Implementation of Coupled Numerical Analysis of Magnetospheric Dynamics and Spacecraft Charging Phenomena via Code-To-Code Adapter (CoToCoA) Framework

Yohei Miyake[1][0000−0001−6491−1012], Youhei Sunada[1], Yuito Tanaka[1], Kazuya Nakazawa[1], Takeshi Nanri[2], Keiichiro Fukazawa[3], and Yuto Katoh[4]

[1] Kobe University, Kobe 657-8501, Japan. y-miyake@eagle.kobe-u.ac.jp
[2] Kyushu University, Fukuoka 819-0395, Japan.
[3] Kyoto University, Kyoto 606-8501, Japan.
[4] Tohoku University, Sendai 980-8578, Japan.

Abstract. This paper addresses the implementation of a coupled numerical analysis of the Earth's magnetospheric dynamics and spacecraft charging (SC) processes based on our in-house Code-To-Code Adapter (CoToCoA). The basic idea is that the magnetohydrodynamic (MHD) simulation reproduces the global dynamics of the magnetospheric plasma, and its pressure and density data at local spacecraft positions are provided and used for the SC calculations. This allows us to predict spacecraft charging that reflects the dynamic changes of the space environment. CoToCoA defines three types of independent programs: Requester, Worker, and Coupler, which are executed simultaneously in the analysis. Since the MHD side takes the role of invoking the SC analysis, Requester and Worker positions are assigned to the MHD and SC calculations, respectively. Coupler then supervises necessary coordination between them. Physical data exchange between the models is implemented using MPI remote memory access functions. The developed program has been tested to ensure that it works properly as a coupled physical model. The numerical experiments also confirmed that the addition of the SC calculations has a rather small impact on the MHD simulation performance with up to about 500-process executions.

Keywords: Coupled Analysis Framework · Magnetospheric Dynamics · Spacecraft Charging · Space Plasma.

1 Introduction

Space plasmas involve complex dynamical processes characterized by a wide range of spatial and temporal scales [1]. One example is the coupling between the global-scale dynamics of the Earth's magnetosphere and micro-scale processes such as plasma wave excitation and energetic particle production. Although these phenomena are closely related through magnetic structures, the spatial and temporal scales that characterize them are very different. In numerical simulations

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_46

of such phenomena, the degree of coarseness in the calculation of each physical process varies greatly [2, 3]. Traditionally, most plasma simulation models have evolved through the development and refinement of individual physical models, each dedicated to monoscale elementary processes [4]. Nowadays, the "cross-scale coupling" is becoming a key topic of interest. There is a strong demand for simulation techniques that allow us to reproduce the evolution of multi-scale physical systems [5]. This should be achieved by running multiple physical models at different scales simultaneously, and coupling them via intercommunication of key physical quantities.

A standard approach is to embed micro-scale physical models within a macro-scale simulation. This approach is best suited for addressing problems where the physical information to be intercommunicated between the models is well defined and verified. Its implementation typically involves bidirectional and high-frequency exchange of physical information, and the coupled models advance their own computations in close synchronization with each other. In general, the approach requires significant development costs to complete the program, but if handled properly, high performance in terms of accuracy and computational efficiency can be expected [6].

There also remain areas of research, where the physical processes involved are themselves fundamental, but the details of how the two different phenomena are coupled are not yet clear or verified. For such targets, one might consider keeping the way of coupling simple (e.g., by assuming only one-way coupling), and wish to investigate the degree of influence of various physical factors by trial and error. The present study aims at such moderate coordination, where multiple governing equation systems are computed in a mostly asynchronous manner, while information is communicated between the models as needed. We have developed the prototype of a Code-To-Code Adapter (CoToCoA) to realize such moderate and flexible coupling, and started to apply it to some physical topics in the field of space plasma [7]. The implementation of a coupled physical system and its performance characteristics strongly depend on how tightly the target physical processes are coupled with each other. Our previous report dealt with a case study where coupled physical models were computed synchronously [7]. In this paper, as a new application case, we report the implementation of spacecraft charging (SC) calculations coupled with the dynamics of the Earth's magnetospheric environment in an asynchronous manner.

2 Physical Models to be Coupled

2.1 Target Physical Phenomena

The coupled physical phenomena addressed in this paper are outlined below. The global-scale model simulates the interaction between the solar wind plasma that is extended out from the Sun and the Earth's dipolar magnetic field. This interaction leads to the formation of the magnetosphere around the Earth. The Sun-facing side of the magnetosphere is compressed by the dynamic pressure of the solar wind, whereas its nightside extends far beyond the Moon's orbit,

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_46

forming a magnetotail. These physical processes are governed by the magnetohydrodynamic (MHD) equations. The MHD simulation reproduces the time evolution of the plasma macro-parameters over the Earth's magnetosphere.

The surface of a solid body such as a spacecraft collects surrounding plasma particles and gets electrically charged. Since such charging phenomena can lead to spacecraft malfunctions and even failures, this has been an active area of research since about 1980. So far, spacecraft charging has been studied under static plasma conditions, but this study aims to reproduce charging phenomena in a dynamic plasma environment. By extracting the parameter values at a spacecraft position from the above MHD data, the SC solver evaluates the rate of charge accumulation on the spacecraft per unit time (i.e., a current) based on a conventional theory. The ordinary differential equation of the spacecraft potential is then solved with the current values as source terms. This strategy would make it possible to predict SC transitions in conjunction with the dynamical changes in the magnetospheric environment.

2.2 Magnetospheric Simulation

The magneto-hydrodynamic (MHD) equations solved in the magnetospheric global simulation are given as follows:

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\boldsymbol{V}\rho) + D\nabla^2\rho, \tag{1}$$

$$\frac{\partial \boldsymbol{V}}{\partial t} = -(\boldsymbol{V}\cdot\nabla)\boldsymbol{V} - \frac{1}{\rho}\nabla p + \frac{1}{\rho}\boldsymbol{J}\times\boldsymbol{B} + \boldsymbol{g} + \frac{\boldsymbol{\Phi}}{\rho}, \tag{2}$$

$$\frac{\partial p}{\partial t} = -(\boldsymbol{V}\cdot\nabla)p - \gamma p\nabla\cdot\boldsymbol{V} + D_p\nabla^2 p, \tag{3}$$

$$\frac{\partial \boldsymbol{B}}{\partial t} = \nabla\times(\boldsymbol{V}\times\boldsymbol{B}) + \eta\nabla^2\boldsymbol{B}, \tag{4}$$

$$\boldsymbol{J} = \nabla\times(\boldsymbol{B}-\boldsymbol{B}_{\mathrm{d}}), \tag{5}$$

where ρ, p, $\boldsymbol{V}$, $\boldsymbol{J}$, and $\boldsymbol{B}$ are the plasma density, plasma pressure, velocity vector, current density, and magnetic field, respectively. D and D_p denote the diffusion coefficients of the plasma density and pressure, respectively, $\boldsymbol{g}$ is the gravity term, $\boldsymbol{\Phi}$ is the viscosity term, η is the temperature dependent electrical resistance, and γ is the specific heat constant. The dipole magnetic field $\boldsymbol{B}_{\mathrm{d}}$ is also incorporated to represent the Earth's intrinsic magnetic field.

In developing the coupled analysis, we adopted the MHD program developed by Fukazawa et al. [9]. The code discretizes a 3-dimensional simulation box and defines physical quantities on the Cartesian grid points. The Modified Leap Frog algorithm [8] is used for the time integration of the system equations. The code is fully parallelized and optimized for distributed-memory computer systems via the domain decomposition method. Since a so-called stencil calculation is performed at each position coordinate, the parallel implementation is based on boundary communication using MPI_Sendrecv at the outer edges of

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_46

the subdomains. The MHD code was evaluated for parallel performance on different computer systems. It achieved a parallel efficiency of 96.5% against 72,000 MPI parallelism on the Fujitsu PRIMERGY CX2550 supercomputer installed at the Kyushu University [10].

2.3 Spacecraft Charging Simulation

Spacecraft charging is generally characterized by the surface potential of the spacecraft with respect to space [11]. A spacecraft usually consists of several components. If the spacecraft is modeled as a multi-conductor electrostatic system, its charge can be described by the following ordinary differential equation.

$$\frac{\mathrm{d}\phi_i}{\mathrm{d}t} = \frac{1}{C_i} \sum_s I_s(\phi_i, n_0, T_s, \ldots) \quad (i = 1, \ldots, N_{\mathrm{sc\_components}}) \tag{6}$$

where ϕ_i and C_i are the potential and capacitance of the i-th spacecraft component, respectively. The I_s, n_0, and T_s are the spacecraft inflow current, plasma density, and plasma temperature of the s-th particle species, respectively. The current term on the right-hand side reflects the effect of charge accumulation due to the motion of charged plasma particles around the spacecraft.

As a rough classification, there are two types of approaches to determine the plasma current on the right-hand side of Eq. 6. One approach is the plasma particle simulation, in which the plasma current is evaluated by numerically tracing the trajectories of charged plasma particles around the spacecraft [12]. While this method allows for sophisticated calculations that take into account the geometric details of the spacecraft, it is better suited for calculations of micro-scale phenomena with short time scales (msec at most). Therefore, it is not easy to work directly with magnetospheric simulations covering long time scales (hours to days).

Another approach is the use of quasi-analytical formulations that relate macro-parameters such as the density and temperature of the surrounding plasma to the values of the plasma current [13]. Although in principle this approach works well only for simple geometries such as spheres and cylinders, in practice it is known to provide a relatively good approximation when the spacecraft size is similar to or smaller than the Debye length. The most important feature is that it is much less computationally expensive than plasma particle simulations. Based on this trait, this quasi-analytical approach is used in the coupled analysis addressed in this study, which facilitates its coupling with MHD simulations for long-term magnetospheric environmental variations. Specifically, the current term on the right-hand side of Eq. 6 is calculated from the plasma density and temperature data at each time. The equation is then numerically integrated by the fourth-order Runge-Kutta method to compute the time evolution of the spacecraft potential.

In the quasi-analytical approach described above, the spacecraft potential is not expressed as a function of spatial coordinates, but only of time. Its paral-

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_46

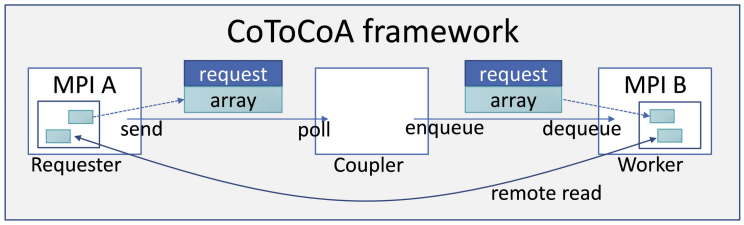

Fig. 1. Overview of code coupling via CoToCoA

lel implementation is challenging due to the time dependence. Therefore, the spacecraft charging calculations in the present implementation are sequential.

3 Code-To-Code Adapter: CoToCoA

3.1 Overview of the Framework

CoToCoA couples multiple physical models based on the Multiple-Program-Multiple-Data (MPMD) execution model. The concept of inter-code collaboration via CoToCoA is shown in Fig. 1. CoToCoA splits all available MPI processes into three subsets (groups) of processes, and assigns to each group one of the roles of Requester, Coupler, and Worker. Requester and Worker each correspond to different physical models to be coupled. These two roles are performed on different process groups that do not overlap with each other. Thus, CoToCoA assumes that each model is implemented as a standard MPI program and executed within its assigned process group as it was before being coupled. Coupler supervises the entire behavior of the framework, and acts as a mediator between Requester and Worker. The separation of the roles and the exclusive assignment of process groups to the respective roles minimize the effort for program modification.

Requester is responsible for physical computations that affect other physical processes. In the coupled physical systems, it offloads necessary computational tasks (i.e., requests) to Worker. Worker is generally responsible for physical computations that depend on the results of Requester computations. Coupler, to which a single MPI process is always assigned, monitors Requester and controls Worker as described later. CoToCoA has the ability to incorporate multiple Worker programs that are responsible for different computation tasks.

The general behavior of Requester and Workers within the CoToCoA framework is summarized in Fig. 2. Requester starts its own computation immediately after program startup. Workers are initially in an idle state, waiting for computation requests that are issued by Requester and forwarded by Coupler. Upon receiving computation requests, Worker starts its own computations and transitions its state to busy. When Worker has completed the requested computations, it sets its own state back to idle.

Coupler constantly monitors the status of Workers and computation requests issued by Requester. When it detects a computation request issued by Requester,

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_46

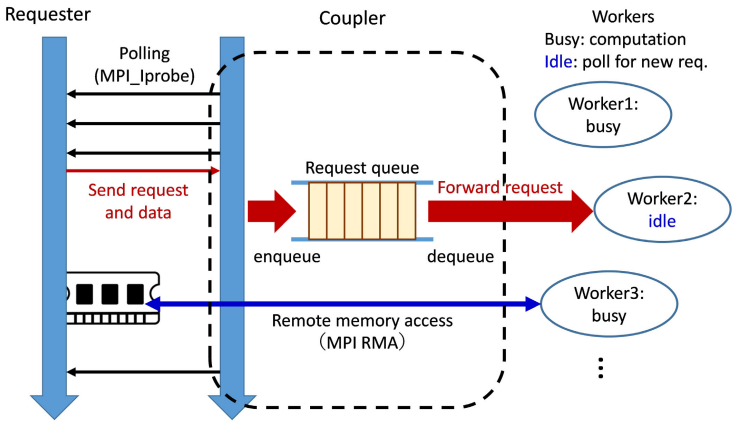

Fig. 2. Typical coordinative behavior of Requester, Coupler, and Workers within the CoToCoA framework.

it delegates the task to one of Workers that are in an idle state. If necessary, Coupler can also perform intermediate processing of the forwarded data on a user-defined basis. If all Workers are in a busy state, Coupler temporarily stores the computation request in a request queue, and waits for one of Workers in a busy state to transition to an idle state. When a worker that has completed its previously assigned work and enters an idle state is detected, it is delegated a new work. By repeating this procedure, CoToCoA sequentially processes the computation requests sent by Requester.

The program termination process is as follows. When Requester has completed all of its computations, it sets itself to an exit state. Coupler sets itself to an exit state when it confirms that Requester has entered an exit state and that its own request queue is empty of unprocessed computation requests. Finally, Workers terminate their own state after confirming that all assigned tasks have been completed and Coupler has entered an exit state. This completes the entire program execution. In the above description, Requester and Coupler do not necessarily know the status of Workers, but they can optionally use a function to check whether all issued requests have been completed on the Worker side or not.

3.2 Asynchronous Control of Coupled Programs

One of the main features of the framework is the handling of asynchronous computation requests, which tend to occur in coupled simulations. CoToCoA implements this through polling, which is hardware independent and can be described by a relatively simple loop statement. The physical models currently being coupled are non-interactive and do not require immediacy or real-time processing of asynchronous requests, so polling processing is sufficient to achieve efficiency.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_46

The polling process is mainly performed by Coupler and Worker. The CoToCoA framework provides API functions for this purpose. Coupler constantly polls whether there are computation requests issued by Requester, notifications of the end of Requester processing, and notifications of the completion of Worker processing. Worker constantly polls for computation requests forwarded by Coupler, and whether or not requests from Requester are being forwarded by Coupler. These functions are implemented by function calls such as MPI_Iprobe and MPI_Probe at constant intervals. The type of notification is distinguished by the tag information contained in the incoming message.

In response to the polling process described above, API functions are provided to send various notifications from Requester, Coupler, and Worker, respectively. The major functions are

1. The issuance of computation requests from Requester to Coupler,
2. The forwarding of computation requests from Coupler to Workers,
3. Notifications from Workers to Coupler that computation tasks have been completed,
4. Notification from Requester to Coupler that all requests have been issued,
5. Notification from Coupler to Worker that all requests have been forwarded.

3.3 Inter-code Exchange of Numerical Data

CoToCoA provides several data exchange methods depending on the situation, taking into account that each program executes its own computations asynchronously. The first method is to transfer the necessary physical data at the same time as a computation request is issued. This method can be used in a situation where the data to be sent have already been generated on Requester. In the API functions provided by CoToCoA, this is implemented by standard blocking-type communications such as MPI_Send and MPI_Recv.

In another situation, one program (either Requester or Worker) needs to refer to data generated by the other during its computations. In this case, since Requester and Worker are independent programs performing completely different computations asynchronously, an implementation using MPI functions for one-sided communications would be appropriate. It also allows the user to exploit the advantages of one-sided communications, such as the suppression of synchronous waiting and data copying, as well as compatibility with the RDMA capabilities. CoToCoA provides API functions for such Remote Memory Access (RMA), which is implemented using MPI.

4 Implementation of Coupled Analysis

Using the functionality of the CoToCoA framework described so far, we have implemented a coupled analysis program of magnetospheric dynamics and spacecraft charging. The MHD simulation plays the role of Requester and the SC calculation program plays the role of Worker. The skeleton of the developed

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_46

Fortran program is shown in Fig. 3. Here, we mainly explain how to use the API functions provided by CoToCoA, and omit the detailed implementation for the physical calculations within each model.

The functions CTCAX_init (X=R, C, or W) are called in Requester, Coupler, and Worker respectively to initiate CoToCoA. The API functions of CoToCoA are named in such a way that the prefixes of the function names easily identify whether the functions are for Requester, Coupler, or Worker. These prefixes are omitted in the following text. The initialization functions handle the construction of various data structures, the grouping and the assignment of processes to each Worker, and the definition of MPI sub-communicators. After the initialization, the CoToCoA function "Regarea_real4" constructs a window object for MPI one-sided communication, for which read/write permissions are set. In this coupled analysis, pressure and density data in spacecraft position coordinates are generated by the magnetospheric MHD simulation. The data are then accessed remotely by Worker.

After the completion of the initialization phase described above, the MHD simulation and the SC calculations are performed in parallel. In fact, the SC calculation on Worker is started based on the calculation request issued by Requester. Coupler polls for the arrival of computation requests from Requester by repeatedly calling the CoToCoA function "Pollreq" in its loop statement. Upon arrival of a computation request from Requester, the request is enqueued by the CoToCoA function "Enqreq". If it finds a Worker in an idle state when the next Pollreq function is called, it forwards the request to that Worker. The Pollreq functions described above are also used to check the completion of Worker and Requester computations.

Before starting the time update loop of the MHD simulation, Requestor calls the CoToCoA function "Sendreq" and issues a computation request to Coupler. At the same time, Requester also sends spacecraft orbit information to Coupler. Coupler then forwards these messages to one of the available Workers. After calling the Sendreq function, Requester immediately starts a time update of its own MHD computation. Although the density and pressure data at the spatial grid points are sequentially updated by the time update, they cannot be cleared until the Worker side has finished reading them. Since the up-to-date version (1.2.3) of the CoToCoA framework does not support this feature, users must implement some mechanism by themselves. There are two possible implementations to guarantee this point. The first is to suspend updating on the Requester side until Worker confirms the completion of reading data on the array. The second method is to copy the time sequence of the required physical data from the time update array and store them in a separate array. In this study, the latter method is adopted because the physical data required by Worker is as small as $4 \text{ byte} \times 2 = 8$ byte per time. This method is also consistent with the policy of aiming for an implementation that does not interfere with the progress of the Requester's computations as much as possible. In fact, after the time update loop starts, Requester can perform its computation without interference from Coupler and Worker.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_46

Requester (magnetospheric MHD simulation)

```
call CTCAR_init( )
call CTCAR_regarea_real4(data, size, areaid)
! MHD initialization (omitted herein)
call CTCAR_sendreq_withreal4(datint, ndatint, datreal, ndatreal)
do step = 1, nsteps
  ! MHD calculation: temporal integration (omitted herein)
  data(step, 1:ncomp) = &
  & field(scx,scy,scz,1:ncomp) ! write data to be offloaded
end do
! MHD finalization (omitted herein)
call CTCAR_finalize( )
```

Coupler

```
call CTCAC_init( )
call CTCAC_regarea_real4(areaid)
do while ( .true. )
  call CTCAC_pollreq_withreal4(reqinfo, fromrank, &
     & datint, ndatint, datreal, ndatreal)
  if( CTCAC_isfin( ) ) exit
  progid = SCCHARGE
  call CTCAC_enqreq_withreal4(reqinfo, progid, &
     & datint, ndatint, datreal, ndatreal)
end do
call CTCAC_finalize( )
```

Worker (spacecraft charging calculations)

```
call CTCAW_init(progid, procs_per_req)
call CTCAW_regarea_real4(areaid)
do while ( .true. )
  call CTCAW_pollreq_withreal4(fromrank, &
     & datint, ndatint, datreal, ndatreal)
  if( CTCAW_isfin( ) ) exit
  ! SC initialization (omitted herein)
  step = 1
  do while (step <= nsteps)
    call CTCAW_readarea_real4(areaid, rank, offset, size, data)
    if( .not.(find_new_data(data(step, 1))) ) cycle
    ! SC calculation: temporal integration (omitted herein)
    step = step + 1
  end do
  call CTCAW_complete( )
end do
call CTCAW_finalize( )
```

Fig. 3. Skeleton code of the MHD–SC coupled analysis.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_46

Worker performs the SC computation inside a double-loop structure. In the outer loop, it polls for a computation request from Coupler by repeatedly calling the Pollreq function. After detecting the arrival of a computation request, the process enters the inner loop and starts the SC calculation. Since the SC calculation refers to the pressure and density data provided by the MHD at each time, it must always be preceded by the MHD simulation. To accomplish this, Worker periodically accesses and checks the physical data storage array on remote memory managed by Requester. This is implemented by using the CoToCoA function "Readarea", which serves as a one-sided communication function. Worker controls its own SC calculations based on the numerical data stored on this remote memory array as follows. At the initial stage of the computation, the memory array is set to a specific value (such as "0.0") to indicate that the computation is incomplete. As the MHD calculation progresses, the "null" values of the array elements reserved for the corresponding time steps are replaced with the obtained plasma pressure and density values. Worker can remotely reference this data to know the progress of the MHD computation. By repeatedly calling Readarea, Worker can always decide whether to proceed with its own SC calculations or wait for Requester to generate plasma macro parameters. This feature provides a mechanism for Worker to control the SC calculations. Finally, when the preset time has been calculated, the Worker calls the CoToCoA function "Complete" to notify Coupler that its task as a Worker is finished.

5 Case Study and Performance Evaluation

This section presents a case study of the coupled analysis performed to verify its validity. We analyzed the charging phenomena of a hypothetical artificial satellite exposed to the Earth's magnetospheric environment on April 5, 2010, when a geomagnetic substorm triggered by a solar flare-driven coronal mass ejection took place [15]. It has been reported that the U.S. satellite Galaxy 15 actually experienced a failure during this substorm. As mentioned in the previous section, the data provided by the MHD computation are the plasma density and pressure. The plasma temperature is derived from these physical parameters and used as an input to the charging calculation. Although the SC calculations require separate electron and ion temperatures, in principle, it is not possible to distinguish between ions and electrons in the MHD simulation. Therefore, the electron and ion temperatures are assumed to be identical.

The experiments were performed on up to 64 nodes of a Fujitsu PRIMERGY CX2550 installed at the Research and Development Center for Information Infrastructure, Kyushu University. Each node of the ITO System A is equipped with two Intel Xeon Gold (Skylake SP) processors and 192 GB of DDR4 memory. To validate the reproduced physical phenomena, a coupled analysis was performed with a simplified configuration considering only currents of the plasma electrons and ions. The computational resources used were 32 nodes of ITO System A. The number of MPI processes assigned to the requester, coupler, and worker were 512, 1, and 1, respectively. The magnetospheric MHD simulation

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_46

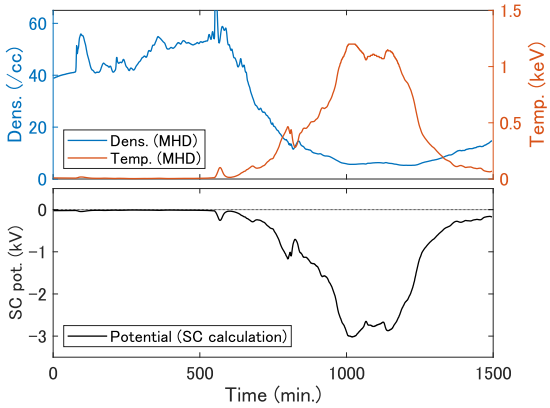

Fig. 4. Numerical results without photo and secondary electron effects. The upper panel shows the plasma density and temperature derived from the MHD simulation. The lower panel shows the spacecraft potential.

uses a 3-dimensional space consisting of $600 \times 400 \times 400 \sim 10^8$ grid points. Fig. 4 shows the result of the numerical experiment. The upper panel shows the plasma density and electron temperature evaluated within the MHD simulation, and the lower panel shows the result of the spacecraft potential calculation. A comparison of the time evolution of each quantity shows that the spacecraft potential has an inverse correlation with the electron temperature. In general, when an object is placed in a two-component plasma consisting of electrons and ions, an electron current with a high thermal velocity dominates. As a result, the spacecraft becomes negatively charged at approximately the electron temperature to repel the surrounding electrons. This leads to a current equilibrium between the electrons and the ions. The time evolution of the spacecraft potential obtained in this coupled analysis also reflects this fundamental physical behavior.

In reality, more types of current components are involved in the SC processes, such as photoemission and secondary electron emission, making the processes more complex. The analysis for such a practical situation is shown in Fig. 5, where the individual current values and a secondary electron emission coefficient are displayed. The current magnitudes and the secondary emission coefficient depend on the temperature of the surrounding plasma as well as on the spacecraft potential itself. It follows that each current is closely correlated to the plasma conditions provided by the MHD simulation. A notable feature is the abrupt drop in the spacecraft potential seen at the time period 1024–1060 min.. This period corresponds to the time when the orbiting satellite enters an eclipse by the Earth and the photoelectron emission ceases. Since the photoelectron emission flux (as the current that positively charges the spacecraft) is generally greater than the incoming electron flux, the spacecraft potential fluctuates around a few V positive outside the eclipse. In contrast, the incoming electron flux in turn becomes the dominant component during the eclipse, and the space-

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_46

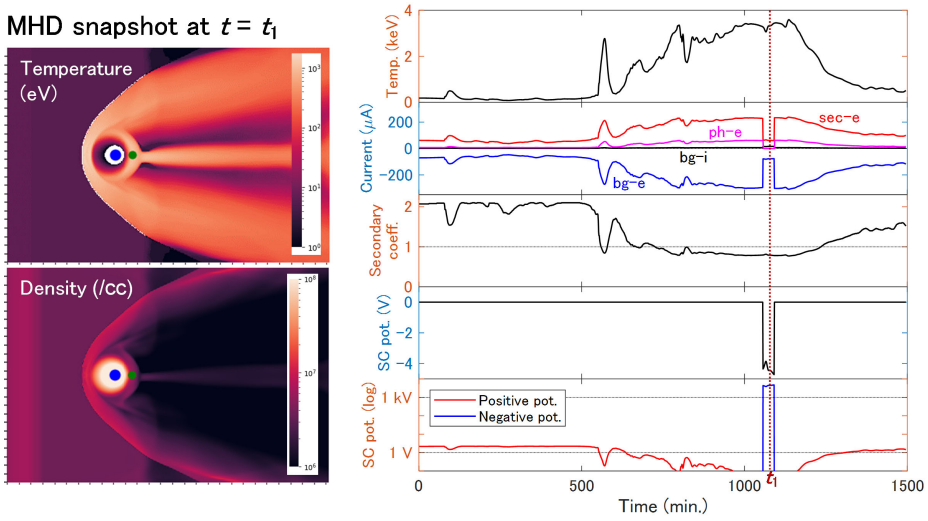

Fig. 5. Numerical results with photo and secondary electron effects. The left panel shows the snapshot of the MHD simulation at $t = t_1$. The right panel shows the time-series result of the SC calculations. The notations: bg-e, bg-i, sec-e, and ph-e in the second plot in the right panel represent the background plasma electron, ion, secondary electron, and photoelectron currents, respectively. The bottom panel shows the spacecraft potential values on a logarithmic scale, with the red line duration being positive and the blue line duration being negative.

craft potential drops down to a deep negative potential. Such an abrupt change in potential can be detrimental to maintaining the integrity of actual satellite operations. The actual drop in spacecraft potential that occurs during an eclipse varies greatly depending on the temperature of the space plasma as well as the secondary electron emission coefficient. The ability to evaluate the potential based on environmental parameters derived from the physics model-based MHD simulations is a major step forward in building a future SC prediction platform.

Next, a performance evaluation was performed to characterize the computational loads for MHD and SC. We measured the elapsed time required for Requester (magnetospheric MHD simulations), Worker (SC simulations), and the entire coupled analysis. In this evaluation, we changed the degree of parallelism (the number of processes) only for the MHD computations. The number of allocated processes was changed from 32 to 2048 while keeping the problem size constant (i.e., strong scaling mesurement).

Fig. 6 shows the results of the performance evaluation. The horizontal axis represents the number of processes assigned to the MHD computation. The total execution time required for the coupled analysis decreases as the number of processes increases in the range from 32 to 512 processes. The execution times for Requester and Worker are nearly identical in the range of 32 to 256 processes. In the coupled analysis, Worker (SC calculation) continues to attempt

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_46

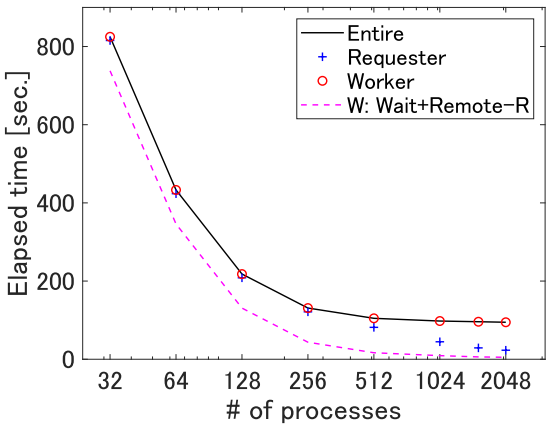

Fig. 6. Elapsed time of the coupled analysis of magnetospheric dynamics and SC phenomena via the developed framework. The pink dashed line represents the time required to "wait" and "read" by RMA in the worker program.

remote memory access to Requester without updating the spacecraft potential until the required plasma physics data become available on the MHD side. The processing time of Worker includes this waiting time, and thus the processing speed of Worker is constrained by that of Requester. In fact, in the 32-process execution, the time spent waiting for the Requester accounted for 89.4% of the total execution time. This indicates that by allocating more MPI processes to Requester to speed up it, the waiting time on the worker side can also be reduced accordingly. The time required for Requester and Worker analysis is identical for 32 to 256 process executions. In this regime, the numerical processing time for the SC analysis is less than and hidden in that for the MHD computation.

For 512 to 2048 process executions, there is a discrepancy between the execution times required by Requester and Worker. For the magnetospheric MHD computation, the execution time is reduced up to about 1024 processes, but saturates above this number. This is because the ratio of boundary communication costs increases, whereas the amount of computation within a small region decreases as the number of decompositions increases. Meanwhile, the actual computation processing time of Worker is basically invariant, and the only factor that can be reduced is the overhead part due to waiting and remote memory access. Thus, once the overhead is minimized (3.5% of the total time), the execution time for Worker does not decrease any further.

Through these numerical experiments, it was found that with the experimental setup used in this study, the MHD computation time of 200,000 to 400,000 grid points and the processing time for one time-step update of the spacecraft potential are in balance with each other. It was also confirmed that the coupled implementation has almost no effect on the processing time of the MHD computation. In fact, there is only a difference of less than 10^{-1} seconds for

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_46

any number of process executions between the MHD processing time during the coupled analysis and the execution time when MHD is run alone. Such overhead is negligible enough when compared to the total processing time of the MHD computation (e.g., ∼80 seconds for the 512 process execution).

6 Conclusions

A coupled numerical analysis of magnetospheric MHD simulations and spacecraft charging calculations is implemented using an in-house code-to-code coupling framework, CoToCoA. The original simulation codes have been developed independently so far and perform computations with completely different content. Therefore, an asynchronous inter-code coordination is required to realize the coupled analysis. In addition, for high productivity of the coupled analysis, it is beneficial to define a requester-worker relationship between the programs to be coupled. Which model should play the role of Requester or Worker depends on the context of what information and when physical information should be exchanged between models. Based on this idea, we designed the CoToCoA framework, which provides a control program (Coupler) that supervises the inter-code coordination as well as the asynchronous code control, and provides API functions for inter-code data exchange based on remote memory access.

In the case of the coupled analysis of magnetospheric MHD and spacecraft charging simulations, the former is positioned as Requester and the latter as Worker. Worker (SC analysis) uses remote memory access to check the status and progress of Requester (MHD) processing as needed, and to determine whether or not the update of a spacecraft potential can be performed.

The coupled analysis program has been verified on the supercomputer system at the Kyushu University. The developed model successfully reproduces the transient behavior of the spacecraft potential, which is based on the variational plasma macro-parameters generated by the magnetospheric MHD simulation. The execution time required for the coupled analysis was also measured while varying the number of MPI processes assigned to the parallelized magnetospheric MHD calculation. It was confirmed that there is a point between 256- and 512-process executions where the computational cost of the magnetospheric MHD and the SC calculations are balanced.

Since the SC calculation is not yet parallelized in the current implementation, the number of processes used for the MHD computation must be kept below a certain level in order to hide the SC analysis within the processing time of MHD. To make the analysis more flexible by changing the problem size and the amount of computational resources used, it is necessary to introduce time-domain parallelization for the SC analysis, which is left as a future work.

Acknowledgements This work was supported in part by the Joint Usage and Research Center for Interdisciplinary Large-Scale Information Infrastructure and Innovative High Performance Computing Infrastructure (project numbers: jh210047-NAH, jh220017, jh230042, hp220040, and hp230046), as well as the

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_46

Japan Society for the Promotion of Science (JSPS) KAKENHI Grant Number JP22K12049. The numerical experiments were carried out using the ITO System at Kyushu University and the Camphor system at Kyoto University.

References

1. Schwartz, S., et al.: Cross-scale: multi-scale coupling in space plasma. Assessment Study Report (2009).
2. Fukazawa, K., Katoh, Y., Nanri, T., and Miyake, Y.: Application of cross-reference framework CoToCoA to macro- and micro-scale simulations of planetary magnetospheres. In: Proc. 2019 Seventh International Symposium on Computing and Networking Workshops (CANDARW), pp. 121–124. (2019)
3. Katoh, Y., Fukazawa, K., Nanri, T., and Miyake, Y.: Cross-reference simulation by code-to-code adapter (CoToCoA) library for the study of multi-scale physics in planetary magnetospheres. In: Proc. 8th International Workshop on Large-scale HPC Application Modernization (LHAM), pp. 223–226. (2021)
4. Matsumoto, H., and Omura, Y.: Computer space plasma physics : simulation techniques and software. Terra Scientific Pub., Tokyo (1993)
5. Toth, G., et al.: Space weather modeling framework: a new tool for the space science community. J. Geophys.Res. **110**, A12226 (2005) https://doi.org/10.1029/2005JA011126
6. Sugiyama, T., Kusano, K., Hirose, S., and Kageyama, A.: MHD-PIC connection model in a magnetosphere-ionosphere coupling system. J. Plasma Phys. **72**(6), 945–948 (2006)
7. Nanri, T., Katoh, Y., Fukazawa, K., Miyake, Y., Nakazawa, K., Zhow, J., and Sunada, Y.: CoToCoA (Code-To-Code Adapter) version 1.2.2, https://doi.org/10.5281/zenodo.5775280
8. Ogino, T., Walker, R. J., and Ashour-Abdalla, M.: A global magnetohydrodynamic simulation of the magnetopause when the interplanetary magnetic field is northward. IEEE Trans. Plasma Sci. **20**, 817–828 (1992)
9. Fukazawa, K., Ogino, T., and Walker, R. J.: The configuration and dynamics of the Jovian magnetosphere. J. Geophys. Res. **111**, A10207 (2006)
10. Fukazawa, K., Ueda, M., Inadomi, Y., Aoyagi, M., Umeda, T., and Inoue, K.: Performance analysis of CPU and DRAM power constrained systems with magnetohydrodynamic simulation code. In: Proc. 2018 IEEE 20th Intl. Conf. High Performance Computing and Communications, pp. 626–631, (2018) https://doi.org/10.1109/HPCC/SmartCity/DSS.2018.00113
11. Whipple, E. C.: Potentials of surfaces in space. Reports on Progress in Physics **44**(11), 1197–1250 (1981)
12. Miyake, Y., and Usui, H.: New electromagnetic particle simulation code for the analysis of spacecraft-plasma interactions. Phys. Plasmas. **33**(3), 258–266 (2019)
13. Massaro, M. J., Green, T., and Ling, D.: A charging model for three-axis stabilized spacecraft. In: Proc. Spacecraft Charging Techology Conf., pp. 237–269 (1977)
14. Simos, T. E.: A Runge-Kutta Fehlberg method with phase-lag of order infinity for initial-value problems with oscillating solution. Computers & Mathematics with Applications **25**(6), 95–101 (1993)
15. Ferguson, D. C., Denig, W. F., Rodriguez, J. V.: Plasma conditions during the Galaxy 15 anomaly and the possibility of ESD from subsurface charging. In: 49th AIAA Aerospace Science Meeting including the New Horizons Forum and Aerospace Exposition, pp. 2011–1061. AIAA (2011)

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_46