SIP-CESE MHD model by comparison with multi-satellite observations. Solar Phys, 2011, 271: 91–110
Yang L P, Feng X S, Xiang C Q, et al. 2012. Time-dependent MHD modeling of the global solar corona for year 2007: Driven by daily-updated magnetic field synoptic data. J Geophys Res, 117: A08110