

Coupled Kinetic-Fluid Simulations of Ganymede's Magnetosphere and Hybrid Parallelization of the Magnetohydrodynamics Model

by
Hongyang Zhou

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Climate and Space Sciences and Engineering and Scientific Computing)
in the University of Michigan
2020

Doctoral Committee:

Associate Professor Xianzhe Jia, Co-Chair
Professor Gábor Tóth, Co-Chair
Professor Tamas Gombosi
Professor Ken Powell
Associate Research Scientist Bart van der Holst

Hongyang Zhou

hyzhou@umich.edu

ORCID iD: 0000-0003-4571-4501

© Hongyang Zhou 2020

This Ph.D. thesis is dedicated to:
my parents, who always stand behind me with unconditional support

Acknowledgements

This thesis is a collaborative work with the help and guidance from many people. First and foremost, I want to thank my advisors Gábor Tóth and Xianzhe Jia for their professional and thoughtful guidance throughout my thesis research. After all the suffering about academia, I have gradually learned how to conduct solid research. In a non-perfect world, we can still perform thorough studies to appealing things and discover interesting connections.

Additional thanks go to the members serving on my Ph. D. committee: Drs. Tamas Gombosi, Bart van der Holst and Ken Powell. Thank you for taking the time out of your busy schedules to read through this dissertation. I also thank you for your roles as educators by facilitating my growth in knowledge and training required for a science career.

Thanks to my teachers Krzysztof Fidkowski, Yueying Lau, Robert Krasny, Shasha Zou, Mark Flanner and many others. I will never forget the excellent lectures in all these years. “If you can’t explain it simply, you don’t know it well enough.”

Thanks to my friends, fellow students and colleagues at University of Michigan, who are always interested in discussing new ideas and techniques with me. I especially appreciate my two roommates, Chu Chen and Changjiang Liu, for the intensive communications and persistent humours that run through our Ph.D. days. As long as we keep thinking and pursuing our goals, there is always passion and fulfillment in life. Good luck to you guys in all of your endeavors!

Many thanks go to the staff of the Climate and Space Sciences and Engineering Department: Darren Britten-Bozzone, Laura Hopkins, Claire Miller, Rachel Long, Cameron Shawcross, Faye Ogasawara, and Sandra Pytlinski. I am proud to say that we have the best IT support across the university and an efficient paperwork procedure.

Travel and funding support from the Rackham Graduate School, National Science Foundation, NASA, and ISSS Modeling Workshop is gratefully acknowledged.

Table of Contents

Dedication	ii
Acknowledgements	iii
List of Figures	vii
List of Tables	ix
Abstract	x
CHAPTER	
I. Introduction	1
1.1 Magnetohydrodynamics Magnetosphere Simulation	3
1.2 Particle-in-Cell Simulations of Space Plasma	5
1.3 Exploration of Ganymede	8
1.3.1 Orbit and rotation	8
1.3.2 Internal structure	9
1.3.3 Magnetosphere	10
1.3.4 Magnetosphere Modeling	15
1.4 Modern Supercomputer Architecture	16
1.4.1 Parallelism	17
1.4.2 Platform	19
II. Model Description	20
2.1 Space Weather Modeling Framework	20
2.2 BATSUS	21
2.2.1 Hall MHD + Electron Pressure Equation	22
2.3 Particle-in-Cell Model	25
2.3.1 Implicit Particle-in-Cell Method	27
2.3.2 Energy Conserving Semi-Implicit Method	27
2.4 MHD-EPIC	29
2.5 Ganymede's Magnetosphere Simulation	30
2.5.1 Initial Conditions	30
2.5.2 Boundary Conditions	31
2.5.3 Coupling Procedure	35
III. Validation and Inferences	36
3.1 Magnetic Field Comparison for the 6 Flybys	38
3.2 Plasma Velocity Measurements	41
3.3 Upstream Dynamics	45

3.4	Ion Velocity Distribution	49
3.5	Energy Flux Density and Auroral Emission	51
3.6	Unresolved Electron Kinetics	56
IV. Reconnection-driven Dynamics		59
4.1	Time-accurate Magnetic Field Comparison	61
4.1.1	Plasma Pressure and Total Current Density	62
4.2	Magnetopause Dynamics and FTEs	63
4.3	Kinetic Signatures near the Diffusion Region	70
4.3.1	Violations of Frozen-in Condition	72
4.3.2	Non-gyrotropy Measures	73
4.3.3	Dissipation Measure	74
4.3.4	Reconnection Site with a Flux Rope	76
4.3.5	Phase Space Distributions	76
4.4	Reconnection Rate	80
4.5	Wave Detection	85
4.5.1	Ultralow Frequency Wave	85
4.5.2	Lower Hybrid Drift Instability	89
4.5.3	Discussion	91
V. Hybrid Parallelization		95
5.1	OpenMP in BATSRUS	95
5.1.1	Hybrid Parallelization Strategy	96
5.1.2	Implementation	100
5.1.3	Testing and Debugging	107
5.1.4	Performance	108
5.2	GPU Offloading	120
VI. Visualization and Data Analysis Package: VisAna		124
6.1	Brief History of Dynamic Programming Languages	124
6.2	VisAna: A Community Package for Simulation Data Processing	126
6.2.1	Field Tracing	127
6.2.2	Particle Tracing	130
6.2.3	File Format Conversion	131
6.2.4	Parallel Execution	133
6.2.5	Programming Language Interoperability	133
6.2.6	Test Tracking	133
VII. Conclusions		135
7.1	Summary of Results	135
7.1.1	Embedded Kinetic Simulation of Ganymede's Magnetosphere	135
7.1.2	Reconnection-driven Dynamics at Ganymede's Upstream Magnetosphere	136
7.1.3	Hybrid Parallel MHD Model Development	137
7.1.4	Data Visualization and Analysis	138
7.2	Future Work	138
7.2.1	Ganymede's Magnetosphere Modeling	138
7.2.2	Hybrid Parallelization	140
Bibliography		142

List of Figures

1.1	PIC scheme flow chart	6
1.2	Ganymede’s internal structure	9
1.3	Demonstration of Alfvén wing structure	11
1.4	Ganymede’s magnetic field	13
1.5	Supercomputer performance	18
2.1	BATSRUS high level structure	23
2.2	PIC cell structure for field storage	26
2.3	Temporal discretization of MHD-EPIC coupling	29
2.4	Sketch of simulation boundaries in the Ganymede’s magnetosphere model	31
2.5	Resistivity profile in the mantle region	34
3.1	Ganymede simulation grid structure	37
3.2	G8 flyby Alfvén wing and open-close field line boundary	38
3.3	G28 flyby Alfvén wing and open-close field line boundary	39
3.4	Quasi-steady state magnetic field comparison	40
3.5	G8 flyby time-accurate magnetic field comparison	42
3.6	G28 flyby time-accurate magnetic field comparison	43
3.7	G2 flyby plasma velocity comparison	44
3.8	Contour plots of upstream reconnection region	48
3.9	Ion velocity distributions near the upstream reconnection site	49
3.10	Energetic flux mapping from PIC region onto Ganymede’s surface	53
4.1	Magnetic field comparisons with Galileo G8 flyby Hall MHD and MHD-EPIC simulation	62
4.2	Ion and electron pressures in the meridional cuts	64
4.3	Total current density in the meridional and equatorial cuts	65
4.4	Magnetopause in Hall MHD and MHD-EPIC	66
4.5	Flux rope in MHD-EPIC simulation	68
4.6	Thermal pressure perturbation on the magnetopause	69
4.7	Plasma and field measures of electron diffusion region	71
4.8	Electron diffusion region with a flux rope	75
4.9	Electron and ion phase space distributions	77
4.10	Illustration of reconnection efficiency calculation	84
4.11	Global upstream reconnection rates from Hall MHD and MHD-EPIC	85
4.12	Alfvén wave detection from simulations	87
4.13	Magnetosonic waves detection from simulations	88
4.14	ULF wave distribution in the Galileo G8 flyby simulation	90
4.15	Lower hybrid drift instability detection	91
5.1	BATSRUS parallelization structure	99
5.2	BATSRUS hybrid strong scaling	110
5.3	BATSRUS explicit scheme hybrid weak scaling	112
5.4	BATSRUS implicit scheme hybrid weak scaling	113
5.5	BATSRUS explicit timing by module	115
5.6	BATSRUS implicit timing by module	116
5.7	BATSRUS hybrid weak scaling on Stampede2	117

5.8	BATSRUS hybrid timing comparison	119
6.1	Benchmark for VisAna	127
6.2	Demonstration of asymptotic field tracing	129
6.3	Demonstration of dipole field tracing	130
6.4	Unstructured 2D grid tracing	131
6.5	Particle tracing in Earth's dipole field	132
6.6	Tracking of SWMF nightly test timings.	134

List of Tables

1.1	Global magnetosphere models.	5
1.2	Platform Information	19
2.1	Simulation Parameters for Galileo's Six Close Encounters	30
2.2	Boundary conditions	33
2.3	Grid size and particle number of MHD-EPIC models.	35
5.1	Platform Information	109
5.2	Timings for the skeleton BATSRUS for porting to GPU with OpenACC	123

Abstract

The largest moon in the solar system, Ganymede, is the only moon known to possess a strong intrinsic magnetic field. The interaction between the Jovian plasma and Ganymede's magnetic field creates a mini-magnetosphere with periodically varying upstream conditions, which creates a perfect laboratory in nature for studying magnetic reconnection and magnetospheric physics. Using the latest version of Space Weather Modeling Framework (SWMF), we study the upstream plasma interactions and dynamics in this subsonic, sub-Alfvénic system. We have developed a coupled fluid-kinetic Hall Magnetohydrodynamics with embedded Particle-in-Cell (MHD-EPIC) model for Ganymede's magnetosphere, with a self-consistently coupled resistive body representing the electrical properties of the moon's interior, improved inner boundary conditions, and high resolution charge and energy conserved PIC scheme.

I reimplemented the boundary condition setup in SWMF for more versatile control and functionalities, and developed a new user module for Ganymede's simulation. Results from the models are validated with Galileo magnetometer data of all close encounters and compared with Plasma Subsystem (PLS) data. The energy fluxes associated with the upstream reconnection in the model is estimated to be $\sim 10^{-7} \text{ W/cm}^2$, which accounts for about 40% to the total peak auroral emissions observed by the Hubble Space Telescope. We find that under steady upstream conditions, magnetopause reconnection in our fluid-kinetic simulations occurs in a

non-steady manner. Flux ropes with length of Ganymede's radius form on the magnetopause at a rate about 3/minute and create spatiotemporal variations in plasma and field properties. Upon reaching proper grid resolutions, the MHD-EPIC model can resolve both electron and ion kinetics at the magnetopause and show localized crescent shape distribution in both ion and electron phase space, non-gyrotropic and non-isotropic behavior inside the diffusion regions. The estimated global reconnection rate from the models is about 80 kV with 60% efficiency. There is weak evidence of ~ 1 minute periodicity in the temporal variations of the reconnection rate due to the dynamic reconnection process.

The requirement of high fidelity results promotes the development of hybrid parallelized numerical model strategy and faster data processing techniques. The state-of-the-art finite volume/difference MHD code Block Adaptive Tree Solarwind Roe Upwind Scheme (BATS-R-US) was originally designed with pure MPI parallelization. The maximum problem size achievable was limited by the storage requirements of the block tree structure. To mitigate this limitation, we have added multithreaded OpenMP parallelization to the previous pure MPI implementation. We opt to use a coarse-grained approach by making the loops over grid blocks multithreaded and have succeeded in making BATS-R-US an efficient hybrid parallel code with modest changes in the source code while preserving the performance. Good weak scalings up to 50,000 and 25,000 of cores are achieved for the explicit and implicit time stepping schemes, respectively. This parallelization strategy greatly extends the possible simulation scale by an order of magnitude, and paves the way for future GPU-portable code development. To improve visualization and data processing, I have developed a whole new data processing workflow with the Julia programming language for efficient data analysis and visualization.

As a summary,

1. I build a single fluid Hall MHD-EPIC model of Ganymede's magnetosphere;
2. I did detailed analysis of the upstream reconnection;
3. I developed a MPI+OpenMP parallel MHD model with BATSRUS;
4. I wrote a package for data analysis and visualization.

CHAPTER I

Introduction

The Space Age is generally considered to have begun with the first artificial satellite Sputnik 1 in 1957, continuing to the present day. The exploration of space physics with the aid of modern knowledge of science and technology has been moving along a fast track for more than half a century. People have found that plasma, i.e. ionized gases, which is uncommon on the surface of Earth, fills up 99% of the universe and is involved in most of the interplanetary physical processes. While in-situ satellite observation is limited to usually one small spatial location at a given time, and remote sensing is constrained in its capability to detect plasma properties, numerical physical models based on solving governing equations given the initial and boundary conditions serve as an important tool in understanding the physics happening in space. Especially for the past decade, 3D global simulations of magnetospheres, i.e. a cavities created by plasma interaction with planetary magnetic fields, has come to reality thanks to the rapid development of numerical physics models, computer architecture, and large scale parallelization techniques. Researchers have managed to simulate literally every magnetosphere within the solar system, from the terrestrial planets Mercury, Venus, Earth and Mars, gas giants Jupiter and Saturn, to the ice giants Uranus and Neptune. Beyond these well known planets, our solar

system contains many moons of interest, especially in the Jovian system, which is enclosed by the large Jovian magnetosphere. Jupiter's four largest moons, Io, Europa, Ganymede, and Callisto, are called the Galilean satellites after Italian astronomer Galileo Galilei, who first observed them in 1610. Ganymede, which is the target of the present research, is the largest moon in the solar system (with a radius larger than Mercury) and the only moon that possesses an intrinsic magnetic field. The existence of its mini-magnetosphere inside Jupiter's large magnetosphere raises considerable interest among space physicists. One of the most interesting phenomena in magnetospheric physics is the magnetic reconnection. Magnetic reconnection is a physical process occurring in highly conducting plasmas in which the magnetic topology is rearranged and magnetic energy is converted to kinetic energy, thermal energy, and particle acceleration. Because of the steady periodically varying Jovian plasma environment around Ganymede's orbital revolution, it is regarded as a perfect laboratory in nature for studying magnetic reconnection and its related effects. This thesis focuses on the application of the state-of-the-art coupled kinetic-fluid model Magnetohydrodynamics (MHD) with embedded Particle-in-Cell Model (MHD-EPIC) to Ganymede's magnetosphere. Hybrid Message Pass Interface (MPI)+ Open Multi-Processing (OpenMP) model has been developed for large scale simulations, as well as a whole new workflow of data processing and analysis incorporating the latest techniques.

Chapter I introduces some fundamental concepts in space physics simulations. Chapter II describes our Ganymede MHD-EPIC model in detail. Chapter III shows the validation and comparison of the model against Galileo observations, which is published in [Zhou et al., 2019]. Chapter IV shows the modelling results focusing on upstream reconnection that is submitted to JGR Space Physics [Zhou et al.,

[accepted, 2020](#)]. Chapter V presents the new hybrid parallelization model for large scale simulations. The OpenMP section 5.1 is published in [[Zhou and Tóth, 2020](#)]. Chapter VI introduces the latest toolkit for data processing and analysis [[Zhou, 2020](#)], and Chapter VII summarizes the thesis, points out the limitations and ideas for future work.

1.1 Magnetohydrodynamics Magnetosphere Simulation

Magnetospheric numerical simulations can be divided into two classes: local simulations and global simulations based on their spatial coverage. Often we use a particle description to model localized kinetics and a fluid description to model the global convection pattern.

The fluid description, based on the foundation of mass, momentum and energy conservation laws and Maxwells equations of electromagnetism, is called **magneto-hydrodynamics (MHD)**. The global magnetosphere MHD numerical simulation approach is a self-consistent mathematical description of the interaction of a stream of charged particles with the magnetic field. The importance of the development of global 3D MHD models lies in two aspects. On one hand, they help deepen the physical insight of the observations and gain more information beyond the observed regions. Despite the number of the satellites launched and ground-based instruments employed, the observed regions are limited with respect to the vast space in the universe. It is thus vital to use models to grasp the global picture based on the limited observational data. On the other hand, they help test and improve theoretical frameworks, give quantitative description of space physics phenomena, and provide the scientific base to develop practical physics-based space weather forecast models. Global models have a true sense of the testability, can be compared with

the observations, and provide guidance to local models [Wang et al., 2013].

Development of the global MHD magnetosphere models started in the 1970s, and initially limited to two-dimensional (2D) cases. Due to the increased computing power, 3D MHD simulations emerged in the 1980s, in an attempt to model the large-scale structures and fundamental physical processes in the real geospace. Modeling work started to be combined with the space exploration missions in the 1990s and compared with observations, and gradually became an integrated component of space missions. Physics-based space weather forecast models started to be developed in the new century. While large-scale parallelization on supercomputers and advances in CPU clock speed performance make it possible to refine the numerical grid until the physics of interest is well resolved, it also allows including more physical terms in the governing equations. In both ways, one is closer to grasp the global picture of space beyond the observed regions and understand the physics behind the phenomena. Global MHD magnetospheric models provide the scientific base for practical space weather forecast models as well, making it possible to predict natural hazards like geomagnetic storms and prevent damages accordingly.

MHD models have many different levels of approximation under the quasi-neutrality assumption. Let n be the plasma number density, \mathbf{u} the plasma velocity, \mathbf{J} the current density, \overleftrightarrow{P}_e the electron pressure tensor, and \mathbf{B}, \mathbf{E} the magnetic and electric field. In the generalized Ohm's law [Gurnett and Bhattacharjee, 2005]

$$\mathbf{E} = \underbrace{-\mathbf{u} \times \mathbf{B}}_1 + \underbrace{\eta \mathbf{J}}_2 + \underbrace{\frac{1}{en} \mathbf{J} \times \mathbf{B}}_3 - \underbrace{\frac{1}{en} \nabla \cdot \overleftrightarrow{P}_e}_4 + \underbrace{\frac{m_e}{ne^2} \left[\frac{\partial \mathbf{J}}{\partial t} + \nabla \cdot (\mathbf{J}\mathbf{u} + \mathbf{u}\mathbf{J}) \right]}_5, \quad (1.1)$$

the first term on the right-hand side is called the “convective term”, the second is the “resistive term”, the third is the “Hall term”, the fourth is the “electron pressure term”, and the fifth is the “electron inertia term” since it is proportional to the

mass of electrons. Depending on how the electric field is expressed in Equation 1.1, MHD can be further categorized into ideal MHD (1), resistive MHD (1+2), Hall MHD (1+2+3) and beyond. Using nondimensionlization for the usual space plasma parameters, we find that the above terms are sequentially decreasing in magnitude. However, the terms 2-4 become important when a large current density exists, e.g. at the magnetopause and reconnection sites where the magnetic fields drastically change.

A list of current numerical global magnetosphere models is given in Table 1.1. These models differ in the choices of grid structure, order and type of numerical schemes, and language of implementation. The BATS-R-US code, especially the Hall MHD module with independent electron pressure equation, will be further introduced in Chapter II.

Code	Model	Grid	Scheme	Language
BATS-R-US 2012	Multi-physics	Block-AMR	FV/FD*	Fortran
LFM 2019a	Ideal MHD	Non-orthogonal curvilinear grid	FV	Fortran
Gkeyll 2019	10 moment	Stretched-Cartesian	DG**	LuaJIT, C++
GUMICS 2012	Ideal MHD	Cell-AMR, M-I coupled	FV	C++
MPI-AMRVAC 2020	Multi-physics	Block-AMR	FV	Fortran
PPMLR-MHD 2005	Ideal MHD	Stretched-Cartesian	FD	Fortran
OpenGGCM 2008	Resistive MHD	Stretched-Cartesian	FD	Fortran

* FV stands for finite volume method and FD stands for finite difference method.

** DG stands for Discontinuous Galerkin method.

Table 1.1: Global magnetosphere models.

1.2 Particle-in-Cell Simulations of Space Plasma

To simulate kinetic processes of plasma, the **particle-in-cell (PIC)** method is has been developed. It allows the statistical representation of general distribution functions in phase space [[Verboncoeur, 2005](#)]. It usually consists of two main parts: the particle pusher, which tracks individual macro-particles in continuous phase space, and the field solver, which updates the electromagnetic (EM) field on a grid given

the moments of particle distribution. At the heart of the PIC scheme, a particle-mesh method was devised to describe the interaction between particles and fields, as shown in Figure 1.1. This converts the $O(N^2)$ complexity of calculating Coulomb's law for each particle into a $O(N)$ complexity system, which is a great saving in computational cost. This simplification is quite accurate in "collisionless" plasma, where the direct Coulomb interaction between particles is negligible. Most of space plasma can be considered collisionless. The use of fundamental equations without much approximation retains the full nonlinear effects and also allows the incorporation of relativistic effects. In general, particle models retain most of the physics and are often used to benchmark more approximate methods such as MHD. New models and methods continue to expand the domain of particle methods, as well as improving fidelity and performance.

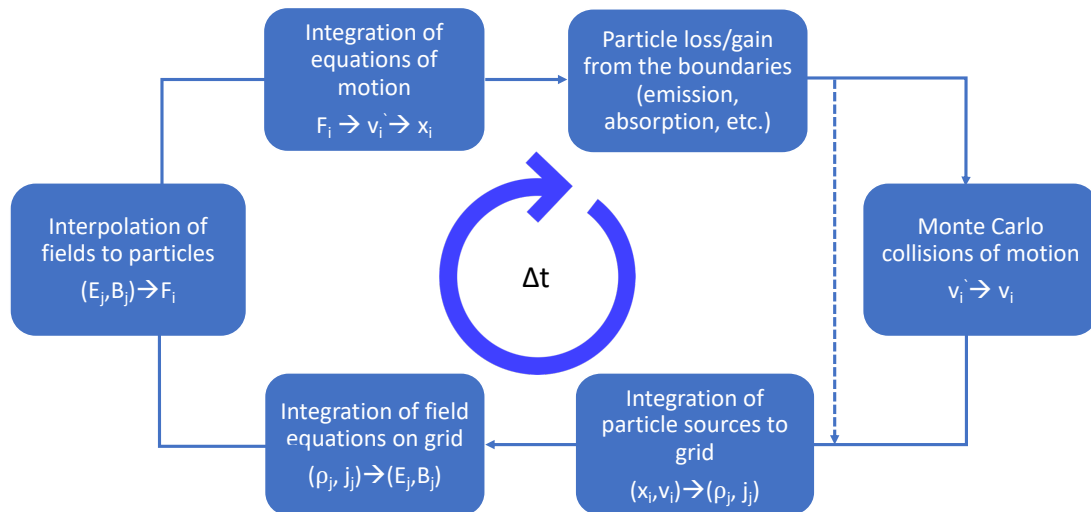


Figure 1.1: Schematic for the PIC algorithm.

Often in space plasmas, we face the situation of several orders of spatial-temporal scales, from the electron skin depth, ion inertial length, all the way to the size of the magnetosphere. The electron skin depth d_e is the depth in a collisionless

plasma to which low-frequency electromagnetic radiation can penetrate (as defined by attenuation of the wave amplitude by a factor of e^{-1}), given by $d_e = c/\omega_{pe}$, where c is the speed of light and ω_{pe} is the electron plasma frequency. Similarly, the ion inertial length is the scale at which ions decouple from electrons and the magnetic field becomes frozen into the electron fluid rather than the bulk plasma, given by $d_i = c/\omega_{pi}$, where ω_{pi} is the ion frequency. These characteristic lengths are important in PIC, especially when coupled with a global MHD model because the mesh in PIC must be able to resolve the corresponding length scales to simulate the kinetic physics. Practically we use some standard tricks to shorten the separation of scales like artificially lowering the speed of light and the ion-electron mass ratio. However, the total number of cells may still be too large for current computer capability. For example, near the dayside reconnection site at Earth $\lambda_i \approx 0.01R_E$, where R_E is the Earth's radius, which means that $d_e \approx 0.001R_E$ for the $m_i/m_e = 100$ mass ratio. However, in Ganymede's magnetosphere $\lambda_i \approx 0.1R_G$ where R_G is Ganymede's radius. This relatively close ratio makes the coupling of a kinetic PIC region inside a global MHD model possible.

Traditional PIC models are limited by computational efficiency. This is a consequence of (1) the statistical model in which numerical fluctuations converge as $N^{1/2}$ for N particles; (2) explicit time stepping PIC schemes that require to spatially resolve the Debye length, the scale beyond which plasma can be considered neutral. To relax the spatial resolution requirement, a new type of **implicit time stepping** PIC algorithm is proposed by [Mason \[1981\]](#), [Brackbill and Forslund \[1982\]](#), [Lapenta et al. \[2006\]](#) and implemented by [Markidis et al. \[2010\]](#), which is later improved by [Lapenta \[2017\]](#) with a combination of explicit particle pusher and implicit field solver and [\[Chen and Tóth, 2019\]](#) with satisfaction of charge conservation. The

semi-implicit integration in time of the Vlasov-Maxwell system removes the numerical stability constraints encountered in explicit time-stepping and enables kinetic plasma simulations at MHD time scales. This loosening of spatial and temporal resolution makes the coupling between the large scale MHD description of magnetosphere and the small scale PIC description of local reconnection regions feasible, which is the key of the state-of-the-art MHD-EPIC model [Daldorff et al., 2014]. More details are presented in Chapter II about the semi-implicit PIC and the coupling procedure with application on Ganymede’s magnetosphere.

1.3 Exploration of Ganymede

Ganymede has been studied for over 400 years with many interesting observations. Ganymede is the seventh satellite and the third Galilean moon of Jupiter, which is also the largest and most massive moon in the solar system. Ganymede’s discovery is credited to Galileo Galilei, who was the first to observe it on January 7, 1610. [Galilei, 2016] The satellite’s name was soon suggested by astronomer Simon Marius, after the mythological Ganymede, kidnapped by Zeus to serve as a cupbearer for the gods as well as a lover for him. Beginning with Pioneer 10, several spacecraft have explored Ganymede. The Voyager probes, Voyager 1 and 2, refined measurements of its size, while the Galileo spacecraft discovered its underground ocean and magnetic field.

1.3.1 Orbit and rotation

Ganymede orbits Jupiter in seven days and three hours at an average distance of 1,070,400 km. Like most known moons, Ganymede is tidally locked, with the same side always facing toward the planet. [Hartmann and Miller, 2005] Its orbit is slightly eccentric and inclined relative to the Jovian equator, with the eccentricity

and inclination changing quasi-periodically due to solar and planetary gravitational perturbations on a timescale of centuries. These orbital variations cause the axial tilt (the angle between rotational and orbital axes) to vary between 0 and 0.33° . [Bills, 2005]

Ganymede participates in orbital resonances, or more specifically Laplace resonance [Showman and Malhotra, 1997], with two other moons Europa and Io: for every orbit of Ganymede, Europa orbits twice and Io orbits four times. [Musotto et al., 2002]

1.3.2 Internal structure

Ganymede appears to be fully differentiated, with an internal structure consisting of an iron-sulfide, iron-nickel core, a silicate mantle and outer layers of water ice and liquid water. [Showman and Malhotra, 1999, Sohl et al., 2002]

The precise thicknesses of the different layers in the interior of Ganymede depend on the assumed composition of sil-

icates (fraction of olivine and pyroxene) and amount of sulfur in the core. Ganymede has the lowest moment of inertia factor, 0.31 [Showman and Malhotra, 1999], among the solid Solar System bodies. This results in a fully differentiated interior, as a consequence of its substantial component of low density water ice and tidal heating due to the Laplace resonance.

Independent lines of evidence are most simply explained by, but do not require, the existence of a subsurface liquid water ocean.

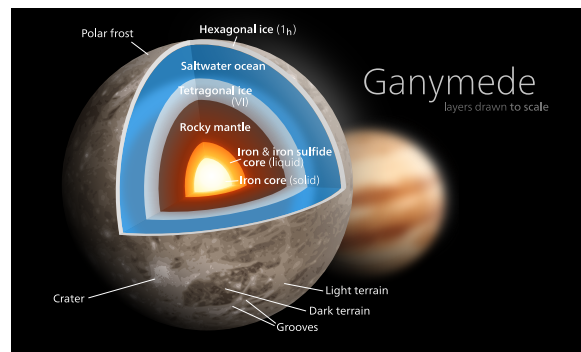


Figure 1.2: Artist's cut-away representation of the internal structure of Ganymede [Wiki, a]. Layers drawn to scale.

- Analysis of the Doppler shift of radio signals from Galileo has now shown that Ganymede is strongly differentiated with a relatively dense core surrounded by a thick shell of ice. [Anderson et al. \[1996\]](#)
- The magnetic field perturbations near the moon indicate an induced dipole which would most likely result from the existence of a conducting salty subsurface ocean. [Kivelson et al., 1996](#)
- Measurements with the Hubble Space Telescope of how the auroras move over Ganymede’s surface. [Saur et al., 2015](#)
- Analogy between Europa and Ganymede due to their similarities and geological evidence on Europa implying an underlying ductile layer of either liquid water or soft ice. [Carr et al., 1998](#)

1.3.3 Magnetosphere

The interaction between the Ganymedian magnetosphere and Jovian plasma is in many respects similar to that of the solar wind and Earth’s magnetosphere. The plasma co-rotating with Jupiter impinges on the trailing side of the Ganymedian magnetosphere much like the solar wind impinges on the Earth’s magnetosphere. Ganymede’s **magnetopause**, as an analogy to Earth’s magnetopause, is the boundary between the moon’s magnetic field and the co-rotating Jovian plasma. The main difference is the speed of plasma flow — super-Alfvénic/supersonic in the case of Earth and sub-Alfvénic/subsonic in the case of Ganymede. Because of the subsonic flow, there is no bow shock off the trailing hemisphere of Ganymede.

Instead, an Alfvén wing structure arises from the interaction between the sub-Alfvénic flow and the conducting moon, shown as the heavy solid lines in Figure 1.3(a). As a magnetic field encounters an conducting obstacle and starts to bend,

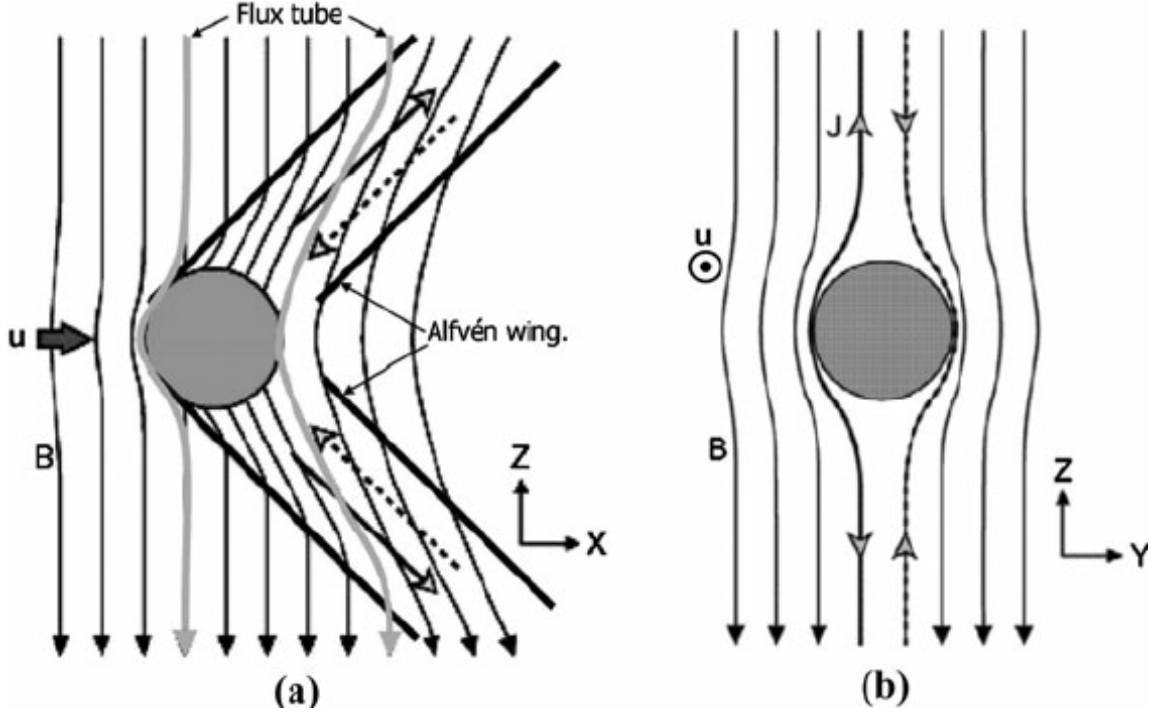


Figure 1.3: A schematic illustration of the Alfvén wing structure arising from the interaction between a sub-Alfvénic flow and a conducting obstacle. Panel (a) represents the X-Z plane that contains the directions of the unperturbed flow ($+\hat{x}$) and the background magnetic field ($-\hat{z}$). Panel (b) shows the cross-section of the interaction region in the Y-Z plane normal to the plasma flow direction. The Alfvén wing currents flowing along the field lines are represented by arrows. Solid and dashed lines represent the currents flowing on the $-\hat{y}$ and $+\hat{y}$ side, respectively. The figure is adapted from [Kivelson et al., 2004].

Alfvén waves are generated along the field lines away from that point. These Alfvén waves propagate along the magnetic field line with the speed $V_A = B/\sqrt{\mu_0\rho}$, where μ_0 is the vacuum permeability, B is the magnetic field strength, and ρ is the plasma mass density. In addition, the plasma still advects the magnetic field with a given velocity V . The Alfvén wave, therefore, travels at an angle $\theta = \tan^{-1}(u/V_A)$, where u and V_A are the flow speed and the Alfvén speed of the unperturbed flow, respectively [Drell et al., 1965]. The flow diverts around the obstacle and forms two tubes (above and below) the object in which the flow characteristics of the plasma are altered significantly from the surrounding medium. This structure is the Alfvén wing, which can be very long due to the nearly constant Alfvén wave amplitude. Far from the

moon in the Alfvén wing, the slow mode waves are not present since they do not propagate in the direction of the Alfvén wing, and the fast mode waves decay with the distance from the moon, so the interaction becomes pure Alfvénic. [Neubauer, 1998]

Most of the observation data came from the instruments onboard Galileo spacecraft. In the study of magnetosphere, we are mostly concerned with the magnetometer (MAG) data, the Plasma Subsystem (PLS) data for the charged particles' mass and energy density, the Plasma Wave Subsystem (PWS) for plasma waves, and Energetic Particles Detector (EPD) for high energy (> 20 keV) particles.

The Galileo spacecraft made six close flybys of Ganymede from 1995-2000 (G1, G2, G7, G8, G28 and G29). The G1, G2, G7 and G29 flybys went across the Alfvén wing near the north pole in the slightly downstream region, while G8 and G28 flybys went through the open-closed field line boundary at low latitudes on the upstream side. Researchers have discovered that Ganymede has a permanent (intrinsic) magnetic moment independent of the Jovian magnetic field [Kivelson et al., 2002], probably created by convection within its liquid iron core. The relatively weak magnetic field is buried within Jupiter's much larger magnetic field and would create only local perturbations of the Jovian field lines. The value of the moment is about $1.3 \times 10^{13} \text{ T} \cdot \text{m}^3$, which is three times larger than the magnetic moment of Mercury. The magnetic dipole is tilted with respect to the rotational axis of Ganymede by 176° , which means that it is directed against the Jovian magnetic moment. The dipole magnetic field created by this permanent moment has a strength of 719 ± 2 nT at Ganymede's equator.

The satellite has a thin oxygen atmosphere that includes O , O_2 , and possibly O_3 (ozone). [Hall et al., 1998] Atomic hydrogen is a minor atmospheric constituent.

Whether the satellite has an ionosphere associated with its atmosphere is unresolved. On one hand, the existence of a neutral atmosphere implies that an ionosphere should exist, because oxygen molecules are ionized by the impacts of the energetic electrons coming from the magnetosphere [Paranicas et al., 1999] and by solar EUV radiation [Eviatar et al., 2001b]. On the other hand, some Galileo measurements such as PWS found an elevated electron density near Ganymede during the G1 and G2 flyby, suggesting an ionosphere, whereas others failed to detect anything. Despite of the contradictions, physical-chemistry model of the solar UV flux on Ganymede’s atmosphere [Cessateur et al., 2012] and test particle model of Ganymede’s ionosphere [Carnielli et al., 2019, 2020] are applied to constrain the constitution of species and electron-impact ionization rate.

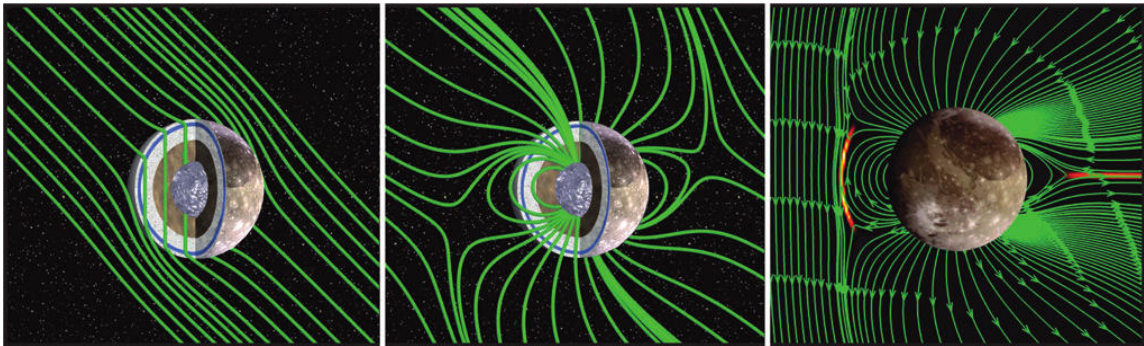


Figure 1.4: Ganymede’s induced field (left), internally generated magnetic field (middle), and resulting miniature magnetosphere (right). The induced magnetic field is most certainly generated by electrical currents in a liquid ocean trapped between the icy crust and the high-pressure icy layer. [Credits: X. Jia (University of Michigan) and M. Kivelson (UCLA).] Color images available online at www.liebertonline.com/ast

The permanent magnetic moment carves a part of space around Ganymede and creates a tiny magnetosphere embedded inside that of Jupiter’s, which is confirmed with synergetic Galileo multi-instrument observations [Kivelson et al., 1996, Anderson et al., 1996, Schubert et al., 1996, Gurnett et al., 1996]. The main ion species in the magnetosphere is singly ionized oxygen O^+ , which fits well with Ganymede’s tenuous oxygen atmosphere. In the polar cap regions, magnetic field lines are open,

connecting Ganymede with Jupiter's ionosphere. In these areas, energetic (tens and hundreds of keV) electrons and ions have been detected, which may cause the auroras observed around the Ganymedian poles. Meanwhile, heavy ions precipitate continuously on Ganymede's polar surface, sputtering and darkening the ice.

In addition to the intrinsic magnetic moment, Ganymede has an induced dipole magnetic field, the existence of which is connected with the variation of the Jovian magnetic field near the moon's trajectory. The induced moment is directed radially to or from Jupiter following the direction of the varying part of the planetary magnetic field, and is an order of magnitude weaker than the intrinsic one. The induced magnetic field of Ganymede is similar to those of Callisto and Europa, indicating that Ganymede also has a subsurface water ocean with a high electrical conductivity. A demonstration of the induced field, intrinsic field, and the resulting miniature magnetosphere is shown in Figure 1.4.

Ganymede's magnetosphere remains enigmatic, particularly given that similar bodies do not have a magnetosphere. As an ideal object of reconnection studies, there are still many unanswered questions:

1. What are the signatures of reconnection at Ganymede's magnetopause?
2. What are the properties of the flux transfer events (FTEs) at the upstream magnetopause?
3. How efficient is the upstream reconnection process quantitatively?
4. Are there any intrinsic periodicities in the interaction between Jovian plasma and Ganymede's magnetosphere?

We are trying to construct physical models based on current observations, analyze and predict the behavior of Ganymede's magnetosphere.

1.3.4 Magnetosphere Modeling

Through numerical simulations, many of the observational findings have been confirmed and well explained. The first 3D resistive MHD model for Ganymede's magnetosphere was published in 2002 [Kopp and Ip, 2002], in which researchers described how the magnetic field configuration of Ganymede's magnetosphere could change under different external plasma conditions. A follow-up study [Ip and Kopp, 2002] using the same model linked the ion acceleration and sputtering effects associated with magnetopause reconnection. Later a multi-ion MHD model [Paty and Winglee, 2004, 2006, Paty et al., 2008] was adapted to Ganymede to study the behavior of different ion species within Ganymede's magnetosphere, such as estimation of ion sputtering rates, energy distribution and ionospheric outflow. A different resistive MHD model was applied to Ganymede by Jia et al. [2008], where they coupled, for the first time, the moon's interior to the global magnetosphere. Later they refined their MHD model by developing improved inner boundary conditions and incorporating an anomalous resistivity model that allows for simulating fast reconnection [Jia et al., 2009]. The new model not only yields satisfactory agreement with the Galileo observations but also predicts that Ganymede's magnetopause reconnection occurs in a non-steady manner under fixed upstream conditions [Jia et al., 2010]. [Duling et al., 2014] later proposed a new treatment of non-conducting surface boundary in the resistive MHD frame, which also achieves nice agreement with the Galileo magnetometer data. Meanwhile, [Dorelli et al., 2015] extended the MHD model to include Hall effect, which allows asymmetries and ion drifts inside the magnetosphere. [Fatemi et al., 2016] presented a hybrid model for constructing the electromagnetic field of the magnetosphere and a test particle model with the field information as input to demonstrate that the precipitating energetic particle fluxes are the primary

driver for altering the surface brightness of Ganymede. Recently, [Wang et al., 2018] have employed a 10-moment closure model for Ganymede with electron kinetics included, which is shown to have the potential of capturing local electron and ion kinetics within global magnetosphere simulations.

The coupled fluid-kinetic model [Tóth et al., 2016], which is the predecessor of the current model used in this study, embed a local kinetic Particle-in-Cell (PIC) region inside the global Hall MHD domain. This approach allows resolving important kinetic processes near the reconnection site, which is of great interest to magnetosphere study. We have improved the model by adding a resistive layer at the inner boundary, switching to high resolution spherical grid for better refinement inside the magnetosphere, and employing a new charge and energy conserving PIC scheme for reducing numerical errors. Our study focuses on the structure and dynamics of the magnetosphere, with special emphasis on the upstream magnetic reconnection, using numerical modelling. The model is thoroughly introduced in Chapter II. All the input parameters used in our simulations are inferred from the observations by the Galileo spacecraft that recorded six close encounters of Ganymede during its eight years in the Jovian system. The improvements and inferences and presented in Chapter III, and the detailed reconnection-driven dynamics at Ganymede’s upstream magnetopause are presented in Chapter IV.

1.4 Modern Supercomputer Architecture

High fidelity physical models nowadays cannot be built without computers. Parallel computing has risen to prominence since the 1990s with the advancement of supercomputing power. Supercomputers in 1990s were faster than PCs because of higher clock speed. Current supercomputers are fast not because of high single core

frequency, but because of the large scale of cores involved. The speed of a supercomputer is typically measured in *Floating Point Operations Per Second (FLOPS)*. The High Performance Linpack (HPL) [Dongarra et al., 2003], which solves a dense system of linear equations, is chosen as a standard benchmark for the performance. Intel ASCI Red was the first TFLOPS (Tera, or 10^{12} FLOPS) supercomputer. In 2008, IBM Roadrunner became the first supercomputer with the speed of PFLOPS (Peta, or 10^{15} FLOPS). As of November 2019, Oak Ridge National Laboratorys Summit system holds top honors with an HPL result of 148.6 petaflops, and the aggregate performance of the top 500 system sits at 1.65 exaflops. [Organization, 2019] The next breakthrough in the area of supercomputing will be the Exascale supercomputer with the processing speed to be measured in Exa-FLOPS (10^{18}).

Figure 1.5 shows the exponential growth of computing power for the top 500 supercomputers in the past decade. The fitted lines show that in all 3 measures the computing power increases by one magnitude every 5 years, although in the last 10 years the progress has slightly slowed down. As an illustration, the 20-minute time-accurate simulation in this thesis research takes 230 hours to finish with 2000 cores on the fifth fastest machine Frontera, which is in total 0.46 million core hours. If this growth trend continues and our new model suits the massive parallel architecture, within 10 years we will be able to do the same simulation in less than 3 hours.

1.4.1 Parallelism

Parallelism means that the computational tasks are carried out in parallel and dealt with simultaneously. Current global magnetosphere research models usually contain tens of millions of grid cells, which are divided into regions or blocks and solved in parallel by the computers. Two different types of parallelisms are supported:

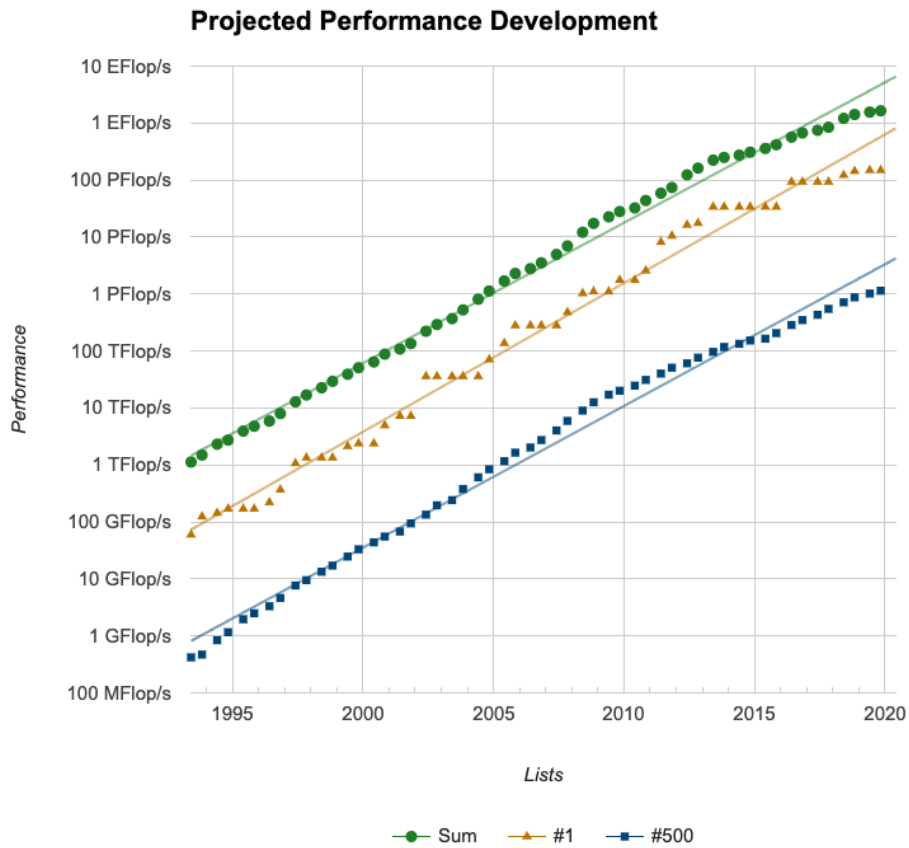


Figure 1.5: The performance in terms of FLOPS for the top 500 supercomputers from 06/1993 to 11/2019.

1. multithreading
2. distributed processing

In the high performance computing community, the multithreading is typically handled by **Open Multi-Processing (OpenMP)** and the distributed processing is typically handled by the **Message Passing Interface (MPI)**. As an extension to CPU-based computing, GPU computing has risen up in recent years for algebraic-intensive work, such as machine learning and particle simulations. The strengths and weaknesses of each of them is further discussed in Chapter V. Preliminary attempts

of MHD computation offload to GPU with **OpenACC**, a programming standard for simplifying parallel programming of heterogeneous CPU/GPU systems, is also presented. Parallelism is extensively used in this thesis work, from large scale physical model development, post-processing, to data analysis.

1.4.2 Platform

The computing resources for this thesis research were mostly provided by NSF flagship supercomputers Bluewaters and Frontera. Table 1.2 summarized the key parameters of these two supercomputers.

Table 1.2: Platform Information

Name	Blue Waters	Frontera
Processor	AMD 6276 Interlagos	Intel Xeon Platinum 8280 ("Cascade Lake")
Sockets/Node	2	2
Cores/Node	32*	56
Threads/Core	1	1**
Clock rate (GHz)	2.3	2.7GHz***
RAM (GB/core)	2	3.4
Memory Bandwidth (GB/s)	102.4	228.2
Fortran compiler used	ifort, gfortran	ifort

* The Interlagos processor has 8 Bulldozer cores, but viewed as 16 "processors" by the Linux system. These "processors" are the schedulable integer cores that work with the floating point unit.

** Hyperthreading is off by default.

*** 1.4-3.7GHz depending on instruction set and number of active cores, 2.7GHz is the nominal value.

CHAPTER II

Model Description

We present the **MHD with embedded Particle-in-Cell** model, i.e. **MHD-EPIC**, in great detail, and emphasize its application on Ganymede’s magnetosphere study. Simulations presented in this thesis are performed with the **Space Weather Modeling Framework (SWMF)** [Tóth et al., 2012]. The global magnetosphere component (GM) is running the magnetohydrodynamics model **BATS-R-US**, and the kinetic particle-in-cell component (PC) is running the PIC model **iPIC3D** or **GL-ECSIM**. The two models are coupled together within SWMF, making it possible to obtain detailed kinetic physics inside the global fluid description of magnetosphere. Each model is introduced separately below, followed by the specific setup for Ganymede’s environment during the Galileo close encounter flybys.

2.1 Space Weather Modeling Framework

The **Space Weather Modeling Framework (SWMF)** is a collection of loosely coupled, physics-based numerical models for simulating space weather and space physic processes on a wide range of spatiotemporal scales [Tóth et al., 2005, 2012, CSE]. On the top level, the framework is responsible for the parallel execution of the models and the data transfer required by the coupling of the models through Message Pass Interface (MPI). Common utilities like linear solvers, mesh interpolation

algorithms, coordinate transformations and timings are provided through the shared libraries. The development goal is to minimize the cost of coupling different numerical models while maintaining their independence. User manuals and documents can be found at the [CSEM](#) website.

2.2 BATSURUS

The **Block-Adaptive-Tree Solarwind Roe Upwind Scheme (BATS-R-US)** [[Powell et al., 1999](#), [De Zeeuw et al., 2000](#)] is a multi-physics MHD code written in Fortran 90+ that has been actively developing at the University of Michigan for over 20 years. It is the most complex and often the most computationally expensive model in SWMF that has been applied to simulate multi-scale space physics systems including, but not limited to, the solar corona, the heliosphere, planetary magnetospheres, moons, comets and the outer heliosphere. For the purpose of **adaptive mesh refinement (AMR)** and running efficiency, the code was designed from the very beginning to use a 3D Cartesian block-adaptive mesh with MPI parallelization [[Stout et al., 1997](#), [De Zeeuw et al., 2000](#)]. In 2012, the original block-adaptive implementation has been replaced with the newly designed and implemented **Block Adaptive Tree Library (BATL)** [[Tóth et al., 2012](#)] for creating, adapting, load-balancing and message-passing a 1, 2, or 3 dimensional block-adaptive grid in generalized coordinates. The major advantages of adaptive block approach include locally structured grid in each block, cache optimizations due to relatively small arrays associated with the grid blocks, loop optimization for fixed sized loops over cells in the block, and simple load balancing. Larger blocks reduce the total number of ghost cells surrounding the grid blocks, but make the grid adaptivity less economic. Smaller blocks allow precise grid adaptation, but require a large number of blocks

and more storage and computation spent on ghost cells. The typical choice of block size in 3D ranges between 4^3 to 16^3 grid cells, with an additional 1-3 layers of ghost cells on each side depending on the order of the numerical scheme.

BATS-R-US has been gradually evolving into a comprehensive code by adding new schemes as well as new physical models. Currently, 60 equation sets from ideal hydrodynamics to the most recent six-moment fluid model [Huang et al., 2019] are available. The most important applications solve various forms of the magnetohydrodynamic (MHD) equations, including resistive, Hall, semi-relativistic, multi-species and multi-fluid MHD, optionally with anisotropic pressure, radiative transport and heat conduction. There are several choices of numerical schemes for the Riemann solvers, from the original Roe scheme to many others combined with a second order total variation diminishing (TVD) scheme or a fifth order accurate conservative finite difference scheme [Chen et al., 2016]. The time discretization can be explicit, point-implicit, semi-implicit, explicit/implicit or fully implicit. A high level abstraction of the code structure is presented in Figure 2.1.

A powerful feature of BATS-R-US is the incorporation of user modules. This is an interface for users to modify literally any part of the kernel code without interfering with other modules. It provides a neat and easy way to gain high-level control of the simulations. Currently there are 51 different user modules in the repository, mostly used for the setup of specific initial and boundary conditions and additional user-defined source terms for the specific applications.

2.2.1 Hall MHD + Electron Pressure Equation

The Hall MHD module of BATS-R-US is initially described in [Tóth et al., 2008]. As an extension to the ideal/resistive MHD model, Hall MHD decouples the electron

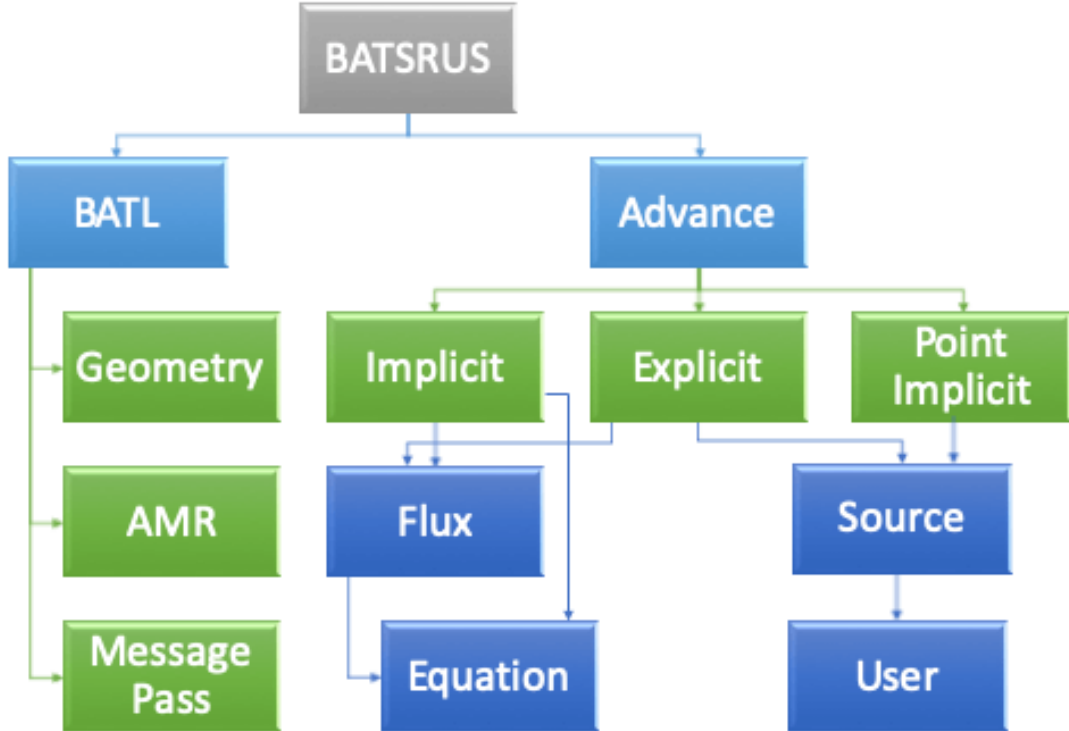


Figure 2.1: The high level structure of BATS-R-US. The AMR library BATL is used for mesh generation, refinement and message passing. Explicit, fully-implicit, semi-implicit, part-implicit and point-implicit schemes can be used for time advance. Numerical flux schemes of 1st, 2nd and 5th order can be chosen. Multiple versions of the equation and user modules containing the equation variable definitions and the application specific codes are available.

and ion motions by retrieving the Hall term in the generalized Ohm’s law

$$\mathbf{E} = -\mathbf{u} \times \mathbf{B} + \eta \mathbf{J} + \frac{1}{en} \mathbf{J} \times \mathbf{B} - \frac{\nabla p_e}{ne}, \quad (2.1)$$

where the first term on the right-hand-side is the convective term, the second is the resistive term, the third is the Hall term, and the fourth is the electron pressure gradient term. The electron pressure scalar is simplified from the electron pressure tensor, and is obtained from an independent electron pressure equation. For all the simulations of Ganymede’s magnetosphere, we choose an explicit-implicit timestepping scheme ((the Hall and resistive terms are solved implicitly)) and a numerical flux of Sokolov scheme [Sokolov et al., 1999] with third-order monotized central (Koren) limiter in BAT-S-RUS. The Hall effect is restricted to $|x| < 5R_G$, $|y| < 4R_G$ and $|z| < 4R_G$ box region excluding a sphere of radius $1.05R_G$ centered at the moon

to speed up the simulation. (x, y, z) is defined in the GphiO system, where \hat{x} is along the flow direction, \hat{y} is along the Ganymede-Jupiter vector with positive direction pointing towards Jupiter, and \hat{z} is along the spin axis.

The Hall MHD equations (with electron pressure gradient term and separate electron pressure equation) to be solved are

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho \mathbf{u}), \quad (2.2)$$

$$\frac{\partial(\rho \mathbf{u})}{\partial t} = -\nabla \cdot \left(\rho \mathbf{u} \mathbf{u} + (p + p_e) \bar{I} + \frac{B^2}{2\mu_0} \bar{I} - \frac{\mathbf{B}\mathbf{B}}{\mu_0} \right), \quad (2.3)$$

$$\frac{\partial e}{\partial t} = -\nabla \cdot \left[(\epsilon + p) \mathbf{u} + (\epsilon_e + p_e) \mathbf{u}_e + \mathbf{u}_e \cdot \left(\frac{\mathbf{B}^2}{\mu_0} \bar{I} - \frac{\mathbf{B}\mathbf{B}}{\mu_0} \right) - \mathbf{B} \times \eta \mathbf{j} \right], \quad (2.4)$$

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \left[\mathbf{u}_e \times \mathbf{B} + \eta \mathbf{j} - \frac{\nabla p_e}{ne} \right], \quad (2.5)$$

$$\frac{\partial p_e}{\partial t} + \nabla \cdot (p_e \mathbf{u}_e) = -(\gamma - 1) p_e \nabla \cdot \mathbf{u}_e. \quad (2.6)$$

where \bar{I} is the identity matrix, ρ is the mass density, \mathbf{u} is the plasma bulk velocity, \mathbf{B} is the magnetic field, p_e is the electron pressure, p is the ion thermal pressure, and $\mathbf{j} = \nabla \times \mathbf{B} / \mu_0$ is the current density. The Hall velocity is defined as

$$\mathbf{v}_H = -\frac{\mathbf{j}}{ne}, \quad (2.7)$$

and the electron bulk velocity is given by

$$\mathbf{u}_e = \mathbf{u} + \mathbf{v}_H. \quad (2.8)$$

The total energy density is

$$e = \epsilon + \epsilon_e + \frac{B^2}{2\mu_0} = \frac{1}{2} \rho \mathbf{u}^2 + \frac{1}{\gamma - 1} (p + p_e) + \frac{B^2}{2\mu_0}, \quad (2.9)$$

where ϵ is the hydrodynamic energy density and γ is the adiabatic index. Note that in our Hall MHD model only $(\rho, \mathbf{u}, \mathbf{B}, p, p_e)$ are unknowns; all others are derived quantities.

The continuity equation (2.2), momentum equation (2.3), energy equation (2.4) and electron pressure equation (2.6) are solved with an explicit timestepping scheme, while the magnetic induction equation (2.5) is solved with a semi-implicit scheme. Specifically, the convection term $\mathbf{u} \times \mathbf{B}$ and electron pressure gradient term $\nabla p_e / ne$ are advanced using explicit timestepping, while the resistivity term $\eta \mathbf{J}$ and Hall term $\mathbf{v}_H \times \mathbf{B}$ are advanced with an implicit scheme to handle the stiff problems without limiting the discrete timesteps.

The hyperbolic cleaning and eight-wave schemes [Dedner et al., 2002, Powell et al., 1999] are used to keep the divergence-free magnetic field constraint. Special treatment of small cells near the axis of symmetry of the spherical grid is taken into account by smoothing the cell-center quantities in $\hat{\phi}$ to increase the timesteps and reduce the discontinuities near the pole axis. In addition, an accurate parallel fieldline-tracing scheme is used to capture the open-closed field line boundary and magnetic field topology.

2.3 Particle-in-Cell Model

In the PIC model we use, the Maxwell's equations are differenced in space on a uniform Cartesian grid. The electric field \mathbf{E} and the current densities \mathbf{J} are evaluated at the vertices of the grid, while the magnetic field \mathbf{B} and charge densities ρ are calculated at the centers of the cells. The field storage and indexing for a single cell (i,j,k) are demonstrated in Figure 2.2.

The particle-in-cell (PIC) model consists of two major parts. First, the macro

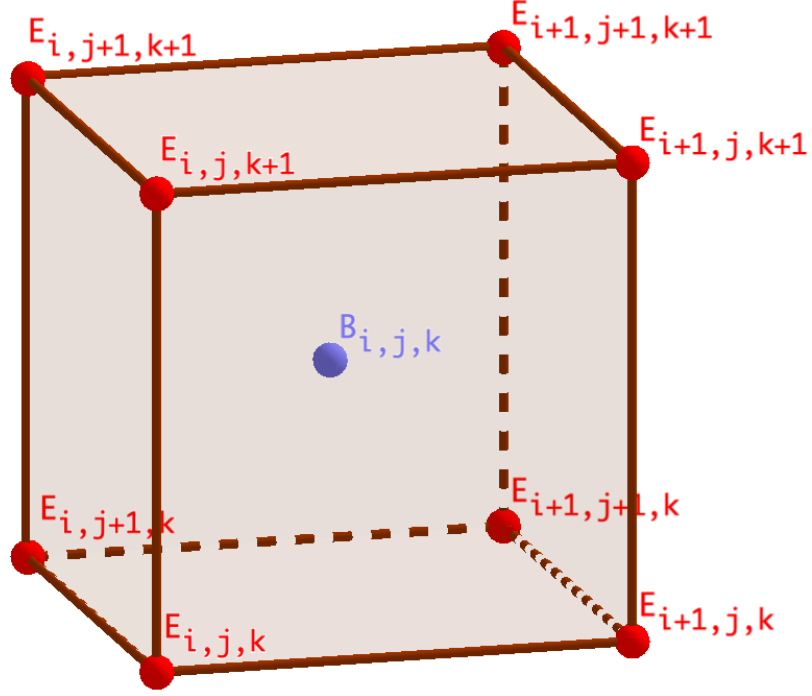


Figure 2.2: Staggered field storage in the PIC algorithm for cell index (i,j,k) . The electric field \mathbf{E} and the current densities \mathbf{J} are evaluated at the vertices of the grid, while the magnetic field \mathbf{B} and charge densities ρ are calculated at the centers of the cells.

particles satisfying the Vlasov equation are pushed via

$$\frac{d\mathbf{x}_p}{dt} = \mathbf{v}_p, \quad (2.10)$$

$$\frac{d\mathbf{v}_p}{dt} = \frac{q_p}{m_p} (\mathbf{E}_p + \mathbf{v}_p \times \mathbf{B}_p), \quad (2.11)$$

where \mathbf{x}_p is the particle position and \mathbf{v}_p is the particle velocity, E_p and B_p are field values interpolated at the particle's position. Secondly, the electric field \mathbf{E} is obtained from the second order form of Maxwell's equation

$$\epsilon_0 \mu_0 \frac{\partial^2 \mathbf{E}}{\partial t^2} - \nabla^2 \mathbf{E} = -\mu_0 \frac{\partial \mathbf{J}}{\partial t} - \frac{1}{\epsilon_0} \nabla \rho, \quad (2.12)$$

and the magnetic field is then updated from the induction equation

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E}. \quad (2.13)$$

The speed of light c is reduced to 4000km/s to decrease the number of iterations for the implicit solver, and the ion-electron mass ratio m_i/m_e is set to 100 to reduce the scale separation of electron skin depth and ion inertial length. We have tested two different numerical schemes in solving the above equations: iPIC3D and GL-ECSIM.

2.3.1 Implicit Particle-in-Cell Method

The implicit 3D PIC (iPIC3D) code uses a 3-step Picard iteration for the particle mover as well as an implicit GMRES solver [Saad and Schultz, 1986] for the electric field. The complete process is described in [Markidis et al., 2010]. It was later discovered that by not conserving either momentum and energy, unphysical numerical waves are generated near the magnetopause from our coupled runs. Therefore we have developed and implemented a more accurate PIC algorithm presented in the next subsection.

2.3.2 Energy Conserving Semi-Implicit Method

The Energy Conserving Semi-Implicit Method (ECSIM) from [Lapenta, 2017] has recently been implemented into the in-house iPIC3D model, and has been improved with better stability, charge conservation, and particle splitting-merging algorithm. The new model is called Gauss's Law satisfying Energy Conserving Semi-Implicit Method (GL-ECSIM) [Chen and Tóth, 2019]. Here we present the key steps in the algorithm.

The macro particles with position \mathbf{x}_p and velocity \mathbf{v}_p are updated explicitly via

$$\mathbf{x}_p^{n+1/2} = \mathbf{x}_p^{n-1/2} + \Delta t \mathbf{v}_p^n, \quad (2.14)$$

$$\mathbf{v}_p^{n+1} = \mathbf{v}_p^n + \frac{q_p \Delta t}{m_p} \left[\mathbf{E}^{n+\theta} + \frac{\mathbf{v}_p^n + \mathbf{v}_p^{n+1}}{2} \times \mathbf{B}^n \right], \quad (2.15)$$

where the position \mathbf{x}_p and velocity \mathbf{v}_p are evaluated alternately in a leapfrog style, the electric field is evaluated at $n + \theta$ time level, and magnetic field \mathbf{B}^n is evaluated at n time step [Lapenta, 2017].

The electric and magnetic fields are updated implicitly at the same time

$$\frac{\mathbf{B}^{n+1} - \mathbf{B}^n}{\Delta t} = -c \nabla \times \mathbf{E}^{n+\theta}, \quad (2.16)$$

$$\frac{\mathbf{E}^{n+1} - \mathbf{E}^n}{\Delta t} = -c \nabla \times \mathbf{B}^{n+\theta} - 4\pi \bar{\mathbf{J}}, \quad (2.17)$$

where $\bar{\mathbf{J}}$ is the predicted current at $n + \theta$ time stage, and it depends on the unknown electric field $\mathbf{E}^{n+\theta}$. The values at time level $n + \theta$ are defined as a linear combination of the values at the n and $n + 1$ stages with weights $1 - \theta$ and θ :

$$\mathbf{E}^{n+\theta} = (1 - \theta) \mathbf{E}^n + \theta \mathbf{E}^{n+1} \quad (2.18)$$

$$\mathbf{B}^{n+\theta} = (1 - \theta) \mathbf{B}^n + \theta \mathbf{B}^{n+1} \quad (2.19)$$

Substituting Equation 2.18, 2.19 into 2.16 and 2.17, we obtain the second order implicit equation of $\mathbf{E}^{n+\theta}$:

$$\mathbf{E}^{n+\theta} + (c\theta\Delta t)^2 \left[\nabla(\nabla \cdot \mathbf{E}^{n+\theta}) - \nabla^2 \mathbf{E}^{n+\theta} \right] = \mathbf{E}^n + c\theta\Delta t \left(\nabla \times \mathbf{B}^n - \frac{4\pi}{c} \bar{\mathbf{J}} \right), \quad (2.20)$$

where $\bar{\mathbf{J}}$ is the predicted current at $n + \theta$ time stage, which depends on the unknown electric field $\mathbf{E}^{n+\theta}$ [Lapenta, 2017]. Note that the discretized equations are written in CGS units for convenience in the code.

The exact energy conservation can be achieved only if $\theta = 0.5$ and proper spatial discretizations are used. However, simulation tests show that with $\theta = 0.5$ it may

still create numerical waves [Chen and Tóth, 2019]. In our coupled simulation, we choose $\theta = 0.51$ that sacrifices the energy conservation a little bit but significantly improves the robustness. Besides, the above procedure does not guarantee charge conservation, so after the updates we apply methods of correction to reduce the numerical artifacts.

The new scheme is applied to the MHD-EPIC model presented in Chapter III and IV.

2.4 MHD-EPIC

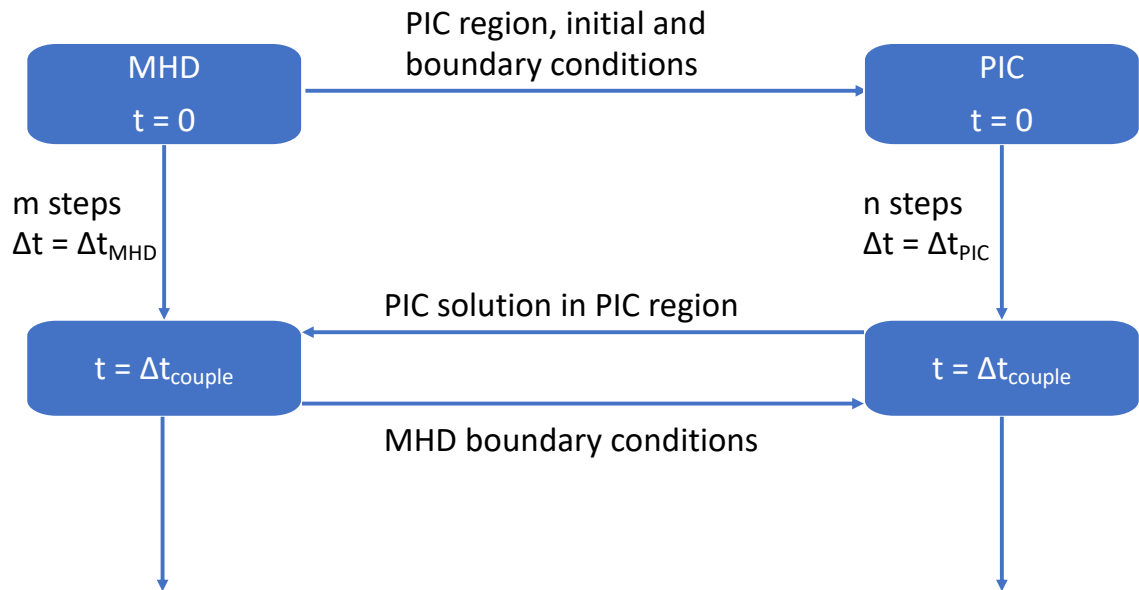


Figure 2.3: Temporal discretization of the MHD-EPIC coupling algorithm, based on Figure 1 in [Daldorff et al., 2014].

The global computational domain is simulated by a Hall-MHD model, with an embedded PIC region at the upstream magnetopause. The MHD and PIC models exchange information for plasma and electro-magnetic field through the framework. These two models are coupled together through SWMF and form the MHD-EPIC fluid-kinetic model, as shown in Figure 2.3. Multiple versions of MHD-EPIC have

Table 2.1: Simulation Parameters for Galileo’s Six Close Encounters

Flyby	Ganymede’s Dipole Moment						Background Flow				
	M_z [nT]	M_y [nT]	M_x [nT]	B_x^{bk}	B_y^{bk}	B_z^{bk}	v [km/s]	ρ [amu/cm ³]	P [nPa]	M_A	β
G1	-716.8	82.5	-24.7	6	-79	-79	140	28	1.9	0.30	0.38
G2	-716.8	80.0	-29.3	17	-73	-85	140	28	1.9	0.30	0.38
G7	-716.8	14.0	-20.9	-3	84	-76	130	28	1.9	0.28	0.37
G8	-716.8	51.8	-18.0	-10	-6	-86	140	56	3.8	0.55	1.60
G28	-716.8	17.0	-19.3	-7	78	-76	140	28	1.9	0.31	0.41
G29	-716.8	84.2	-18.4	-9	-83	-79	140	28	1.9	0.30	0.36

been successfully applied to Mercury [Chen et al., 2019], Earth [Chen et al., 2017], Mars [Ma et al., 2018], and Ganymede [Tóth et al., 2016, Zhou et al., 2019]. As an extension to the previous MHD-EPIC modeling work on Ganymede [Tóth et al., 2016], we now solve the Hall MHD equations with a separate electron pressure equation in BAT-S-RUS. The magnetic induction equation is solved throughout the mantle of the moon to allow for the magnetic field to diffuse through the planetary body. Electrons and ions inside the upstream reconnection region covered by the PIC box are simulated with the improved GL-ECSIM scheme [Zhou et al., 2019, accepted, 2020].

2.5 Ganymede’s Magnetosphere Simulation

For the simulations presented in the next two chapters, we use the same initial and boundary conditions as described below, but different grid resolutions. The grid information is given in each chapter respectively.

2.5.1 Initial Conditions

Ganymede’s magnetic field is composed of a permanent dipole generated by the core and an induced dipole from interactions with the time-varying Jovian magnetic field [Kivelson et al., 2002]. Both dipole moments can be calculated from statistical fitting of Galileo magnetometer data over flybys, while higher order moments are ignored in our simulations. Table 2.1 lists the total dipole moments for each flyby.

The initial conditions for primitive variables are based on the upstream conditions. The plasma density is set to the observation average of 4cm^{-3} for G8 flyby where Ganymede’s orbit is inside Jupiter’s current sheet and 2cm^{-3} for other flybys with an average mass per unit charge 14 amu representing a mixture of H^+ and O^+ . We assume ions are singly charged and treat energetic and thermal ions as a single fluid. The total thermal pressure of the ambient plasma is set as $p_t = 3.8$ nPa for G8 flyby, and $p_t = 1.9$ nPa for the other five flybys which occurred away from the central plasma sheet, according to [Kivelson et al., 2004].

2.5.2 Boundary Conditions

There are three kinds of boundaries in the BAT-S-RUS model: the outer boundary (cut off by a box), the surface boundary ($r = 1R_G$), and the core boundary ($r = 0.5R_G$). Figure 2.4 shows a sketch of the three boundary geometries.

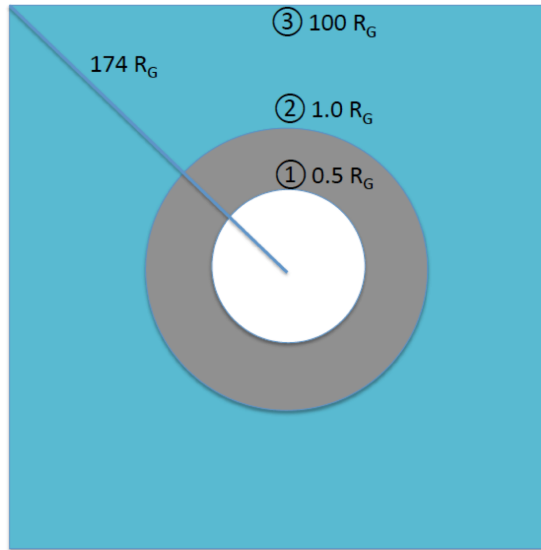


Figure 2.4: Sketch of the boundaries in the MHD model. 1 represents the core boundary at $r = 0.5R_G$, 2 represents the surface boundary at $r = 1R_G$, and 3 represents the outer boundary.

We are using steady upstream conditions for each flyby simulation since Jupiter’s rotation period (~ 10 h) and Ganymede’s orbital period (~ 7.15 d) are much longer

than the time it takes for Jupiter’s corotating plasma to flow past Ganymede’s magnetospheric system, which is on the order of minutes [Jia et al., 2008]. For the outer boundaries, we specify primitive variables $\rho, \mathbf{u}, \mathbf{B}, p, p_e$ at upstream and downstream faces with constant values corresponding to those observed by Galileo for each individual pass as shown in Table 2.1. The ion-electron temperature ratio is set to 18 based on Table 21.1 in [Kivelson et al., 2004]. We apply a fixed boundary condition at the upstream and downstream faces because the flow is subsonic and sub-Alfvénic. The other sides of the outer box boundary are set with zero-gradient boundary conditions to allow for the Alfvén wings, although fixed boundaries would also work.

For the inner boundaries, we followed Jia et al. [2009] with special care on velocities and magnetic fields. The dipole field in our model is set as the sum of a permanent dipole \mathbf{B}_0 and a prescribed induced dipole field \mathbf{B}_1 due to the interaction between the time-varying background field and the conducting layer calculated by Kivelson et al. [2002]. The dipole moments at the core boundary $0.5R_G$ are shown in Table 2.1. In the MHD model we only solve for \mathbf{B}_1 while keeping \mathbf{B}_0 constant. The semi-implicit scheme implemented in BAT-S-RUS allows us to solve for $\partial\mathbf{B}/\partial t$ contributed from the magnetic diffusion term separately from the convection term. This capability has been successfully applied to model the induction effect at Mercury [Jia et al., 2015], and becomes more robust with the new option of setting multi-boundary layer in BATS-R-US. During each timestep, the surface boundary ($r = 1R_G$) is turned on at first to solve for $\rho, \mathbf{u}, \mathbf{B}, p, p_e$ from surface boundary to the outer boundary. A mass density of 550 amu/cm^3 with an average temperature of 20 eV (corresponding to pressure $p = nk_B T = 0.125 \text{ nPa}$) are fixed, and the flow velocity is set to be continuous perpendicular to the local magnetic field and zero

Table 2.2: Boundary conditions

	Core Boundary ($r = 0.5R_G$)	Surface Boundary ($r = 1R_G$)	Outer Boundary (box)	
			upstream, downstream	other
ρ		fixed, $550 \text{ amu}/\text{cm}^3$	fixed	float
p		fixed, 0.115 nPa	fixed	float
p_e		fixed, 0.01 nPa	fixed	float
\mathbf{V}		$\mathbf{V} \perp \mathbf{B}$	fixed	float
\mathbf{B}	dipole		fixed	float

along the parallel direction ($\mathbf{u}_{\text{perp}2} = \mathbf{u}_{\text{perp}1}$, where subscripts “1” and “2” denote the physical and ghost cell faces respectively) at the surface boundary. Then the surface boundary is turned off, and the magnetic diffusion term with resistivity as well as the Hall term are calculated in the full domain with an implicit scheme and added to \mathbf{B} . The inner boundary of the Hall region is set slightly away from the surface at $r = 1.05R_G$ to avoid numerical issues. The core boundary ($r = 0.5R_G$) for \mathbf{B} is always set to a fixed dipole during the flyby. After everything is updated, we move on to the next timestep and repeat the above process. The boundary conditions are summarized in Table 2.2.

It has been suggested from both gravity [Anderson et al., 1996] and magnetometer [Schubert et al., 1996] measurements that Ganymede’s interior is most likely composed of a metallic core of radius $0.15 - 0.5R_G$ that sustains the moon’s internal magnetic field and a silicate mantle enclosed by an ice shell. This implies that the interior of Ganymede and its ionosphere together with the ambient space plasma have different electrical conductivity. We therefore set an ad hoc resistivity profile in the model shown in Figure 2.5 as a function of radial distance r (assuming spherical symmetry in θ and ϕ). To include the effect of the moon’s interior, the innermost simulation boundary is placed at $r = 0.5R_G$, i.e. the maximum of inferred core radius. Between the core boundary ($0.5R_G$) and the moon’s surface ($1R_G$) is the insulating rocky mantle whose electrical conductivity is extremely low. Note that there

is no plasma flowing in this region so the model solves only a diffusion-like equation $\partial \mathbf{B} / \partial t = -\nabla \times (\eta \nabla \times \mathbf{B})$ for the magnetic field. Also as a numerical approximation to a conducting core surface, we set the resistivity to zero at the physical cells next to the core boundary so that the magnetic field does not change and remains equal to the dipole field value. Using the constraints from geophysical measurements [Anderson et al., 1996, Schubert et al., 1996], we apply a simplified, spherically symmetric resistivity profile from the core boundary at $r = 0.5R_G$ to a radius slightly above the surface $r = 1.05R_G$. A surface resistivity $\sim 4 \times 10^5 \Omega \cdot m$ (within the range of the magnetic diffusivity estimation by [Duling et al., 2014]) is used in our simulations, and the overall profile is similar to [Jia et al., 2009]. The resistivity profile set from $r = 1R_G$ to $r = 1.05R_G$ couples the magnetic field inside and outside the moon in our numerical approach and serve as the simplest treatment of ionosphere conductivities. The flowing plasma outside the ionosphere ($r > 1.05R_G$) is regarded as infinitely conducting, so the background resistivity is set to zero. The extension of the resistive region above the moon’s surface is necessary to properly couple the two regions.

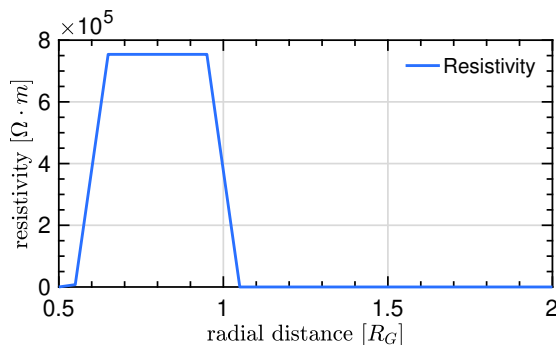


Figure 2.5: Radial profile of the spherical-symmetric resistivity used in the model. Values in each cell is interpolated linearly from specified points. The resistivity is set to 0 for region $r > 1.05R_G$.

Table 2.3: Grid size and particle number of MHD-EPIC models.

	Hall MHD, Spherical Coords.	PIC, Cartesian Coords.	
Grid 1, Zhou+ 2019	0.1 d_i , 6 M cells	0.2 d_i , 0.6 M cells	0.15 billion particles
Grid 2, Zhou+ 2020	0.05 d_i , 27 M cells	0.1 d_i , 2.5 M cells	0.6 billion particles
Grid 3, Zhou+ 2020	0.05 d_i , 27 M cells	0.05 d_i , 20 M cells	4.8 billion particles
$m_i/m_e = 100$, $d_i/d_e = 10$			

2.5.3 Coupling Procedure

We used a stretched spherical grid with levels of adaptive mesh refinement in the whole computational domain for MHD, and a uniform Cartesian grid for PIC. The grid size and particle number information for the models are listed in Table 2.3, where Grid 1 is used in Chapter III and Grid 2 and 3 are used in Chapter IV. Interpolation and communication of variables between the two models is handled by the coupler. In time-accurate simulations, BATS-R-US and iPIC3D advance with different timesteps individually, and they exchange information every 0.02s. The PIC model overwrites the MHD solution in the overlapped region, and obtains time-dependent boundary conditions from the MHD solution. Figures 3.2 and 3.3 show the PIC box embedded inside the MHD domain.

Computationally, we go through the steps as follows. First we obtain a quasi-steady state solution by running BATS-R-US in local timestep mode for 30,000 steps in ideal MHD and 20,000 steps in Hall MHD in the full computational domain. Then we continue the simulation in time-accurate mode coupled with iPIC3D covering a box region at the upstream magnetopause. The MHD-EPIC model runs for 20 minutes in physical time and the outputs are saved every second. Taking advantage of a semi-implicit scheme for solving the magnetic induction equation, we are allowed to use time steps that are not limited by whistler waves and the large resistivity in the subsurface ocean and mantle regions. The mixture of explicit-implicit timestepping enables us to construct a robust model with affordable computational cost.

CHAPTER III

Validation and Inferences

Ganymede’s global magnetosphere MHD-EPIC model has been improved over the years. This chapter presents the validation of the model with various observations and demonstrate the PIC results in detail [Zhou et al., 2019]. We use a stretched spherical grid in GphiO coordinate system with adaptive mesh refinement (AMR) up to 2 levels near Ganymede’s magnetosphere and Alfvén wing structures, enabling high resolution in r about $0.02R_G$ (~ 50 km), θ of 0.7° and ϕ of 1.4° near the moon’s surface. In order to include Ganymede’s Alfvén wing structure as much as possible and avoid the unphysical wave reflection issue at the outer boundaries, we set a large simulation domain of a cube centered at the moon with edge length $l = 200R_G$ cut out of the spherical grid, as shown in Figure 3.1a. The grid near the inner boundary from $r = 0.5R_G$ to $r = 5R_G$ is shown in Figure 3.1b.

In Cartesian coordinates, we set a PIC box at the upstream magnetopause between $-2.5R_G \leq x \leq -1.125R_G$, $-2R_G \leq y \leq 2R_G$, $-2.2R_G \leq z \leq 2.2R_G$ with grid resolution $1/32R_G$ ($\sim 0.2d_i$, or $2d_e$). We initialize the PIC code with 216 ions and 216 electrons respectively in each cell with Maxwellian distribution based on the MHD state at the beginning of the time-dependent simulation. Then we run the coupled model, and allow macro-particles to be lost at the boundary or enter the

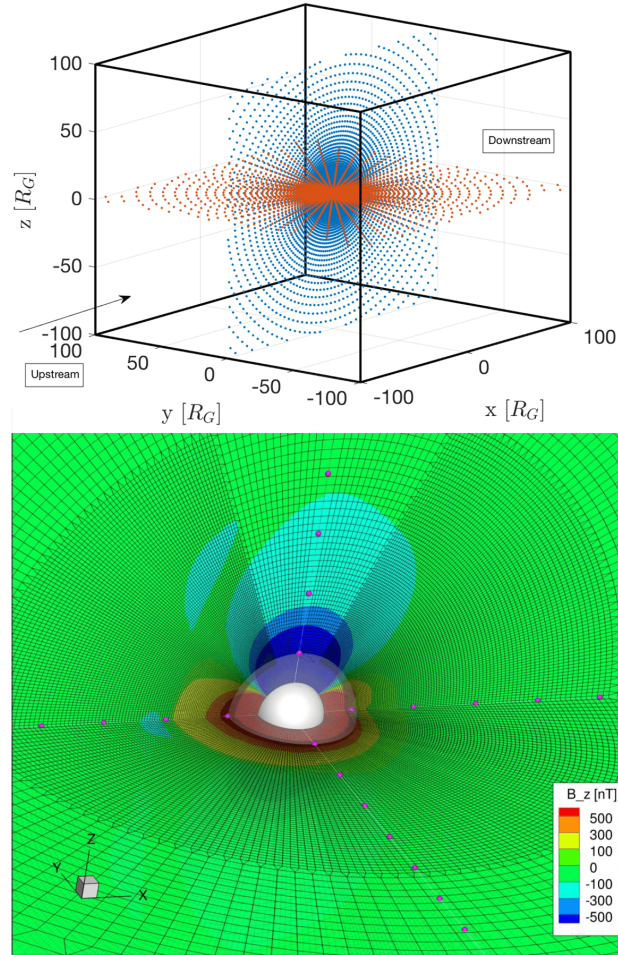


Figure 3.1: On the top shows the whole simulation domain is a cube with edge length $200R_G$ centered around Ganymede cut out of a stretched spherical grid. On the bottom is the grid structure near Ganymede shown by the $y = 0$ and $z = 0$ cuts. The inner boundary is represented by a sphere with radius $0.5R_G$. The distance between neighboring purple balls is $1R_G$. Adaptive mesh refinement is applied up to 2 levels near and within the magnetosphere. The color represents B_z strengths in the two cut planes.

PIC domain as the system evolves.

First we show the general structure of magnetopause boundaries and Alfvén wing during the two upstream crossing Galileo flybys. Then the simulated quasi-steady state magnetic fields in all six close encounter flybys are validated with magnetometer data. The plasma velocity is compared against the G2 flyby PLS data with different interpretations of ion mass-to-charge ratio. After that, we focus on the coupled PIC region which covers most of the upstream magnetosphere, and demonstrate the time-

varying magnetic field, selected plasma moments, ion phase space distributions and aurora estimation from particle acceleration out of the reconnection sites.

3.1 Magnetic Field Comparison for the 6 Flybys

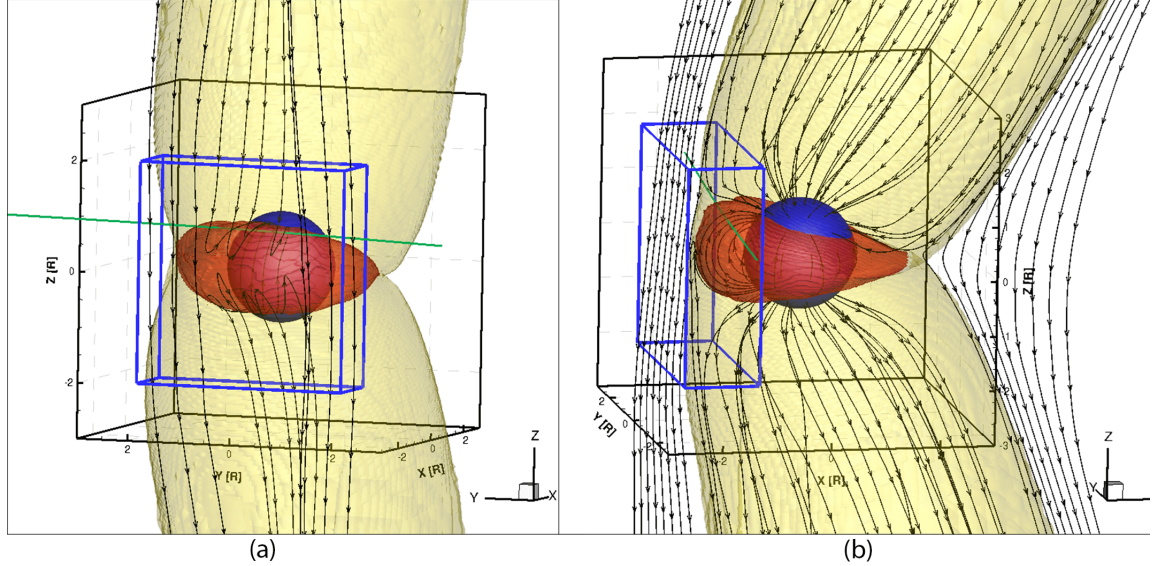


Figure 3.2: (a) 3D front view of Ganymede’s magnetopause from G8 flyby simulation. The dark blue sphere represents the moon’s surface at $r = 1R_G$, the red surface represents the open-closed field line boundary, and the yellow surface displays the Alfvén wing that encloses Ganymede’s magnetosphere. Field lines on the upstream passing through the $x = -1.95R_G$, $z = 0R_G$ line are shown with black arrow lines. The PIC box at the upstream of magneopause is indicated with blue lines. (b) 3D anti-Jovian side view of Ganymede’s magnetopause from G8 flyby simulation. Field lines are traced along $y = 0R_G$, $z = 0R_G$ line at the upstream and tail region, and also over the surface of the moon. The Galileo trajectory is plotted in green.

The most notable feature in sub-Alfvénic plasma interaction is the Alfvén wing. The global structures including the Alfvén wing and magnetopause near Ganymede are shown in Figure 3.2 for the G8 flyby ($\sim 180^\circ$ upstream magnetic field clock angle with respect to the \mathbf{z} axis) and Figure 3.3 for the G28 flyby ($\sim 135^\circ$ clock angle). Since the intrinsic dipole is only 4° off from $-\mathbf{z}$ axis [Kivelson et al., 2002], these two upstream flybys can represent two typical cases with nearly anti-parallel reconnection and strong guide field reconnection, respectively. In the 3D visualization plots, the red surface represents the open-closed field line boundary, and the yellow surface

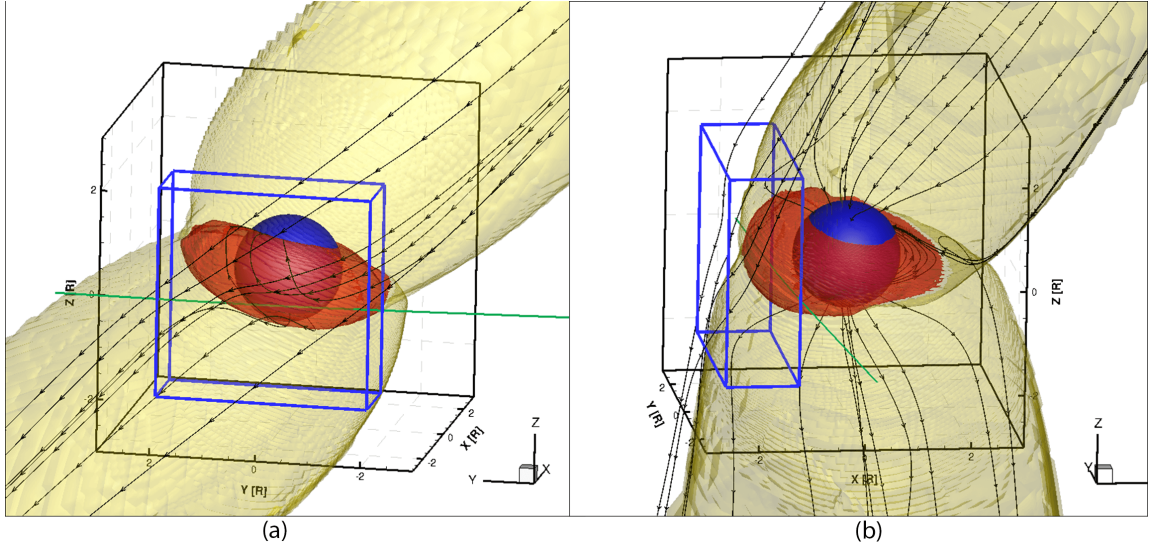


Figure 3.3: a) 3D front view of Ganymede's magnetopause from G28 flyby simulation. Field lines on the upstream are traced along $x = -1.8R_G$, $z = 0R_G$ line. (b) 3D anti-Jovian side view of Ganymede's magnetopause from G28 flyby simulation. Field lines are traced along $y = 0R_G$, $z = 0R_G$ line at the upstream and tail region, and also over the surface of the moon. The Galileo trajectory is plotted in green.

displays the boundary of field lines with one end connected to the moon and the other end connected to the outer boundaries. During the G8 flyby, the perturbed field lines within the yellow surface form the nearly symmetric Alfvén wing structure over the poles, while for the G28 flyby the southern Alfvén wing is tilted towards the sub-Jovian side and the northern Alfvén wing is tilted towards the anti-Jovian side. At the upstream magnetopause, the topology of magnetic field lines are quite different with/without the guide field (approximately B_y component).

With Hall physics included in our simulation, there is no true steady state in the global dynamic system, but we can obtain an approximately steady state solution. First we validate the quasi-steady state Hall MHD solution for all six Galileo flybys. The magnetic field comparisons are shown in Figure 3.4. The Galileo trajectories for G1, G2, G7 and G29 flybys go through the Alfvén wing near the north pole region, while the trajectories for G8 and G28 flybys go across the upstream low latitude magnetosphere. Note that PIC is not turned on in the steady state convergence

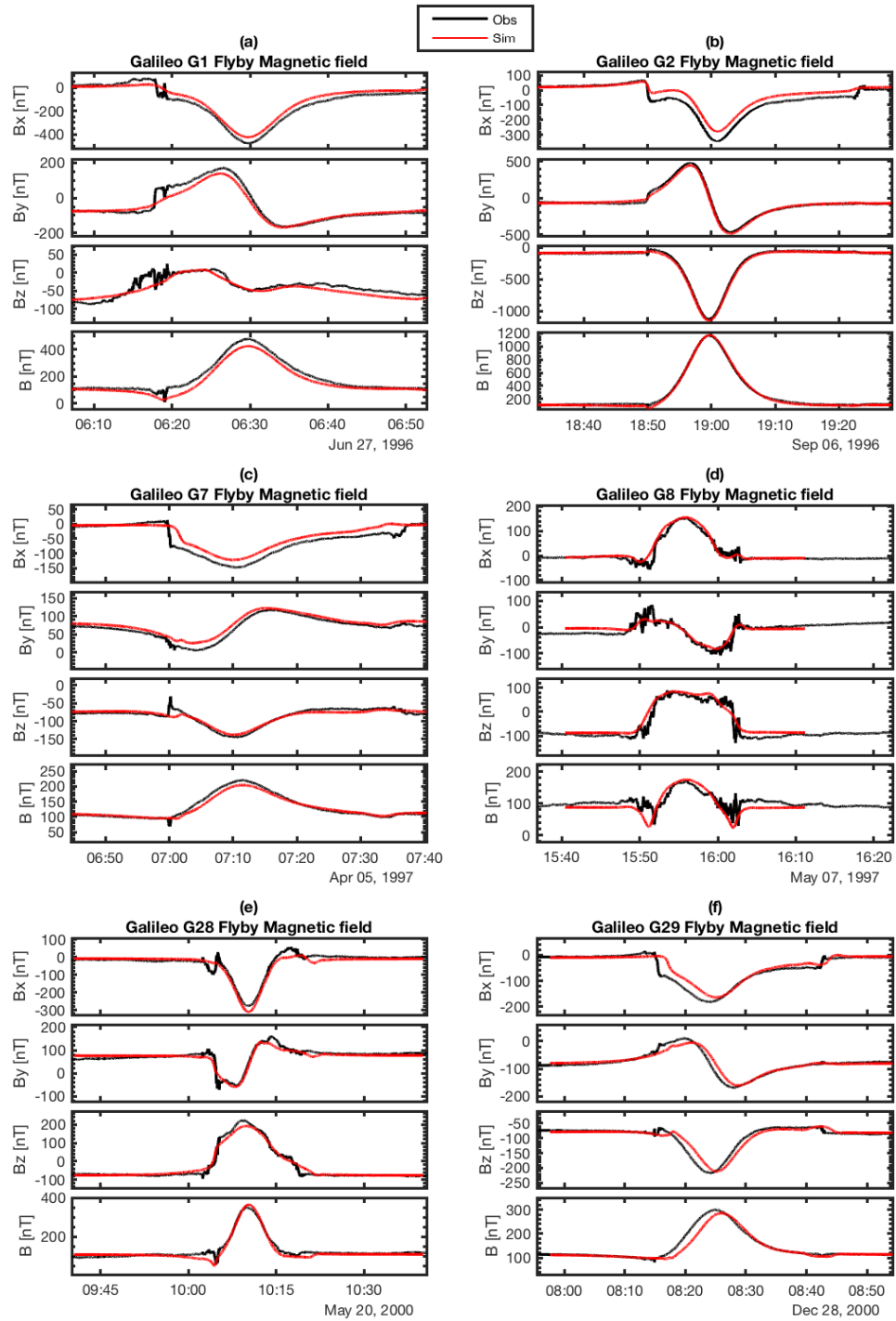


Figure 3.4: Magnetic field comparison with Galileo observation for quasi-steady state simulations of all six flybys. In each subplot, the Galileo magnetometer measurements are shown as black solid lines and the simulation results are plotted as red solid lines.

runs. Due to the fact that the conductance of the mantle, subsurface ocean and ionosphere are not well constrained observationally, our choice of conductivity profile is simply based on estimation of reasonable range and comparisons with observed magnetic fields. Our simulation tests show that the magnetic field near the moon is very sensitive to the profile of conductivity as well as the grid resolution and structure. Taking conductivity into consideration in such a small magnetosphere can significantly modify its size and shape. With the inclusion of the resistive body, the overall size of the magnetosphere matches better with observations such that no stretching or shifting of the Galileo trajectory is needed, in contrast with [Dorelli et al., 2015] and [Tóth et al., 2016]. The validation here shows that our Hall MHD model can faithfully reproduce the magnetic field observed by Galileo under different upstream conditions. For the four Alfvén wing flybys, G1, G2, G7 and G29, the differences in the magnetic field data comparisons indicate that the location or size of our simulated Alfvén wing is slightly different from observation. Especially during the inbound crossings of the G7 and G29 flybys, the locations of the Alfvén wing are shifted inward, which is similar to the resistive MHD simulation results by Jia et al. [2009]. The agreement in the two upstream magnetopause crossings shows that the position of open-closed field line boundary is well captured in Hall MHD, which gives a nice starting point of incorporating the PIC box region at the upstream magnetosphere.

3.2 Plasma Velocity Measurements

Plasma measurements from the Galileo Plasma Subsystem (PLS) have been published for the G2 flyby [Frank et al., 1997b]. Collinson et al. [2018] recently re-analyzed the PLS data to obtain a new set of plasma moments for the G1 and G2

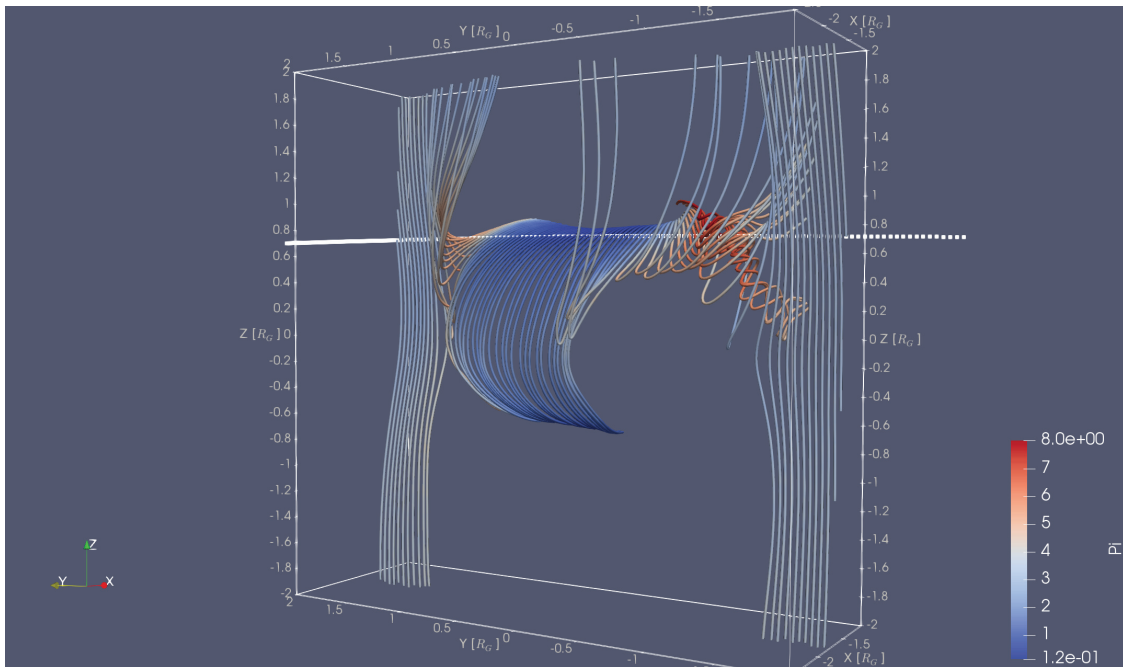
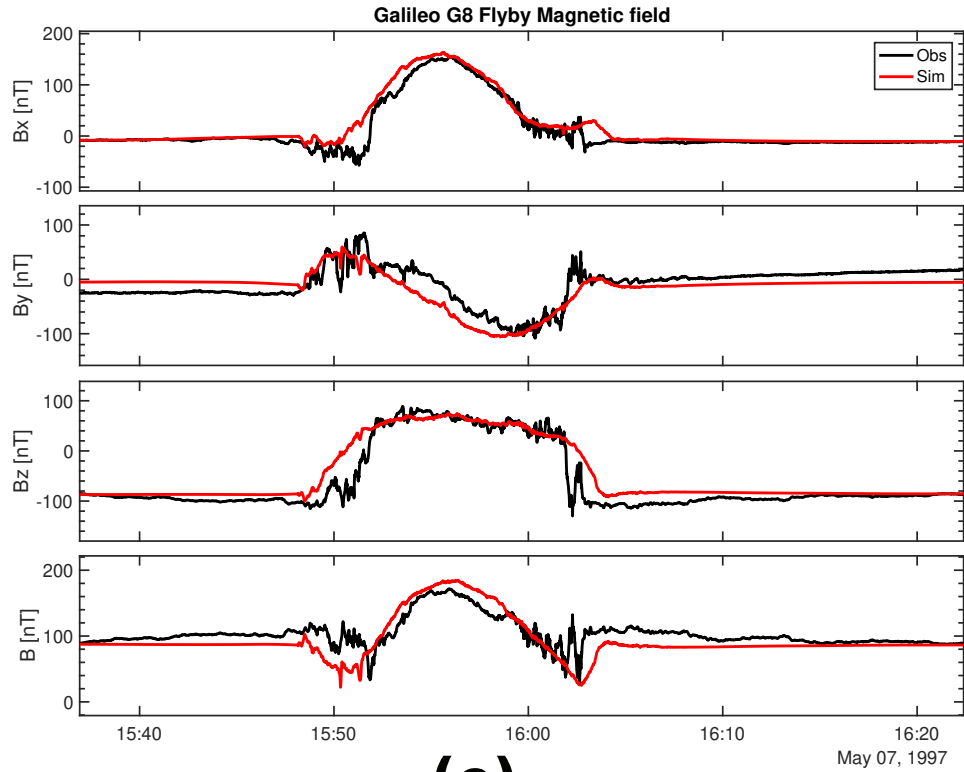
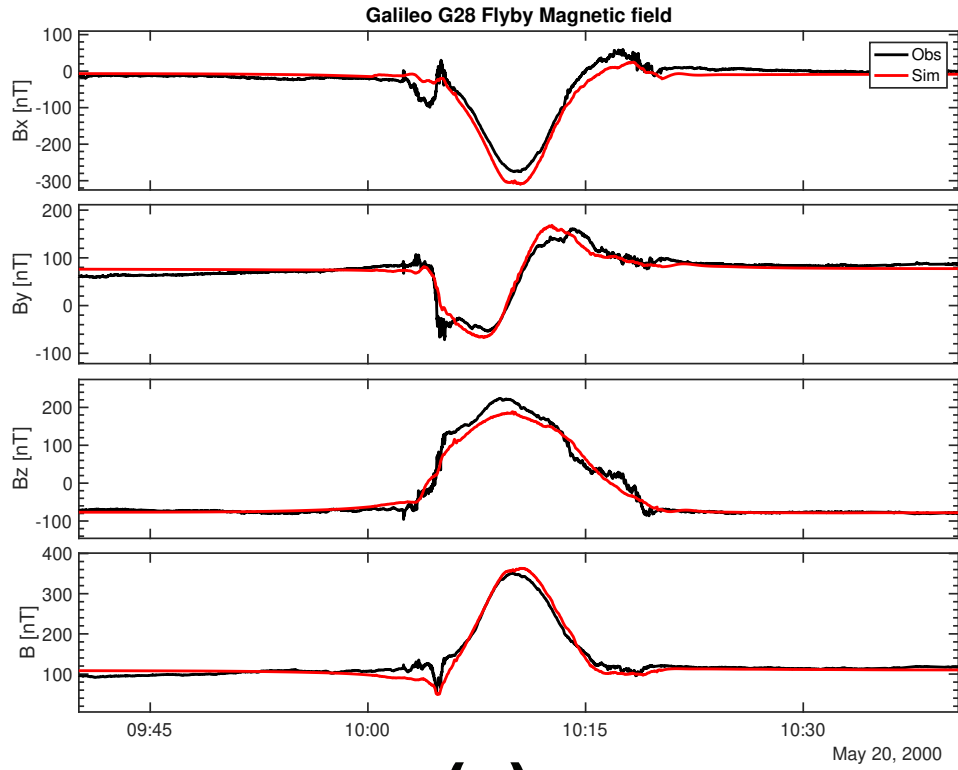
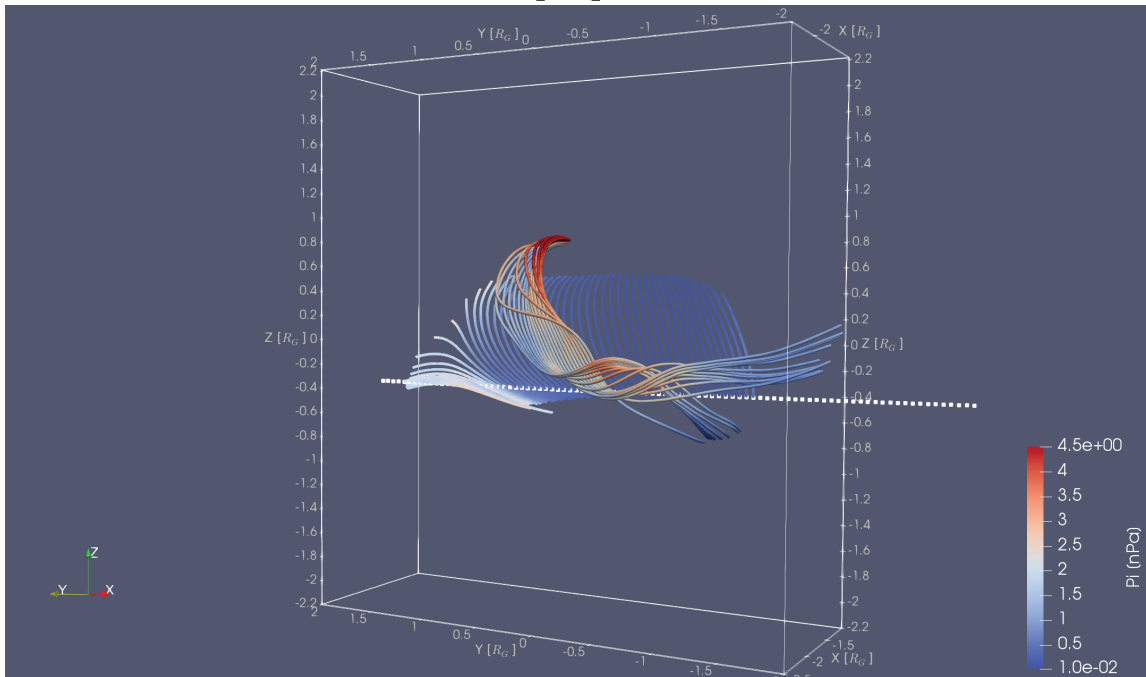


Figure 3.5: (a) Magnetic field extracted along the Galileo trajectory from 20 min time-accurate MHD-EPIC simulation of G8 flyby inside the PIC box region. The black line is the observations and the red line is the simulation results. Signatures of flux rope can be identified during the inbound crossing of magnetopause between 15 : 51 ~ 15 : 52 UT, where sharp rotations of magnetic field, especially in the B_y component, are present. (b) 3D topology of the magnetic field lines traced along the Galileo trajectory (shown by the white dots) inside the upstream PIC box at 15:51 UT. The colors of the field lines show the ion pressure.



(a)



(b)

Figure 3.6: (a) Magnetic field extracted along the Galileo trajectory from 20 min time-accurate MHD-EPIC simulation of G28 flyby inside the PIC box region. The black line is the observations and the red line is the simulation results. (b) 3D topology of the magnetic field lines traced along the Galileo trajectory (shown by the white dots) inside the upstream PIC box. The colors of the field lines show the ion pressure.

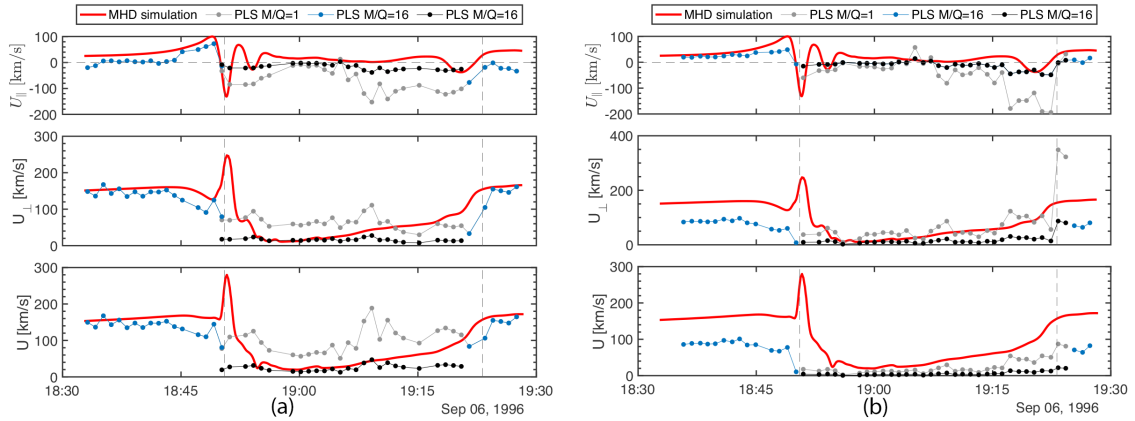


Figure 3.7: Comparisons of parallel and perpendicular components of the plasma bulk flow velocities relative to the magnetic field for the G2 flyby. The PLS plasma data are obtained from (a) [Frank et al., 1997b] and (b) [Collinson et al., 2018], while the simulation data presented in red lines are from steady state Hall MHD. In each panel, the blue dots represent bulk flows derived from the PLS measurements for heavy ions with mass-per-charge (M/Q) = 16. The gray dots are the calculated moments assuming $M/Q = 1$, and the black dots represent bulk flows assuming $M/Q = 16$. The two vertical dashed lines mark the inbound and the outbound magnetopause crossings identified from the magnetic field data, respectively.

flybys. Here we compare the plasma velocities for the G2 flyby from our Hall-MHD simulation with the PLS moments obtained by both Frank et al. [1997b] (Figure 3.7a) and Collinson et al. [2018]. (Figure 3.7b). Following the approach used in the same kind of comparison in [Jia et al., 2009], we have plotted two velocity profiles for the PLS data corresponding to two different assumptions of the ion mass-per-charge (M/Q): blue and black dots for heavy ions of $M/Q=16$, and gray dots for light ions of $M/Q=1$. The velocity components shown in Figure 3.7 have been decomposed into the parallel and perpendicular directions with respect to the magnetic field.

First of all, outside the magnetosphere (the magnetopause crossings are marked by the vertical dashed lines), the flow velocities in our model agree very well with the PLS velocity data derived by Frank et al. [1997b], but deviate from the Collinson et al. [2018] study that gave much reduced plasma speeds. It is important to point out that our simulation assumes a nominal flow speed of ~ 150 km/s for the unperturbed Jovian plasma at the upstream boundary.

Secondly, inside the magnetosphere, due to lack of direct measurement of the ion mass-per-charge, there is ambiguity in deriving the plasma moments using the raw PLS measurements. Therefore, the previously published work typically provide two sets of moments with different assumptions of ion mass-per-charge, both of which are shown in Figure 9 for comparison. Overall, our Hall-MHD model prediction, especially the perpendicular velocity component that is governed by the $\mathbf{E} \times \mathbf{B}$ drift, agrees better with the derived PLS velocity data assuming heavy ions. This result is consistent with the finding obtained in the previous MHD modeling of [Jia et al. \[2009\]](#), both in support of the suggestion first made by [Vasyliūnas and Eviatar \[2000\]](#) that the plasma population detected by the Galileo PLS inside Ganymedes polar cap during the G2 flyby is composed mainly of heavy ions.

Thirdly, near the inbound magnetopause crossing, our Hall MHD model predicts higher plasma velocities than seen in both [Frank et al. \[1997b\]](#) and [Collinson et al. \[2018\]](#) results. These high speed flows in our model are associated with reconnection jets produced by magnetopause reconnection near the flanks. Typically we have observed stronger outflow velocities in Hall MHD than in MHD-EPIC, but in both models the long X-line extends along Ganymede’s magnetopause and create outflow jets that is not revealed by the PLS data. Therefore, the discrepancy between the model and data may be due to lack of high fidelity measurements, but clearly further studies and new observations are needed to confirm our model prediction.

3.3 Upstream Dynamics

Generally, sources of magnetic field variations during the magnetopause crossing include the Hall effect, FTEs, waves and instabilities. It is hard to predict theoretically the contribution from each part in different regions of a nonlinear system.

Similarly to previous work of G8 flyby simulation [Tóth et al., 2016], by extracting the magnetic field from the time-accurate runs along the Galileo flyby trajectories, we can examine reconnection-related features in the magnetic field especially near the magnetopause. From this point forward, we will only discuss results from time-accurate MHD-EPIC simulations of the G8 and G28 flybys. The magnetic field comparisons from the MHD-EPIC of the G8 and G28 flyby simulation are shown in Figure 3.5a and Figure 3.6a. A 20-minute simulation is sufficient to cover the inbound and outbound crossings for both G8 and G28 flybys in a continuous manner. Since there is no time variation in the driving conditions and flux transfer events are generated spontaneously, we have the freedom to shift the start time of the simulation relative to observations. The data outside the simulation time interval are compared with outputs from the same snapshot of the starting and ending points. Figure 3.5a shows the magnetic field comparison for the time-accurate G8 flyby MHD-EPIC simulation, and Figure 3.5b shows the corresponding 3D field line tracing along Galileo’s trajectory in one snapshot inside the PIC region at the upstream magnetopause. The oscillations in the three magnetic field components, especially B_y during the G8 inbound crossing, are clear signatures of a transient FTE formed at the upstream magnetopause, while the overall magnetospheric structure is similar to that in the quasi-steady state. During the G8 flyby, Galileo was moving in the $+y$ direction (right to left in the Figure) from anti-Jovian side to sub-Jovian side. We identify flux ropes during the magnetopause inbound crossing with enhanced ion pressure at the center, and show field lines that just get reconnected from X-lines at the outbound crossing. In the observed magnetic field, there are spikes during the outbound crossing near 16 : 03 UT, which is potentially a transient signature associated with a flux rope that is not entirely captured by the model at the same physical

time. By going through all the snapshots in the 20-min simulation, we do find flux ropes forming near the outbound magnetopause locations. In a complicated nonlinear system where no true steady state exists, it is not possible to reproduce exactly the same observation given the available observational constraints and the nature of flux rope formation. However, the fact that the overall trend and fluctuation as shown by time-accurate magnetic field comparisons and the power spectra (Figure 14 in our previous paper [Tóth et al., 2016] agree well with observations suggests that the MHD-EPIC model provides a very good description of the magnetic field structure at Ganymede.

Figure 3.6 displays the scenario for the G28 flyby in the same format as in Figure 3.5. In this case with near unity guide field ($B_z \sim B_y$), the magnetic field tends to be more steady, even though one can still argue that there are small flux rope signatures during the magnetopause crossing. Actually, we do find flux ropes on the upstream magnetopause as shown in Figure 3.6b; however, these appear mostly in regions away from the G28 flyby trajectory, so we cannot easily see them in the 1D synthetic magnetometer data.

Our model provides many more quantities beyond ideal MHD inside the PIC region. We plot magnetic field, pressure and velocity from PIC outputs on the $y = 0$ meridional cut plane in Figure 3.8 at $t = 420s$. A pattern of quadrupolar Hall magnetic field centered at $(x, z) = (-1.9R_G, 0.05R_G)$ is clearly shown in Figure 3.8a, with a small electron diffusion region (Figure 3.8e) and a relatively large ion diffusion region (Figure 3.8f). There is also a fast electron drift in the out-of-plane (approximately y) direction across the magnetopause (Figure 3.8d), peaked at the reconnection sites due to curvature of \mathbf{B} and the Hall effect. The deflection of electrons and ions in opposite directions at the magnetopause causes a Chapman-Ferraro

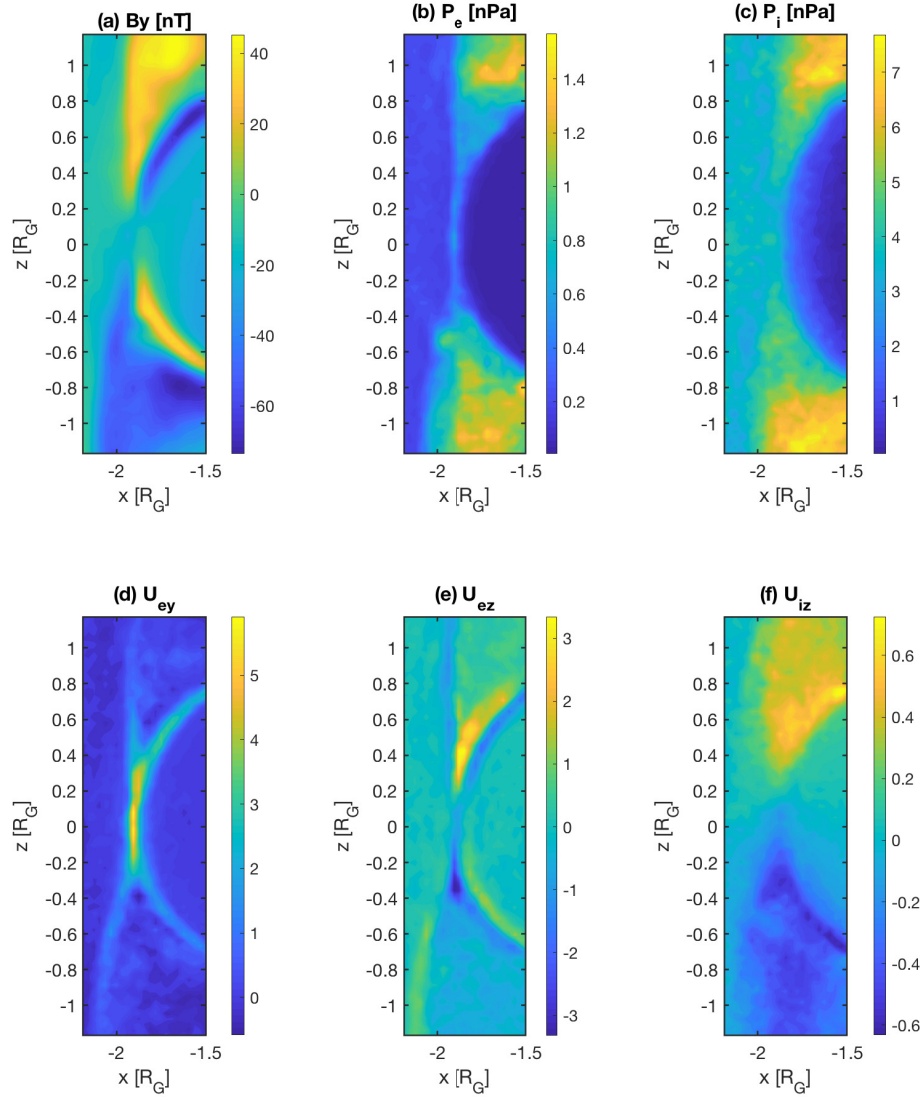


Figure 3.8: Contour plots in the meridional plane $y = 0$ containing the upstream magnetopause from time-accurate G8 simulation at $t = 420s$. The color contours represent (a) B_y (Hall magnetic field); (b) P_e (Electron pressure); (c) P_i (Ion pressure); (d) U_{ey} (Electron velocity in y-direction); (e) U_{ez} (Electron velocity in z-direction); (f) U_{iz} (Ion velocity in z-direction). All the velocities are normalized to the upstream Alfvén speed of 253 km/s.

like current pointing from Jovian side to anti-Jovian side ($+y$ to $-y$). Corresponding in-plane electric field E_x is shown in the left panel of Figure 3.9. The ambipolar structure of E_x arises from the electron-ion charge separation at the magnetopause, which is both included in the Hall MHD and PIC model. Ions are being accelerated by this in-plane electric field, entering the reconnection exhaust across the separatrices.

3.4 Ion Velocity Distribution

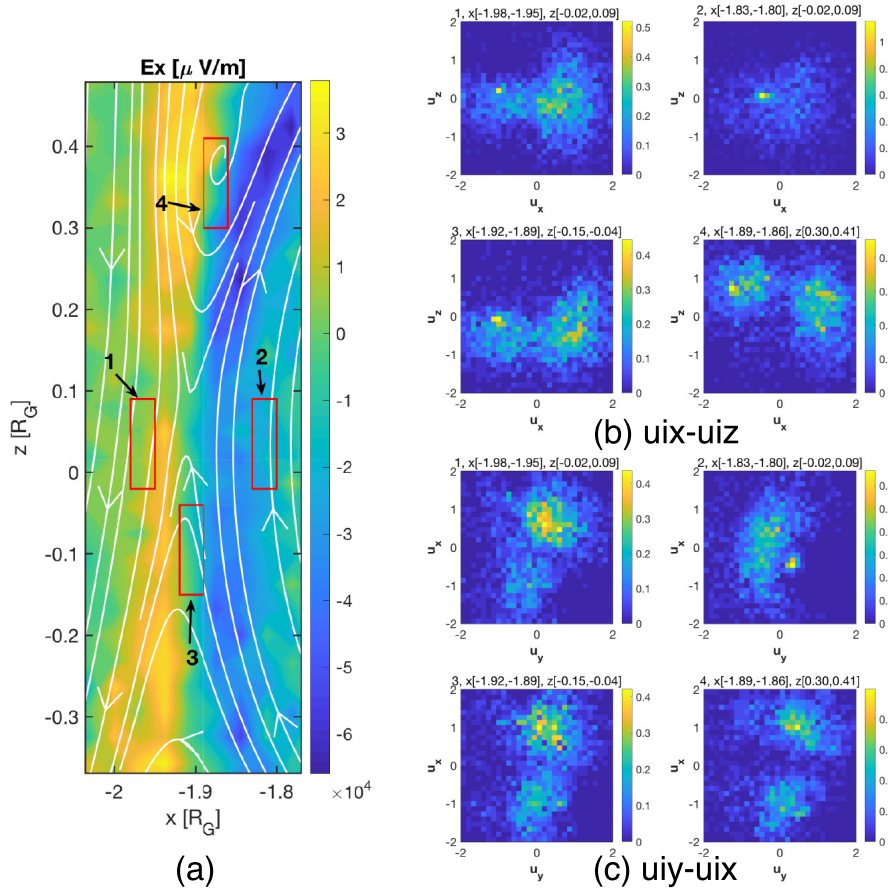


Figure 3.9: Ion velocity distributions in normalized density near the upstream reconnection site from G8 flyby simulation at $t = 420s$. The colored contours on the left show the electric field E_x overlaid with magnetic field lines in white, with four sample boxes (1 upstream inflow region, 2 magnetosphere inflow region, 3,4 outflow region) showing sampling locations used to extract the ion phase space distributions in $u_{ix}-u_{iz}$ (four upper right panels) and $u_{iy}-u_{ix}$ (four bottom right panels). The boxes extend in y direction from $-0.08R_G$ to $0.08R_G$. All the velocities are normalized to the upstream Alfvén speed.

With an embedded PIC model, we can obtain detailed information about the

behavior of the electrons and ions directly by studying their velocity distributions near the X-line. Due to the quasi-symmetry at the nose of the magnetopause, the generally used LMN boundary-normal coordinate system in reconnection studies is aligned with the Cartesian coordinates, where the normal direction N is $-x$ direction, L is z direction, and out-of-plane direction M is y direction, so it is convenient to plot model results in the simulation coordinates. Clear signatures of ion motion near the separatrices are nongyrotropic distributions (e.g. [DeCoster and Frank, 1979, Ashour-Abdalla et al., 1993, Frank L. A. and Kivelson, 1994] for Earth tail reconnection, [Burch and Phan, 2016] for Earth dayside magnetopause reconnection). Based on the $y = 0$ plane Hall electric field E_x contours on the left of Figure 3.9, we select four box regions of the same size to extract the ion distribution functions in the inflow and outflow regions near the X-line in the G8 flyby simulation, where $E_x \sim E_N$, and $B_z \sim B_L$. In the Cartesian coordinates, approximately, u_x is the inflow velocity, u_z is the outflow velocity, and u_y is the out-of-plane velocity. The ion gyroradius r_i in the simulation has a maximum value of about $0.2R_G$ with an average of $0.06R_G$, thus the selected box region has a width of $\sim 0.5r_i$ in the x direction, and $\sim 1.8r_i$ in the z direction. The right-hand side panels of Figure 3.9 show the ion distributions in velocity phase space for the inflow and outflow regions, respectively, where the histograms are normalized by probability distribution separately. In the u_{ix} - u_{iz} plots on the upper right, there are single peaks in the two inflow regions which represent the isotropic inflow streams, and double peaks in the outflow regions that represent the counterstreaming behavior from ion entry with anisotropy in parallel and perpendicular directions. Such counterstreaming behavior has also been observed in earlier local PIC and hybrid simulations of Earth-like reconnection [Drake et al., 2009, Hoshino et al., 1998, Arzner and Scholer, 2001], and by Cluster

spacecraft 1 in the Earth’s magnetosheath [Phan et al., 2007]. The $u_{iy}-u_{iz}$ plots are shown on bottom right. Moving from upstream into the magnetosphere across the X line (Region 1 to Region 2), we see that the two beams in the upstream merge and a crescent-shaped distribution forms on the magnetospheric side, which corresponds to the ion meandering motion in the diffusion region near the reconnection site [Lapenta et al., 2017, Bessho et al., 2016, Egedal et al., 2016]. In the two out-flow regions, there is a similar butterfly shape distribution in this $u_{iy}-u_{ix}$ cut, which represents a drift in the y direction and anisotropy in temperature between the two perpendicular directions. The non-gyrotropic anisotropic distribution functions show that the embedded kinetic model can simulate the magnetic reconnection from first principle.

In the case of the G28 flyby simulation, we also check the ion distribution near the reconnection site (not shown here). There is evidence of ion crescent distribution in the perpendicular plane at the magnetospheric side of inflow region, even in a strong guide field case.

3.5 Energy Flux Density and Auroral Emission

The particle information obtained from the PIC model allows us to calculate the energy flux densities in a self-consistent manner. Observations by the Hubble Space Telescope (HST) indicate that Ganymede possesses auroral emissions [Feldman et al., 2000]. It is shown that the auroral oval lies very close to the polar cap boundary that separates the open and closed magnetic field lines [McGrath et al., 2013]. On the upstream side, the open-closed field line boundary maps to the magnetopause where reconnection is taking place. It is, therefore, reasonable to expect that reconnection-produced energetic particles may contribute to the generation of aurora. Here we

first calculate the energy flux densities using the particle information output from the PIC model to assess the contribution of magnetopause reconnection to the aurora emissions. In our simulation, the inner edge of the PIC box is set at $x = -1.125R_G$. Although the PIC region set at the upstream magnetopause does not include the moon's surface, we can calculate the energy flux densities inside the PIC region and map them to the surface based on the magnetic field topology resolved by the global model under the assumption of flux conservation. Specifically, the procedure goes as follows:

1. Read particle information from PIC and magnetic field information from MHD.
2. For each particle in the selected region, interpolate the magnetic field at the particle locations and calculate the pitch angles.
3. Find the connectivities of magnetic field lines between particle locations and the moon's surface, and then compute the critical angles for the loss cone.
4. Select particles inside the loss cone, and calculate the energy flux densities along the field lines inside the PIC region.
5. Map the fluxes onto the surface along the field lines.

The most important assumptions in this analysis are the conservation of energy and magnetic flux. With the current model and the linear interpolation and integration approaches, the error in magnetic flux conservation is less than 5%, which is sufficient for obtaining an overall distribution of flux densities and comparison with remote observations. The energetic particles inside the loss cone is about 4% of the total number of macro-particles in the sliced PIC region, with an average critical angle of roughly 20° . Besides, because of the size of PIC box, we are restricted to a

finite portion of the surface area which has energetic particle precipitation from the PIC region.

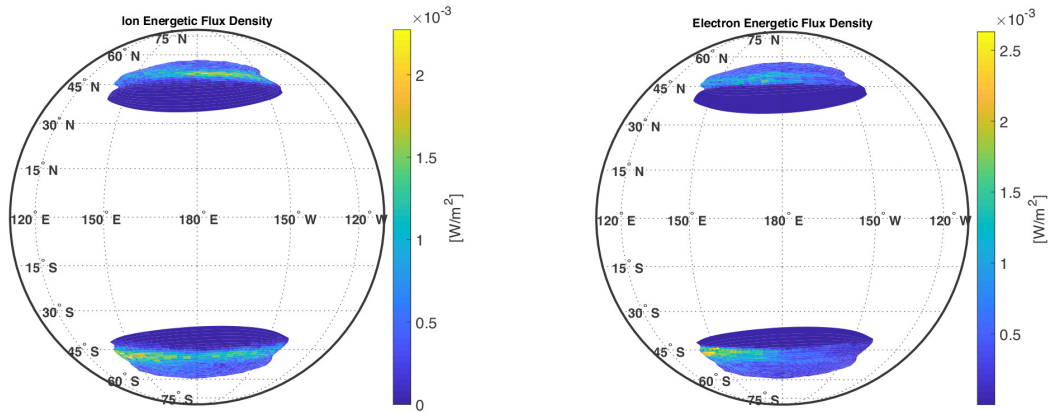


Figure 3.10: Energy flux densities mapped from PIC region onto Ganymede’s surface for (a) ions and (b) electrons, respectively. This is a view from upstream, where $180^\circ E$ points towards $-x$, and $90^\circ E$ points towards $+y$ (Jupiter) in GPhiO system. The mapping regions are limited to mid latitudes between approximately $150^\circ E$ to $150^\circ W$ due to the size of the PIC box.

The mapped energy flux densities for both ions and electrons at $t = 280s$ are shown in Figure 3.10 viewing from upstream. We can clearly see curves of peak flux density on the order of $10^{-7}W/cm^2$ for both species in the northern and southern hemisphere. This is on the same order but slightly higher than a rough average polar cap precipitating energetic flux estimation given by [Frank et al. \[1997a\]](#) for the low-energy electron measurements between 0.5 and 3.0 keV for the G2 flyby, but about one order higher than the G7 flyby estimation assuming an invariant latitude of $\sim 45^\circ$ by [Paranicas et al. \[1999\]](#), probably due to the intensities of low-energy electrons over the polar cap. Both of the previous estimates use raw data from PLS measurements, with extrapolation fit over the energy range. Given the existence of aurora near the cusp region due to the precipitating electrons, concentrated energy fluxes with higher intensities than the surroundings are expected in the cusp. In both hemispheres, the largest flux density lies within approximately $[45^\circ, 55^\circ]$ latitude, which is consistent with Hubble observations of atomic oxygen emission by [McGrath et al. \[2013\]](#). For

the 60° in longitude on the trailing side that is covered by the upstream PIC box, the location of peak energetic fluxes obtained from our model correlates well with the location of the brightest emissions observed by the HST (Figure 3 in [McGrath et al., 2013]). In our model, the energetic flux densities for electrons and ions inside the loss cone are on the same order of magnitude with an obvious asymmetry between Jovian/anti-Jovian ($\pm y$) directions: generally larger for ions in the $-y$ direction and electron in the $+y$ direction. One plausible explanation for the asymmetry is the Hall effect: the separation of electron and ion motions causes a Chapman-Ferraro like current across the magnetopause from left to right ($+y$ to $-y$), thus the mapping of fluxes from the two species from the magnetosphere onto the surface follows the same pattern.

From Figure 3.2, we observe a high correlation of the open-closed field line boundary (shown in red) with the peak emission locations which connect to the “cusp” and upstream reconnection outflow regions. The bright peak line near the moon’s surface can be caused by the magnetic mapping from the magnetosphere to the surface, or the distribution of energetic particles out of reconnection sites. To check which is the effective cause, we replaced the real energetic particle fluxes along the field lines with uniform flux and performed the same mapping. The result shows that even though there is still a single peak curve on the surface, the whole distribution has no sharp gradient like the ones shown in Figure 3.10. This experiment suggests that both the magnetic focusing and the distribution of energetic particles at the origins have influence on the distribution of the energetic flux densities near the surface.

For a brightness estimation, Payan et al. [2015] used a radially-averaged relation (Equation 5 in the paper) between the average electron number density, average collisional excitation rate and molecular oxygen column density. We apply the same

idea in estimating the emission caused by reconnection process in the upstream, considering only the energetic electrons out of the upstream reconnection region. If we map these electrons with ~ 1 keV temperature inside the loss cone to the surface of Ganymede under B/n conservation, we get a surface energetic electron number density $\sim 3\text{cm}^{-3}$. The upper constraint on the polar region molecular oxygen column density is $N(O_2) \sim 4 \times 10^{14}\text{cm}^{-2}$ from Voyager observation [Broadfoot et al., 1981]. The electron pitch angle distribution is expected to become roughly isotropic near the surface where the neutral oxygen density is high, so the length of the path of the electron is about $\sqrt{3} \approx 1.7$ times higher than the length of the field line. In addition, the field lines are inclined, which further increases the integrated column density along the trajectory of the electron, so we use $N' \approx 8 \times 10^{14}\text{cm}^{-2}$ for the effective column density. With an average collisional excitation rate $\sim 5 \times 10^{-8}\text{cm}^3\text{s}^{-1}$ for keV electrons colliding with atomic oxygen at 1356\AA emission wavelength (derived from [Payan et al., 2015]), these reconnection-accelerated electrons contribute to a maximum of ~ 120 R in the northern and southern hemisphere, which corresponds to 40% of the peak emission brightness observed by Feldman et al. [2000].

Eviatar et al. [2001a] investigated the cause of Ganymede's auroral emission and discussed the possible mechanisms for local acceleration required to heat the electrons. In the analog to the terrestrial case, they stressed that the orders of magnitude smaller atmosphere column density at Ganymede than that at Earth leads to the penetration of energetic electrons almost totally without collisions. Thus the secondary electrons shall not play an important role in Ganymede's aurora emission, and we do not include them in the estimation. Our calculation from PIC simulation on the upstream side shows that hot electrons coming from the reconnection site can contribute a certain fraction of the peak emission brightness, which is consistent with

the estimation of continuous aurora by [Eviatar et al. \[2001a\]](#), and tends to favor the suggested local acceleration mechanisms proposed thereafter. A recent Hubble observation analysis by [Molyneux et al. \[2018\]](#) suggests an optically thick O atmosphere besides O_2 , with a ratio of $O/O_2 \sim 10\%$. Since O has a smaller electron impact cross section than that of O_2 , the estimated direct reconnection-contributed emission could only be smaller. Besides, upstream reconnection as shown by our model also produces ion energetic fluxes with the same order of magnitude as electron energetic fluxes. However the ion contribution is negligible due to smaller thermal velocities.

3.6 Unresolved Electron Kinetics

Considering the reduction of the ion-electron mass ratio 100 in the PIC simulation, the electron results must be interpreted with care. For the simulations presented in this chapter, our grid resolution in the PIC domain is $0.2d_i \sim 0.4r_i \sim 2d_e$, which is sufficient to resolve ion-scale physics but not enough to fully resolve electron-scale physics, especially near the electron diffusion region. Indeed, we do not find the crescent shape in the electron phase space distribution near the X-line reconnection sites.

However, even if the grid is coarser than the electron skin depth, the implicit PIC scheme is still intrinsically different from a hybrid scheme, where electrons are represented as fluid while ions are represented as macroparticles:

1. The implicit PIC still retains all the electron kinetic physics even when the simulation doesn't resolve the electron scales. However, in such coarse configurations the dynamics of electrons is described at lower accuracy. That means that the length that we don't resolve become equal to the grid spacing and frequency of wave we don't resolved are compressed to the Nyquist frequency and damp

with a rate that is proportional to the time step. While hybrid and semi-implicit PIC models might operate over the same spatial and time scale, semi-implicit PIC still retains all the kinetic physics, even though at a low accuracy, while the hybrid model can only incorporate a fluid description. In particular, fluid electrons can't interact with waves (e.g. heating, acceleration and instability) making impossible to transfer energies between waves and particles. So they are very different when it comes to modeling capability.

2. Collisionless magnetic reconnection is due to the decoupling of electrons and the magnetic field, and one major driving force is the non-diagonal term in the electron pressure tensor. Neither of these are represented by hybrid codes, both of these are represented by semi-implicit PIC codes, even if the grid is coarser than the electron skin depth.
3. In hybrid codes the electrons are usually described as a massless charge neutralizing fluid. If the Hall term is kept in the induction equation, then the equations support whistler waves, which then requires an implicit solver or some other tricks. So it is not really simpler than the semi-implicit PIC method. If the Hall term is not included, then the hybrid scheme does not even have the Hall physics. So hybrid codes have the same issues as MHD codes with respect to reconnection: they rely on numerical diffusion. Semi-implicit PIC relies on proper physics.
4. We performed a grid convergence study for the GEM reconnection challenge with the implicit PIC (Figures 11 and 12 in [Chen and Tóth, 2019]). As the grid resolution is degraded, some details get lost, but the overall reconnection physics is still properly captured. The coarsest grid has only 1 cell per electron skin

depth, still the reconnection rate is correct, and the solution looks reasonable.

5. Numerically, the time steps and required grid resolution are very similar for the hybrid and implicit PIC model. The implicit PIC has more particles (electrons), but otherwise the cost is quite comparable. So the argument that hybrid codes are much cheaper than semi-implicit PIC is not true.

In the MHD-EPIC application to Earth's dayside magnetopause with an extra scaling factor included [Chen et al., 2017], the crescent distribution for electrons can be identified with a grid resolution of $0.5d_e$ which is 4 times finer than our Ganymede's case. Based on our previous scaling studies [Tóth et al., 2017], we are confident that the global ion-scale dynamics is well captured by the MHD-EPIC model, while the modeling of electron-scale physics can be improved by further increasing the grid resolution. An improved MHD-EPIC model which resolves both the ion and electron scale physics is presented in Chapter IV.

CHAPTER IV

Reconnection-driven Dynamics

The two most important processes in the magnetosphere are magnetic reconnection and wave-particle interaction. The first explains the conversion of magnetic field energy into kinetic energy, the second explains the conversion of energy between charged particles and electromagnetic waves.

This chapter presents the reconnection-driven dynamics near Ganymede's upstream magnetopause. We have chosen to use a set of simulation parameters (including both the external and internal boundary conditions) that correspond to those of the Galileo G8 flyby, during which the spacecraft passed through the low-latitude, upstream magnetopause where reconnection is expected to be active. We set the upstream ion number density $n_i = 4 \text{ cm}^{-3}$, plasma velocity $V_x = 140 \text{ km/s}$, magnetic field $\mathbf{B} = [-10, -6, -86] \text{ nT}$, and thermal pressure $P_i = 3.6 \text{ nPa}$, $P_e = 0.2 \text{ nPa}$. Both the Hall MHD and MHD-EPIC simulations have been run for a total duration of 20 minutes, which is several times the typical time it takes the ambient flow to pass the magnetosphere. The time-accurate Hall MHD simulation starts from the quasi-steady state solution and the time-accurate MHD-EPIC simulations start from $t = 300\text{s}$ after the Hall MHD run. The time lag is chosen such that the solution has been fully settled into time-dependent Hall MHD that does not drastically change

with time. The computational domain is defined in the GPhiO coordinate system, where x is along the flow direction, y is along the Ganymede-Jupiter vector with positive direction pointing towards Jupiter, and z is along the spin axis. Compared with the simulations presented in Chapter III, we have further increased the grid resolution for both fluid and kinetic models. We doubled the resolution inside the magnetosphere to reach an average of $1/120R_G \sim 0.05d_i$ in the radial direction, 0.7° in the azimuthal direction and 0.35° in the polar direction for the stretched spherical MHD grid and $[1/64, 1/32, 1/64]R_G \sim [0.09, 0.19, 0.09]d_i$ for the Cartesian PIC grid. These result in a total number of 27 million cells in MHD and 2.5 million cells in PIC with 1.2 billion particles (256 particles per cell per species). The Hall MHD time-accurate run starts from the quasi-steady state after 80,000 steps, and the PIC simulation starts after 300s of the Hall MHD run. The sharp transition period represented by the beginning ~ 60 s time in Hall MHD simulation is ignored in the analysis. The MHD-EPIC model under this grid resolution takes 430 core hours to simulate 1s in physical time with Intel Xeon Platinum 8280 computing nodes on the Frontera supercomputer.

In order to resolve further to the electron scales near the upstream magnetopause, we have performed another short-duration higher-resolution MHD-EPIC run with PIC grid size $[1/128, 1/64, 1/128]R_G \sim [0.05, 0.1, 0.05]d_i$. Given the proton-electron mass ratio of 100 used in the simulation, this corresponds to $\Delta x = \Delta y = 0.05d_i = 0.5d_e$, $\Delta z = 0.1d_i = d_e$ inside the PIC domain, with 2.4 billion particles for each species (125 particles/cell). Such high resolution in a global magnetosphere model requires significant computing resources: 1 second simulation in physical time requires 750 core hours running with 4480 cores on Intel Xeon Platinum 8280 computing nodes. Therefore we only run at this resolution for ~ 100 s physical time

demonstrating the fully resolved electron and ion kinetics.

From the 20 min simulations of Hall MHD and MHD-EPIC which cover the entire G8 flyby magnetosphere crossing, we start with comparing the magnetic field with Galileo observations, and then continue to demonstrate the magnetopause dynamics, diffusion region properties, and reconnection rate estimations. In the end we show some interesting ULF wave detection cases from simulations.

4.1 Time-accurate Magnetic Field Comparison

Given that we have 20 min of simulation for both models with a 1s cadence output and the time between inbound/outbound magnetopause crossing by Galileo is about 10 min, we have identified the best fit to observations by shifting the starting time in the simulations. Figure 4.1 shows the magnetic field comparison with the G8 flyby close encounter observation (black) for Hall MHD (blue) and MHD-EPIC (orange). We align the simulation outputs from 15:45 ULT to 16:05 ULT, during which the magnetic field along the Galileo trajectory is extracted from different snapshots. The field data before 15:45 ULT and after 16:05 ULT are extracted from the first and last snapshot, respectively. Both models have in general nice agreements with the observation, even though we cannot fully reproduce the sharp transitions during the magnetopause crossings. With doubled grid resolution compared to our previous work [Zhou et al., 2019], small scale spatiotemporal perturbations start to show up. Hall MHD behaves more dynamic than the coupled MHD-EPIC model near the upstream reconnection regions. As have been shown in our previous study, the fluctuations during the inbound and outbound crossings are related to the magnetopause surface motion as well as flux rope generation. These will be discussed further in later sections. Note that the Galileo magnetometer data were collected at

a rate of 3 samples/s during the close flybys, which means that the perturbations with frequencies between 0.5 and 1.5 Hz are missing from the simulation due to the choice of 1s output cadence.

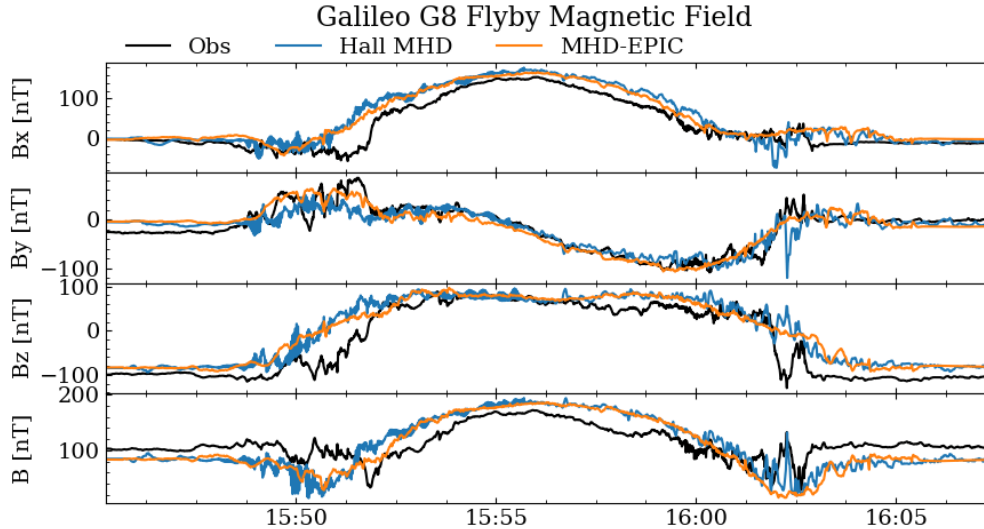


Figure 4.1: Magnetic field comparisons with Galileo observation during G8 flyby close encounter (black) for Hall MHD (blue) and MHD-EPIC (orange) simulations.

4.1.1 Plasma Pressure and Total Current Density

Although Hall MHD and MHD-EPIC runs show many similarities, such as the quadrupolar B_y magnetic field, we find certain quantities that are different especially around the upstream magnetopause. We select one snapshot from each model where FTEs do not exist and the reconnection X-line is roughly along the equator. Figure 4.2 shows the ion and electron pressures in the meridional cut from the two models. While the total plasma pressure is about the same, the energy partitions between ions and electrons are different: the kinetic PIC model shows more heated electrons than the Hall MHD model near the upstream magnetopause. The Hall MHD equations we solve [Zhou et al., 2019] do not include physical terms for controlling electron pressure except adiabatic heating, thus we have much cooler electrons near the magnetopause

in the Hall MHD results.

Figure 4.3 shows the total current densities in the meridional and equatorial cuts from the two models. While the location, magnitude and general shape of the current sheet look similar, it is thicker in the Hall MHD than in the MHD-EPIC model. In the meridional cut Figure 4.3b, the current density near the reconnection site extends more in the z -direction. Note that in the tail reconnection site we only have Hall MHD, so it serves as a nice reference between the upstream and tail reconnection regions in the two models.

4.2 Magnetopause Dynamics and FTEs

The magnetopause motion can be directly visualized with the movies in the supporting materials made from 3D data outputs. Figure 4.4 shows selected frames from the movie where the magnetopause surface is defined approximately by the $B_z = 0$ isosurface. Because of the small guide field B_y during the G8 flyby, we find $B_z = 0$ is a good approximation for the magnetopause surface. We select one quasi-steady snapshot and one highly-perturbed snapshot with flux ropes from each model and convert the vectors into the local LMN coordinates, where N points normal to the magnetopause outward into the upstream, L lies along the projection of the dipole axis onto the magnetopause (positive northward), and M completes the triad by pointing towards sub-Jovian side. The colored contour of ion pressure and velocity component u_L are displayed in the top and bottom rows, respectively.

The X-lines, shown by the white region where u_L diverges around zero, extend along the M direction on the magnetopause. The formation of long X-lines in both models is consistent with the prediction of onset conditions over the majority of Ganymede's magnetopause from an analytical model [Kaweeyanun et al. \[2020\]](#).

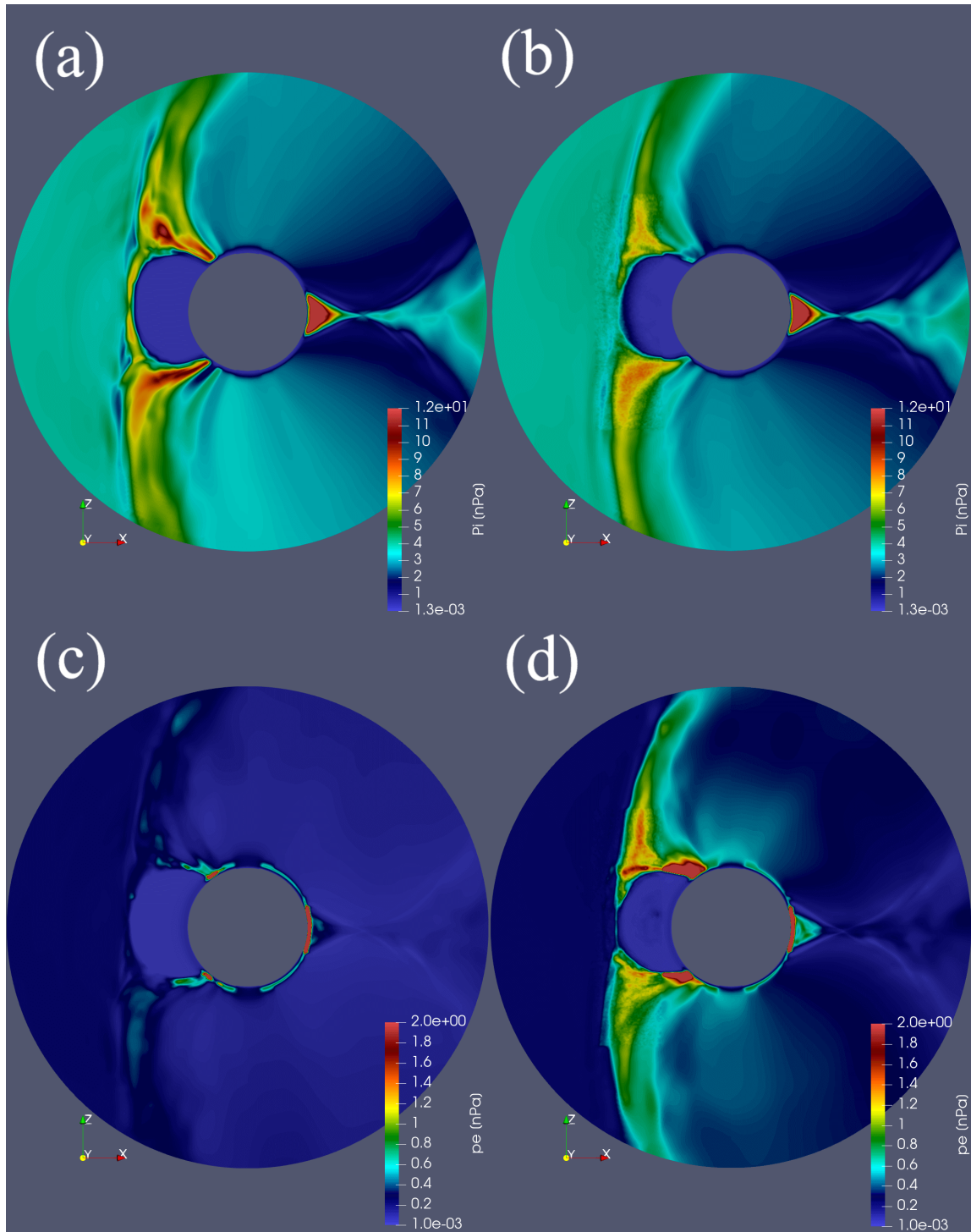


Figure 4.2: Ion and electron pressures in the meridional cut in (a,c) Hall MHD at $t = 595$ s and (b,d) MHD-EPIC at $t = 650$ s.

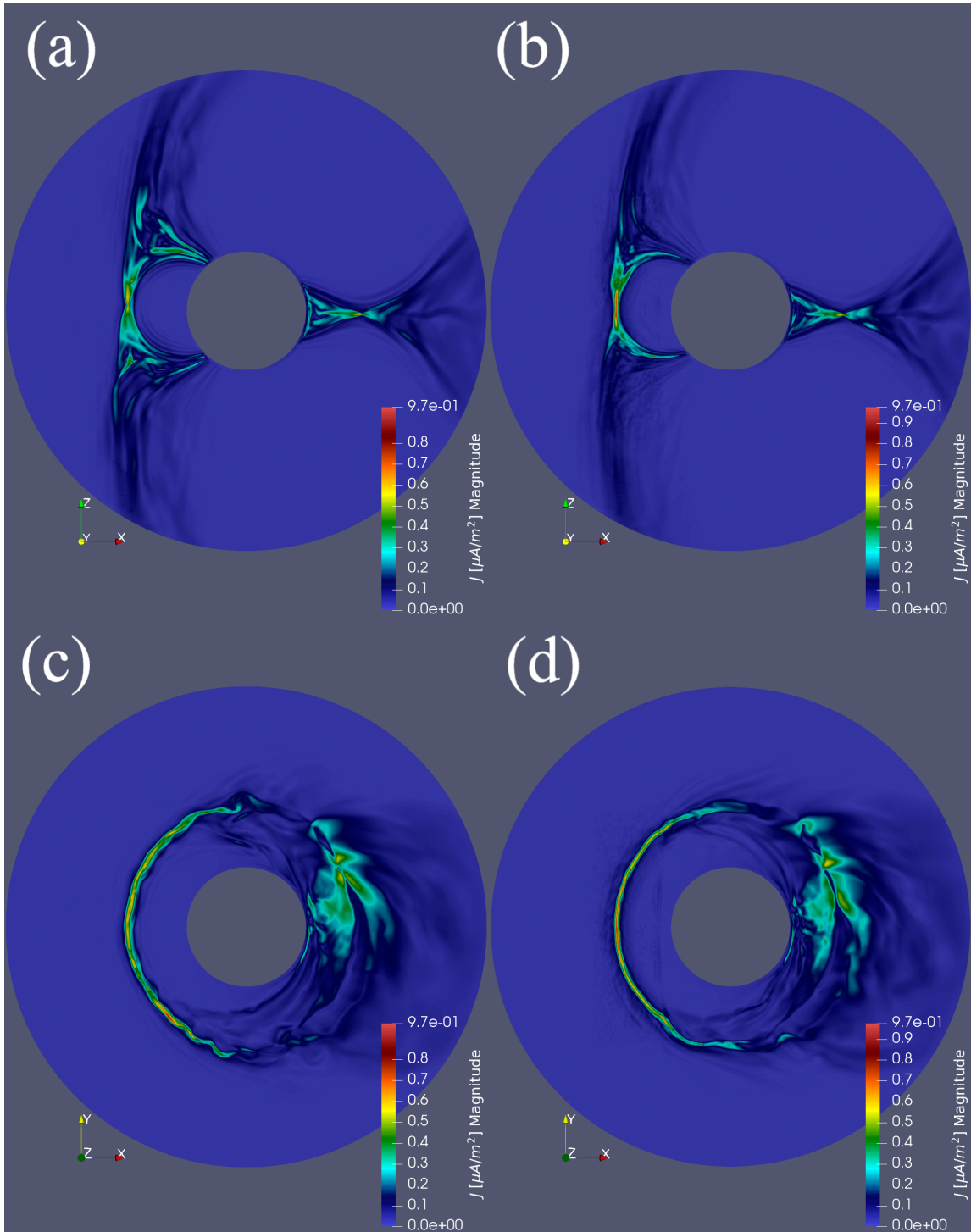


Figure 4.3: Total current density in the meridional and equatorial cut in (a,c) Hall MHD at $t = 595$ s and (b,d) MHD-EPIC at $t = 650$ s.

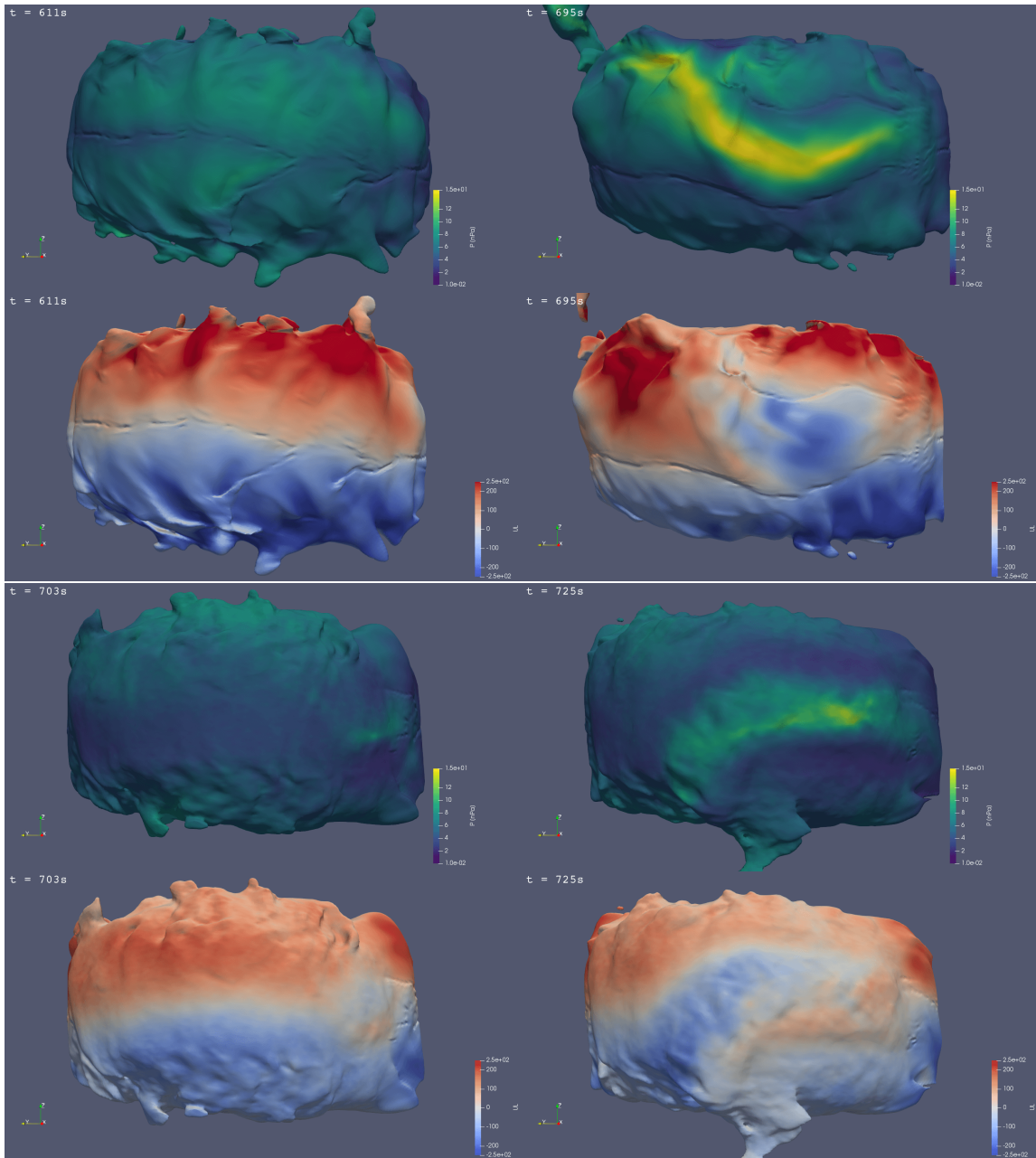


Figure 4.4: Magnetopause surface defined by $B_z = 0$ viewed from the upstream direction in (a) Hall MHD and (b) MHD-EPIC simulations. For each model, the plasma pressure is shown on the top, and the plasma velocity u_L component in the local LMN coordinates is shown at the bottom. The quasi-steady snapshots are shown on the left, and the snapshots with large flux ropes are shown on the right.

Plasma bulk flow on the flanks, as shown by Figure 7 in Zhou et al. [2019] for the G2 flyby, also suggests the extended reconnection sites across the upstream magnetopause. The intermittently generated flux ropes alter the long X-line near the equatorial plane and have high thermal pressure inside the core regions. At a later stage when large flux ropes are well developed, an enhancement of the core field B_y is observed (Figure 4.5), and the high thermal pressure persists in the core region. However, we note that from the simulations core fields are not always present in the identified flux ropes. This suggests that the classical force-free model can only explain part of the flux ropes being observed from simulations.

There is a more dynamic magnetopause surface in the Hall MHD simulation with larger magnitudes of plasma pressure and outflow velocity than in the MHD-EPIC simulation. The dark dip on the $B_z = 0$ illuminated contour surface along the velocity stagnation region in the Hall MHD does not show up in the MHD-EPIC simulations. Figure 4.3a-b in the meridional cut show that the X-line is thicker in Hall MHD results and there is a relatively sharp dip near the center. Thus this visual effect is related to the intrinsic differences of the X-line resolved by MHD and PIC. Full animations for the 20 min runs can be seen in the supplementary materials. In Ganymede’s G8 flyby simulations with constant Jovian upstream driving, we consistently observe magnetopause motion as well as flux rope generation in an intermittent manner. This suggests that there’s no truly steady state in Ganymede’s sub-Alfvénic magnetospheric plasma interaction.

By selecting a series of static satellites located on the average positions of the magnetopause, we are able to quantitatively characterize the generation of flux ropes from simulations. First we extract the average $B_z = 0$ locations on the meridional and equatorial plane from the simulation runs, which form two curved lines along the

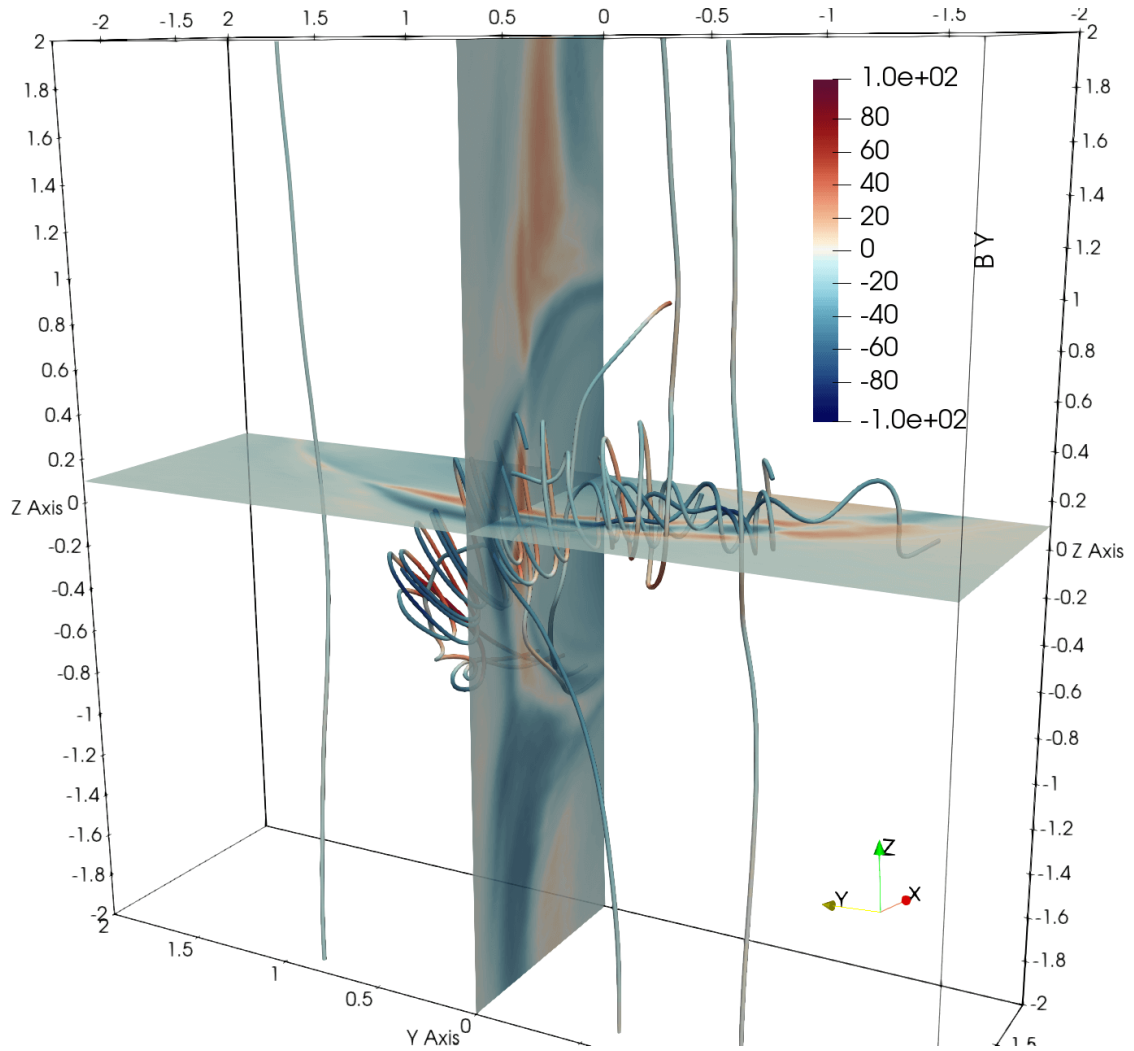


Figure 4.5: Example of well developed flux rope from MHD-EPIC simulation. The B_y colored contours in units of nT are shown in $z = 0.1R_G$ and $y = 0R_G$ cut planes. A core field is clearly present at the center.

center of the magnetopause. Then we interpolate the states onto these fixed locations over the simulation times. The thermal pressure perturbations with respect to the average pressure over a ± 100 s sliding window are shown as a function of spatial location and simulation time in Figure 4.6. A tilted red strip in the contour plots corresponds to a flux rope with increased thermal pressure in the core region moving across the meridional (a,b) and equatorial plane plane (c,d). Negative slopes in (a,b) represent downward propagating flux ropes and positive slopes represent upward

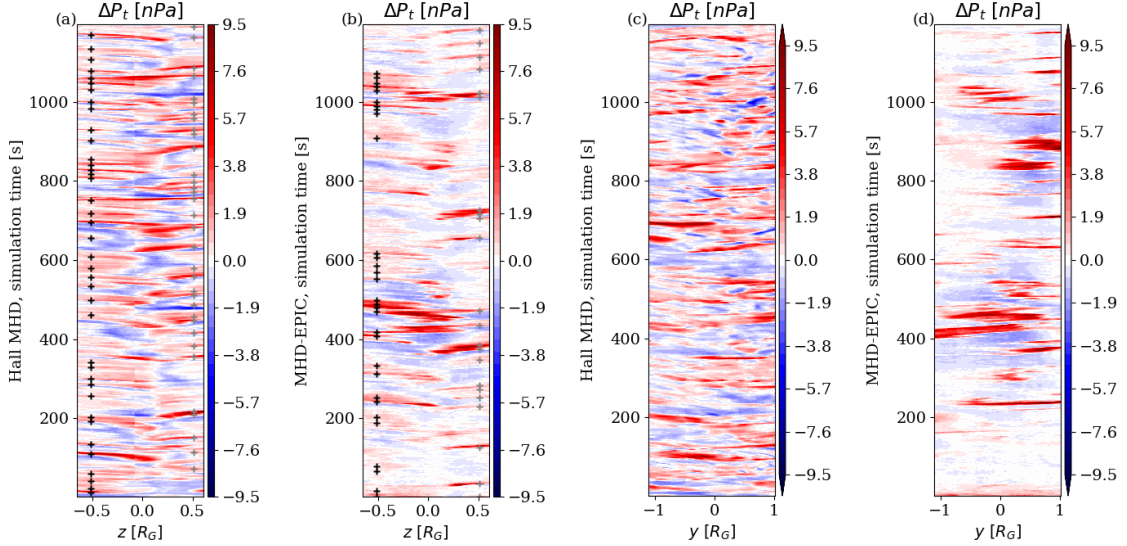


Figure 4.6: Motion of thermal pressure perturbations along the intersection lines of the magnetopause (defined as $B_z = 0$) and the meridional (a and b panels) or equatorial planes (c and d). The colors show the pressure perturbation relative to the mean pressure taken over a sliding ± 100 s interval. Panels (a) and (c) show Hall MHD results, while (b) and (d) are from MHD-EPIC. The gray and black + sign represent identified FTEs at $z = \pm 0.5 R_G$, respectively.

propagating flux ropes on the magnetopause. There is no clear asymmetry in the initial location or propagation direction.

We have checked that there is a one-to-one correspondence between a flux rope generated on the magnetopause (as can be seen from the movies) and a bright red strip in Figure 4.6. For example, the largest FTE in the Hall MHD and MHD-EPIC simulations happen at ~ 700 s and ~ 400 s, respectively, each corresponding to the brightest strips in Figure 4.6. Estimation on the slopes shows that the flux ropes in both Hall MHD and MHD-EPIC move at roughly the upstream Alfvén velocity $V_{A0} = 253$ km/s along the L direction on the magnetopause, consistent with theoretical expectation.

In the meridional cuts, we pick the total plasma pressure perturbations 1 standard deviation larger than the mean value at that location, and set it as the criterion for identifying an FTE. If there are multiple pressure peaks exceeding this threshold

within a 10s duration, only one FTE is counted. These thresholds are somewhat ad hoc due to the lack of a precise definition of FTEs. The identifications are shown with plus signs in Figure 4.6a-b at $z = -0.5R_G$ (black) and $z = 0.5R_G$ (gray). We find 73 and 51 FTEs from Hall MHD and MHD-EPIC simulations, respectively, which gives an average occurrence rate of ~ 3 FTEs per minute. Note that by counting FTEs in this way, we may miss those that never pass through the meridional cut. However, this is still a reasonable estimation given that most flux ropes initiate from the low latitude region, extend in the y direction, and move in the z direction.

From the equatorial cuts in Figure 4.6c-d, the average length of the flux ropes is about $0.8R_G$ in the y direction, which corresponds to roughly $1R_G$ in total length considering the curvature of the magnetopause in the x-y plane. Additionally, many of the flux ropes have one side tilted towards higher latitudes (e.g. Figure 4.5), so the average length may be even larger.

4.3 Kinetic Signatures near the Diffusion Region

With the embedded PIC model using grid resolution comparable to the electron skin depth, we are able to obtain detailed information about electrons and ions directly by looking at kinetic particles and their velocity distributions near the reconnection sites. At low latitudes in the GPhiO Cartesian coordinates during the G8 flyby with dominantly north-south magnetic field, the LMN coordinate system of a reconnection site is approximately aligned with the GPhiO system. Therefore, approximately $u_x \sim u_N$ is the inflow velocity, $u_z \sim u_L$ is the outflow velocity, and $u_y \sim u_M$ is the out-of-plane velocity. Note that the positive x direction in the GPhiO coordinate system is pointing toward the moon, which is the opposite of that in the GSE coordinate system.

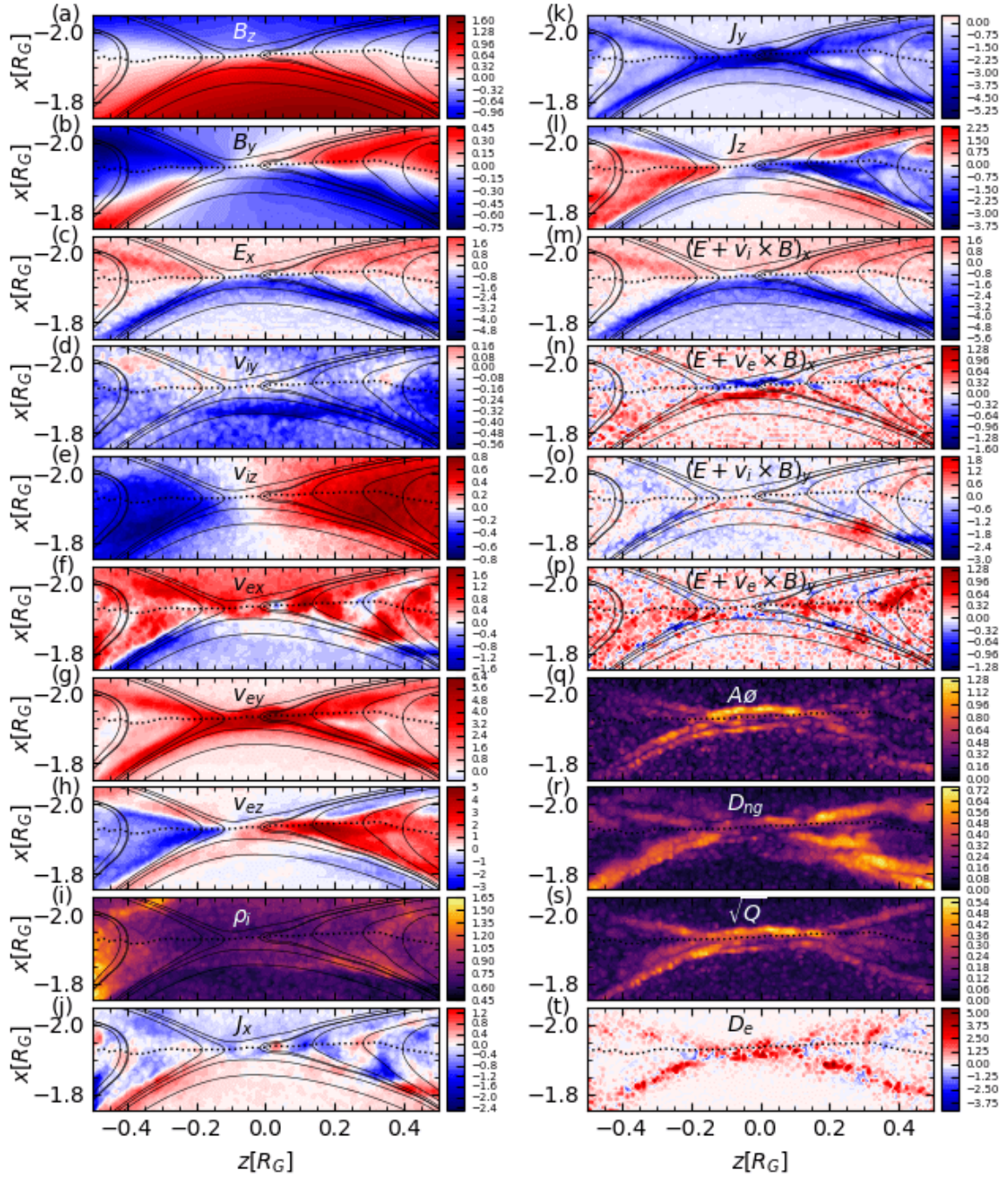


Figure 4.7: Normalized quantities in meridional plane from MHD-EPIC G8 flyby simulation near the reconnection site. $A\theta$, D_{ng} , Q are three non-gyrotropy measures [Scudder and Daughton, 2008, Aunai et al., 2013, Swisdak, 2016] and D_e is a dissipation measure [Zenitani et al., 2011]. Solid black lines are the mapped magnetic field and dotted lines show the locations where $B_z = 0$. Values with signs are colored with red-white-blue colormaps centered at 0.

Figure 4.7 shows one snapshot from the highest resolution simulation in the meridional plane near the reconnection site for the magnetic and electric fields, electron and ion bulk velocities, current density, plasma density, and different measures of the violation of the ion and electron frozen-in conditions. Magnetic field, particle number densities and velocities are normalized to the upstream field strength $B_0 = 86.8$ nT, number density $n_0 = 4 \text{ cm}^{-3}$ and Alfvén velocity $V_{A0} = 253 \text{ km/s}$, respectively. Electric field is normalized to $E_0 = V_{A0}B_0 = 22 \text{ mV/m}$, and current densities are normalized to $J_0 = en_0V_{A0} = 0.16 \mu\text{A/m}^2$. The quadrupolar out-of-plane magnetic field B_y in Figure 4.7b extends from the electron diffusion region, and the Hall electric field E_x in Figure 4.7c shows strong peaks along the separatrices. Ions are accelerated to $\sim V_{A0}$ in the exhaust region, with a drift in the -y direction (Figure 4.7d-e) peaked on the magnetospheric side and small counter streaming portion on the Jovian side. Electrons move into the diffusion region around the X-line and are accelerated to $\sim 5V_{A0}$ (the electron/ion mass ratio in the PIC model is 100) in the outflow region, with a large drift in the +y direction (Figure 4.7f-h). Figure 4.7i shows the non-collocation of X-line center (along the dotted line) and flow stagnation point (indicated by the white color), where the latter is on the magnetospheric side of the X-line [Cassak and Shay \[2007\]](#).

4.3.1 Violations of Frozen-in Condition

Figure 4.7m-p show the x and y components of $\mathbf{E} + \mathbf{V} \times \mathbf{B}$ for ions and electrons, respectively. These represent the violation of the frozen-in condition, or in other words, the deviation of the model from ideal MHD and Hall MHD, respectively. No clear signatures can be identified solely for the diffusion region, although the x component of $\mathbf{E} + \mathbf{V}_e \times \mathbf{B}$ show dipolar peaks near the center.

4.3.2 Non-gyrotropy Measures

Three different scalar non-gyrotropy measures $A\emptyset$, D_{ng} , and Q [Scudder and Daughton, 2008, Aunai et al., 2013, Swisdak, 2016] for electrons are shown in Figure 4.7q-s. The three non-gyrotropy measures mentioned in the paper are all scalars independent of the coordinate. They can be computed efficiently point-wise with the following equations. Note that the electron subscripts are dropped in all the following equations.

The first measure $A\emptyset$ is defined as

$$A\emptyset = 2 \frac{|P_{\perp 1} - P_{\perp 2}|}{P_{\perp 1} + P_{\perp 2}}, \quad (4.1)$$

where the subscripts 1 and 2 represent the two orthogonal perpendicular directions to the magnetic field. It has been shown by Scudder and Daughton [2008] that in any frame (x, y, z) , if we define

$$\begin{aligned} N_{xx} &= b_y b_y P_{zz} - 2b_y b_z P_{yz} + b_z b_z P_{yy}, \\ N_{xy} &= -b_y b_x P_{zz} + b_y b_z P_{xz} + b_z b_x P_{yz} - b_z b_z P_{xy}, \\ N_{xz} &= b_y b_x P_{yz} - b_y b_y P_{xz} - b_z b_x P_{yy} + b_z b_y P_{xy}, \\ N_{yy} &= b_x b_x P_{zz} - 2b_x b_z P_{xz} + b_z b_z P_{xx}, \\ N_{yz} &= -b_x b_x P_{yz} + b_x b_y P_{xz} + b_z b_x P_{xy} - b_z b_y P_{xx}, \\ N_{zz} &= b_x b_x P_{yy} - 2b_x b_y P_{xy} + b_y b_y P_{xx}, \end{aligned}$$

and

$$\begin{aligned} \alpha &= N_{xx} + N_{yy} + N_{zz}, \\ \beta &= -(N_{xy}N_{xy} + N_{xz}N_{xz} + N_{yz}N_{yz} - N_{xx}N_{yy} - N_{xx}N_{zz} - N_{yy}N_{zz}), \end{aligned}$$

then the agyrotropy can be expressed as

$$A\emptyset = 2 \frac{\sqrt{\alpha^2 - 4\beta}}{\alpha}. \quad (4.2)$$

The second measure of non-gyrotropy D_{ng} suggested by [Aunai et al. \[2013\]](#) for electrons can be computed via

$$D_{ng} = 2 \frac{\sqrt{P_{xy}^2 + P_{xz}^2 + P_{yz}^2}}{P_{xx} + P_{yy} + P_{zz}} \quad (4.3)$$

The third measure proposed by [Swisdak \[2016\]](#) can be computed via

$$Q = 1 - 4 \frac{I_2}{(I_1 - P_{\parallel})(I_1 + 3P_{\parallel})} \quad (4.4)$$

where $I_1 = P_{xx} + P_{yy} + P_{zz}$ is the trace and $I_2 = P_{xx}P_{yy} + P_{xx}P_{zz} + P_{yy}P_{zz} - (P_{xy}P_{yx} + P_{xz}P_{zx} + P_{yz}P_{zy})$ is the principle minor.

$A\emptyset$ shows the non-gyrotropy in the plane perpendicular to the magnetic field, which peaks at the electron diffusion region and gets enhanced along the separatrices, especially on the magnetospheric side. It behaves similarly to the later proposed \sqrt{Q} which is based on the property of positive semi-definite matrix and takes the full pressure tensor into account. The other non-gyrotropy measure D_{ng} , which scales with the ratio between the Frobenius norm of the non-diagonal terms and the trace of the pressure tensor, peaks near the X-line along the separatrices but is not localized at the central electron diffusion region. As with the mathematical counter examples proposed in [[Swisdak, 2016](#)], we also found that \sqrt{Q} is more accurate in describing the non-gyrotropy effect near the reconnection site.

4.3.3 Dissipation Measure

Finally, a frame independent dissipation measure derived from energy conversion $D_e = \mathbf{J}' \cdot \mathbf{E}' = \mathbf{J} \cdot (\mathbf{E} + \mathbf{V}_e \times \mathbf{B}) - (n_i - n_e)\mathbf{V}_e \cdot \mathbf{E}$ [[Zenitani et al., 2011](#)] is shown

in Figure 4.7t. D_e peaks at the reconnection site, and is also enhanced along the separatrices.

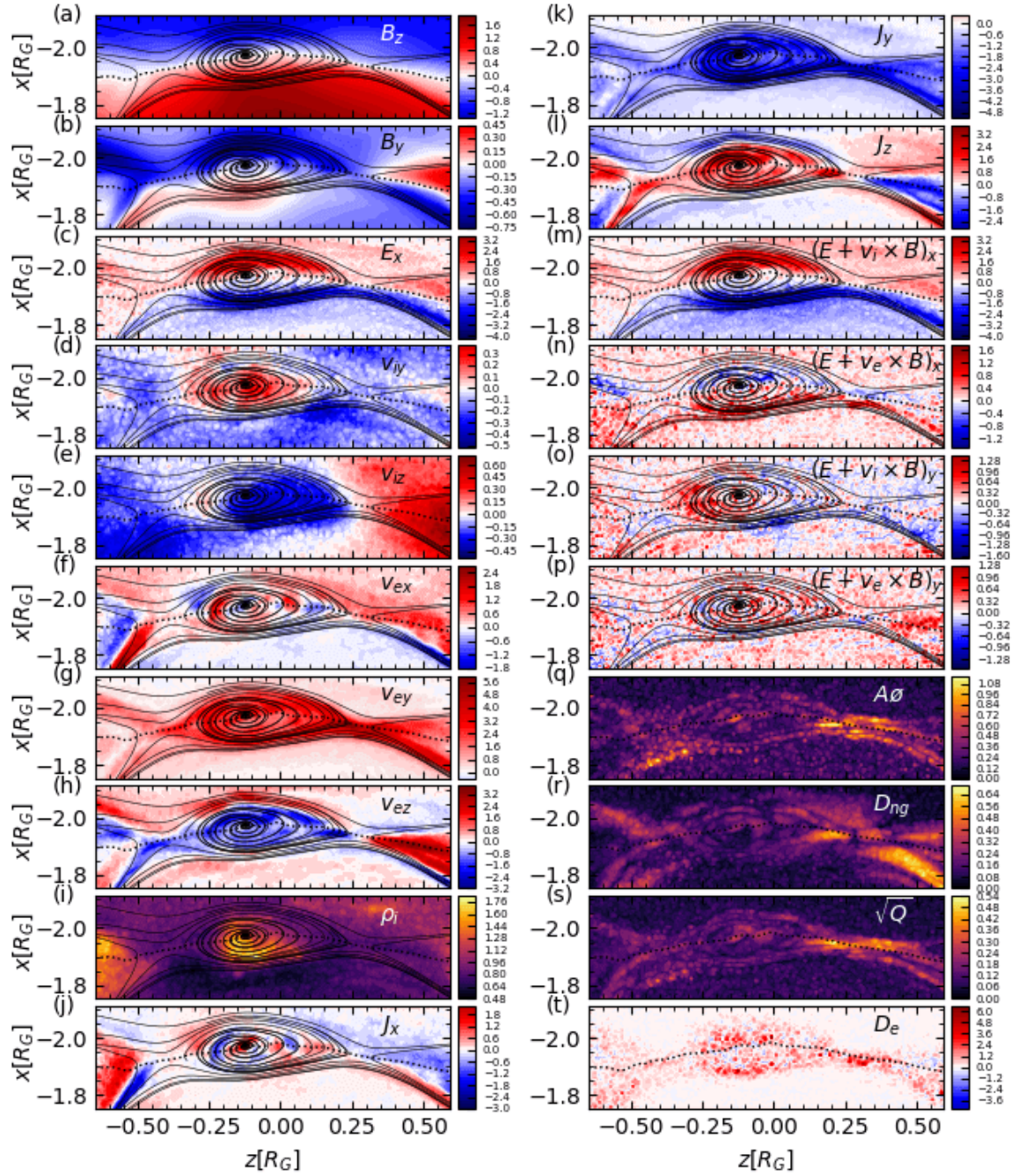


Figure 4.8: Same quantities as in Figure 4.7, but at a time when a flux rope is present in the MHD-EPIC simulation.

4.3.4 Reconnection Site with a Flux Rope

It is interesting to see how these quantities look like near a flux rope formed between two reconnection sites. A snapshot with a flux rope is shown in Figure 4.8. The original X-line is near $z = 0.25R_G$, and the subsequently formed one is near $z = -0.35R_G$. Inside the flux rope, we observe an increase of normal electric field E_x on the Jovian side, oppositely drifting ions in Figure 4.8d-e, perturbations of electron velocities in Figure 4.8f-h, enhancement of density in Figure 4.8i, and the expansion of core current J_y in Figure 4.8k. The ion outflow in the z direction from the new X-line encounters the stronger outflow in negative z direction from the original X-line, thus turns into a drift in the y direction. The non-gyrotropy measures (Figure 4.8q-s) decreases inside the flux rope, but the diffusion measure (Figure 4.8r) gets enhanced.

During the simulation, flux ropes inside the exhaust region do not always show all the corresponding kinetic signatures in the meridional cut: we have seen snapshots (not shown) of small flux ropes with little influence of ion outflow velocity and currents. In general, none of the presented quantities can uniquely identify the electron diffusion region, even though some measures perform better than others. The presence of flux ropes makes the detection even more complicated, both in observations and simulations. As suggested in [Shay et al. \[2016\]](#), one should rely on complementary approaches for identification.

4.3.5 Phase Space Distributions

The selected electron and ion phase space distribution functions (boxes 1-4 for electrons, boxes 5-8 for ions) around the reconnection site (at the same simulation time as in Figure 4.7) are plotted in Figure 4.9. For electrons, the sampled box

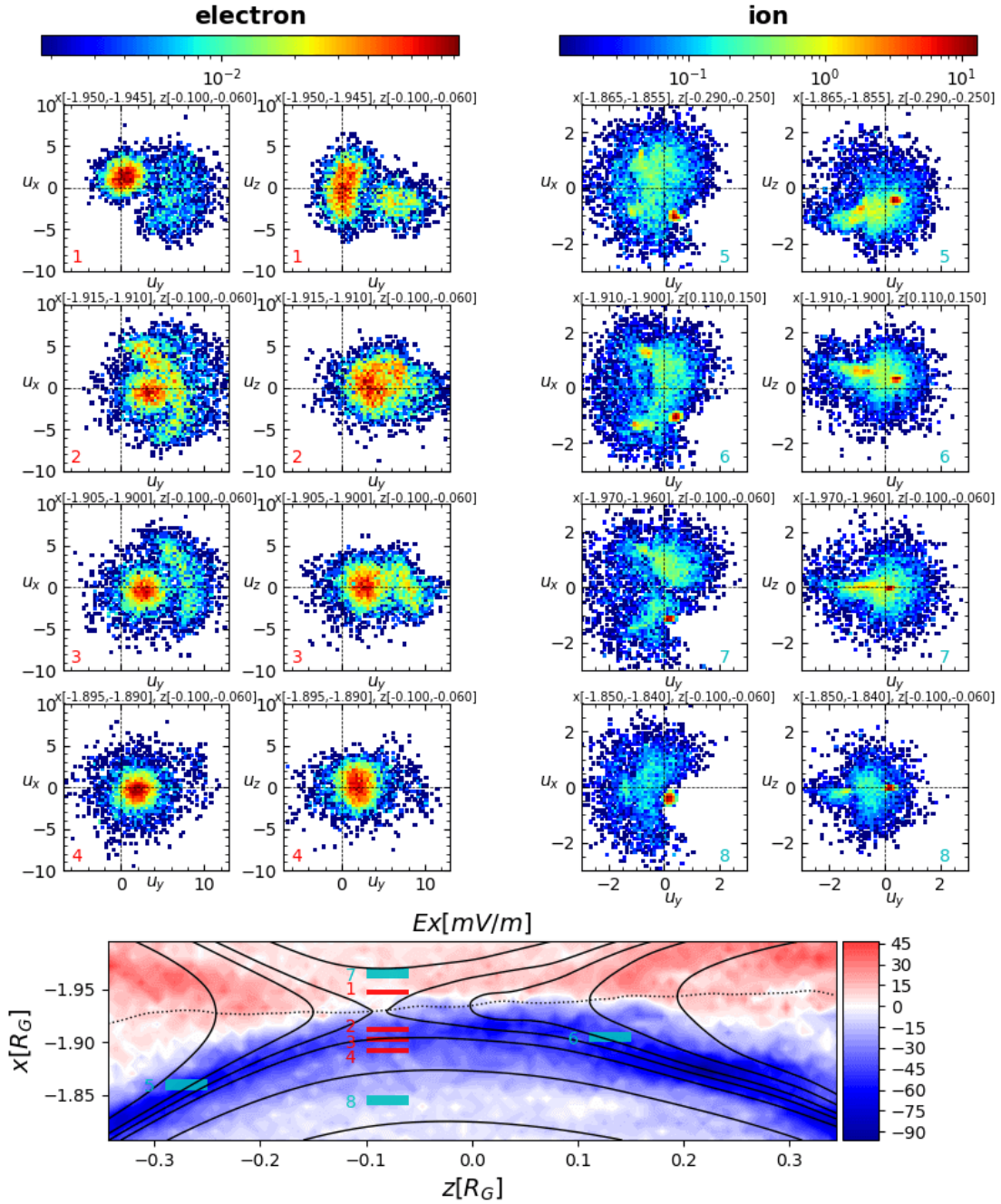


Figure 4.9: Top panels: normalized velocity distribution functions of electrons and ions near the meridional plane in selected boxes shown in the bottom plot. For each species, the integrated $u_y - u_x$ distributions are presented on the left, and the $u_y - u_z$ distributions is presented on the right. Bottom panel: $y = 0$ equatorial cut near the X-line with color contours of E_z , mapped magnetic field lines and a dotted line along the magnetopause of $B_z = 0$. Positive x direction points towards the moon. The selected electron box regions 1-4 are colored in red, and ion box regions 5-8 are colored in cyan.

regions have a width of $0.005R_G \sim 0.3d_e$ in the x direction and $0.04R_G \sim 2.6d_e$ in the z direction; for ions, the sampled box regions have a width of $0.01R_G \sim 0.064d_i$ in the x direction and $0.04R_G \sim 0.3d_i$ in the z direction. In the y direction all the boxes extend from $-0.08R_G$ to $0.08R_G$. In the electron diffusion region, the crescent shape distributions can be observed close to the peak location of E_x and $B_z = 0$ midplane, which is referred to as the "shoulder" region by [Shay et al. \[2016\]](#). Moving farther away from the X-line (Box 2-3), the electrons coming from the Jovian side get further accelerated by E_x , which creates the clear gap from the magnetospheric electrons. We can observe a shift of the stagnation point towards the magnetospheric side, consistent with Figure 4.7f. In Box 4 at about $2.2d_e$ away from the X-line center, the penetration of electrons from Jovian upstream into the Ganymede's magnetosphere nearly vanishes.

For ions, in boxes 5 and 6 along the separatrices near the exhaust region, the $u_y - u_z$ velocity distribution cuts are nearly symmetric. In similar regions of Earth-like simulations [Broll et al. \[2017\]](#) and observations [Smith and Rodgers \[1991\]](#), the so-called "D-shaped" ion distributions have been found. However, no clear signatures of ion "D-shaped" distribution is found here in our simulation. On the upstream side (box 7), the majority of ions are moving towards the X-line with positive u_x , but there are also reflected ions with negative u_x . On the magnetospheric side, ion crescent shape distributions can be found in a wide region $\sim 1d_i$ away from the X-line center (e.g. box 8).

The series of distribution functions can give us an estimation of the size of diffusion regions in reality, using the fact that electron isotropy breaks inside the electron diffusion region. Note that the proton-electron mass ratio is set to 100 in the simulations, therefore we need to convert the length in the simulations back to the real

physical units. Along the center cut through the X-line in the x direction, the distributions become isotropic at about $1.5d_e$ and $2.5d_e$ away from the center on the Jovian upstream side and magnetospheric side, respectively. From Figure 4.7, the diffusion region extension in the z direction is about $0.1R_G \sim 6d_e$ in the simulation. As a result, the actual upstream electron diffusion region in nature is about $4d_e \sim 11$ km wide in x and $6d_e \sim 16$ km wide in z.

The results here have many similar features as in the asymmetric local 2D explicit PIC simulation with grid resolution 20 cells per d_i and 4 cells per d_e by [Shay et al. \[2016\]](#). In the normalized unit length, the ion resolutions in these two simulations are the same and the electron resolutions in MHD-EPIC is half of that in the local 2D PIC simulation. We note that in the implicit PIC simulation ~ 2 cells per d_e is the minimum requirement to accurately resolve electron kinetic signatures, and the coupled MHD-EPIC model has the capability of capturing both local ion and electron kinetic physics in a global magnetosphere simulation with adequate resolution. However, one must be cautious in comparing the results with the local PIC simulation. The key differences are: (1) in the Ganymede simulations we are adopting upstream magnetic fields with all three components and measured tilted dipole field, compared with an idealized, pure B_z setup in the local PIC simulation; (2) our Ganymede simulations do not show a large density jump across the magnetopause (Figure 4.7i), and there is a large electron drift along the M (approximately y) direction due to the curvature of \mathbf{B} and the Hall effect [Zhou et al. \[2019\]](#). In [Shay et al. \[2016\]](#), the density between the sheath and magnetosphere differs by a factor of 10.

4.4 Reconnection Rate

In order to understand the global effects of magnetopause reconnection in this sub-Alfvénic system and compare the predictions between two different global models that contain different approximations of physics, we need to come up with a quantitative description of the reconnection efficiency. One approach is to calculate the global reconnection efficiency defined by the ratio of the imposed electric field integral on the magnetopause to the full possible convective electric field integral across the width of the magnetosphere. Physically, this quantity represents how much magnetic flux get passed into the magnetosphere through upstream reconnection. [Kivelson et al., 1997] first applied this idea to the G2 flyby observation and found an upper limit of nearly 100% reconnection efficiency, indicating a highly efficient reconnection process. [Hu et al., 2007] described in detail about various methods of computing the electric field integral, or total reconnection rate, in global MHD simulations. As pointed out in their estimation, the convective electric field dominates in the upstream half of the equatorial plane, whereas the interplanetary magnetic field lines nearby the upstream half of the reconnection layer are almost equipotential.

In a time-varying dynamical reconnection system with intermittent FTEs, it is very difficult to get all the local reconnection sites at the right locations and do the electric field integral in a proper manner. We pursue a different approach based on the fact that the upstream reconnection corresponds to a topological change: an open magnetic field line with both ends connected to the Jovian field and a fully-closed field line connected to Ganymede at both ends reconnect into half-open field lines connected to Ganymede at one end. We can measure total reconnection

rate as the change in the total half-open magnetic flux. For the Jovian field aligned approximately with the Z direction, taking a plane at $Z = 2 R_G$, for example, will cut through all the open field lines on the northern hemisphere as shown in Figure 4.10. Figure 4.10a shows the 3D view of the field lines that connects to the upstream and tail reconnection regions in red and green, respectively. Figure 4.10b shows the field line topology on the slice, with B_z contours representing the sampled magnetosphere region, red line representing the upstream boundary U and blue line representing the middle cut M that closes the surface A .

The Leibniz integral rule for a two dimensional surface moving in three dimensional space is [Flanders, 1973]:

$$\frac{d}{dt} \iint_{A(t)} \mathbf{F}(\mathbf{r}, t) \cdot d\mathbf{A} = \iint_{A(t)} \left(\frac{\partial}{\partial t} \mathbf{F}(\mathbf{r}, t) + [\nabla \cdot \mathbf{F}(\mathbf{r}, t)] \mathbf{v}_A \right) \cdot d\mathbf{A} - \oint_{\partial A(t)} [\mathbf{v}_A \times \mathbf{F}(\mathbf{r}, t)] \cdot d\mathbf{l}, \quad (4.5)$$

where

$\mathbf{F}(\mathbf{r}, t)$ is a vector field at the spatial position \mathbf{r} at time t ,

A is a surface bounded by the closed curve ∂A ,

$d\mathbf{A}$ is a vector element of the surface A ,

$d\mathbf{l}$ is a vector element of the curve ∂A ,

\mathbf{v}_A is the velocity of movement of the region A .

In our case, the vector field \mathbf{F} is the magnetic field \mathbf{B} . Due to the divergence-free property of \mathbf{B} , we have

$$\frac{d}{dt} \iint_{A(t)} \mathbf{B}(\mathbf{r}, t) \cdot d\mathbf{A} = \iint_{A(t)} \frac{\partial}{\partial t} \mathbf{B}(\mathbf{r}, t) \cdot d\mathbf{A} - \oint_{\partial A(t)} \mathbf{v}_A \times \mathbf{B}(\mathbf{r}, t) \cdot d\mathbf{l}. \quad (4.6)$$

The time derivative of magnetic field can be expressed as the curl of electric field from Faraday's law of induction:

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E}. \quad (4.7)$$

With the help of Stokes' theorem, Equation 4.6 can be written as

$$\begin{aligned} \frac{d}{dt} \iint_{A(t)} \mathbf{B}(\mathbf{r}, t) \cdot d\mathbf{A} &= - \iint_{A(t)} \nabla \times \mathbf{E}(\mathbf{r}, t) \cdot d\mathbf{A} - \oint_{\partial A(t)} \mathbf{v}_A \times \mathbf{B}(\mathbf{r}, t) \cdot d\mathbf{l} \\ &= \oint_{\partial A(t)} -[\mathbf{E} + \mathbf{v}_A \times \mathbf{B}] \cdot d\mathbf{l}. \end{aligned} \quad (4.8)$$

Therefore, the time derivative of the magnetic flux passing through a closed surface equals the opposite of the electric field integral along the boundary *in the comoving frame of the boundary curve*. As shown in Figure 4.10b, the upstream reconnection corresponds to the flux passing through the boundary U on the left where the velocity points inward to the surface. As it is difficult to accurately estimate the motion of the boundary, we replace the integral of the electric field along the moving boundary with the mathematically equivalent time derivative of magnetic flux plus the electric field integral along the rest of the boundary curve M where the flow points to the +x direction and can be regarded as stationary by choosing a fixed line enclosing the surface A . We note that the results don't depend on the choice of M as long as the flow points outward of surface A along it.

With $\partial A = U + M$, Equation 4.8 can be rearranged to get the total upstream reconnection rate as

$$R_t \equiv \int_U [\mathbf{E} + \mathbf{v}_A \times \mathbf{B}] \cdot d\mathbf{l} = -\frac{d}{dt} \iint_{A(t)} \mathbf{B}(\mathbf{r}, t) \cdot d\mathbf{A} - \int_M [\mathbf{E} + \mathbf{v}_A \times \mathbf{B}] \cdot d\mathbf{l}. \quad (4.9)$$

Since the middle line is stationary, $\mathbf{v}_M \equiv 0$. In Hall MHD, electric field can be expressed as

$$\mathbf{E} = -\mathbf{V}_e \times \mathbf{B}, \quad (4.10)$$

where \mathbf{V}_e is the electron bulk velocity. Therefore we have

$$R_t = -\frac{d}{dt} \iint_{A(t)} \mathbf{B}(\mathbf{r}, t) \cdot d\mathbf{A} + \int_M \mathbf{V}_e \times \mathbf{B} \cdot d\mathbf{l}. \quad (4.11)$$

We thus calculate the upstream reconnection rate by computing the two terms on the right-hand-side of Equation 4.11 numerically. We cut a slice plane at $z = 2R_G$, trace the field lines that pass through the plane, and find the half-open field line boundary curve on the slice. The surface integral of A and the line integral along M are evaluated from the magnetic field and electron velocity interpolated to the fine grid. The time derivative of the flux is taken with simple finite differencing of the surface integrals at a 1s cadence. The middle line is picked at $x = 1.28 R_G$, where its length is the largest along y , so the flow points inward along U and outward along M .

The width of the magnetosphere L is taken as the extent of the closed field line region parallel to the external convective electric field $-\mathbf{V} \times \mathbf{B}$. For the Jovian field B approximately parallel to Z axis, the width can be taken as $L \approx 4 R_G$ in the Y direction and the upstream electric field integral $\Delta V = |V_x B_z| L \approx 130$ kV. The global reconnection efficiency ϵ is then given by

$$\epsilon = R_t / \Delta V. \quad (4.12)$$

The results are shown in Figure 4.11a-b for Hall MHD and MHD-EPIC simulations respectively. Regardless of the intrinsic differences between the two models, both give roughly $R_t = 83$ kV or equivalently $\epsilon \approx 0.64$. This indicates about 60% of the plasma flowing onto Ganymede's magnetosphere crosses the magnetopause, which is quite efficient.

To identify if there is any connection between the FTEs and reconnection efficiency, we checked the correlation between FTEs occurrence time and changes of ϵ . Because the field line tracing is done for each snapshot, the field line connectivity and the corresponding change of the open magnetic flux are passed from the upstream reconnection sites to the magnetic flux enclosed by the open-closed boundary

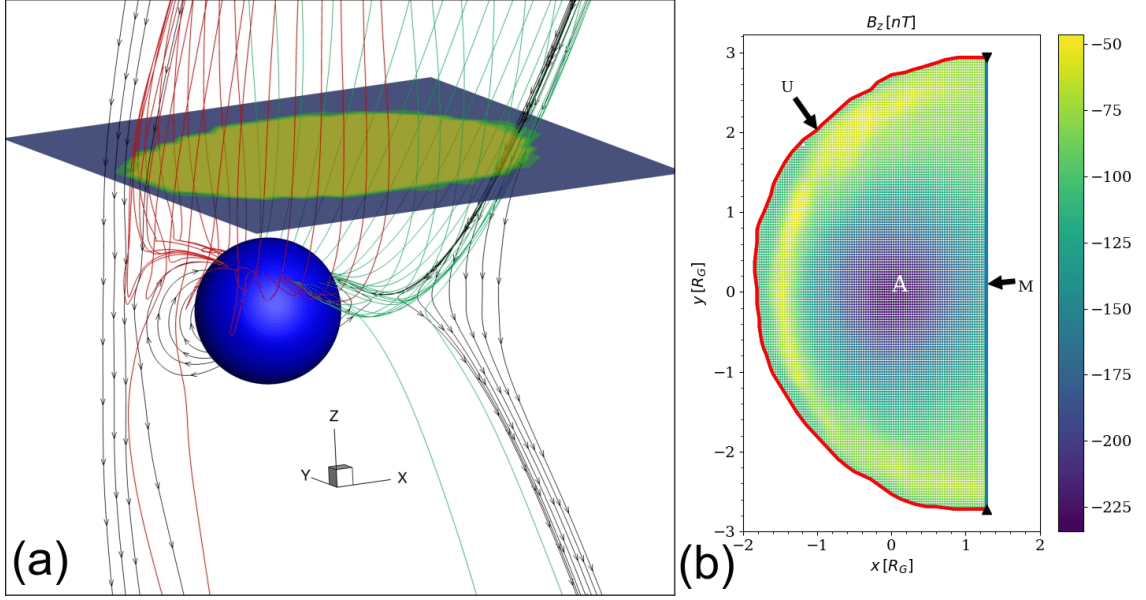


Figure 4.10: Illustration of reconnection efficiency calculation in GPhiO coordinates. (a) shows a 3D view of the magnetic field geometry near Ganymede (represented by a blue sphere). The black lines are magnetic field lines with starting points in the $y = 0$ plane, red lines are ones that just get reconnected at the upstream magnetopause, and green lines are those that connect to the tail reconnection site. (b) displays the upstream half of the half-closed field line region colored with B_z in the $z = 2 R_G$ plane corresponding to the cut plane in (a). The upstream boundary curve U is shown by the red line. The middle straight line M colored in blue closes the boundary of surface A .

curve in the $z = 2 R_G$ plane immediately. The red and green dashed lines in Figure 4.11a-b represent the identified occurrence times from Figure 4.6 for FTEs with total plasma pressure perturbation larger than 1.5 standard deviation moving northward and southward, respectively. The majority of lines coincide with the local peaks of ϵ , suggesting an increase of reconnection efficiency during the FTEs and a decrease of efficiency afterwards. However, we also found that even without relatively large FTEs (e.g. between $t = 700 s$ and $t = 950 s$), the reconnection rate is still fluctuating. Therefore we cannot confirm that the perturbation in upstream reconnection rate is purely related to FTEs.

For the sake of diagnosing if there are any periodicities related to the reconnection, we performed FFTs on the reconnection efficiencies from the two models. The results are shown in Figure 4.11c. In general, the FFT spectra of the estimated reconnection

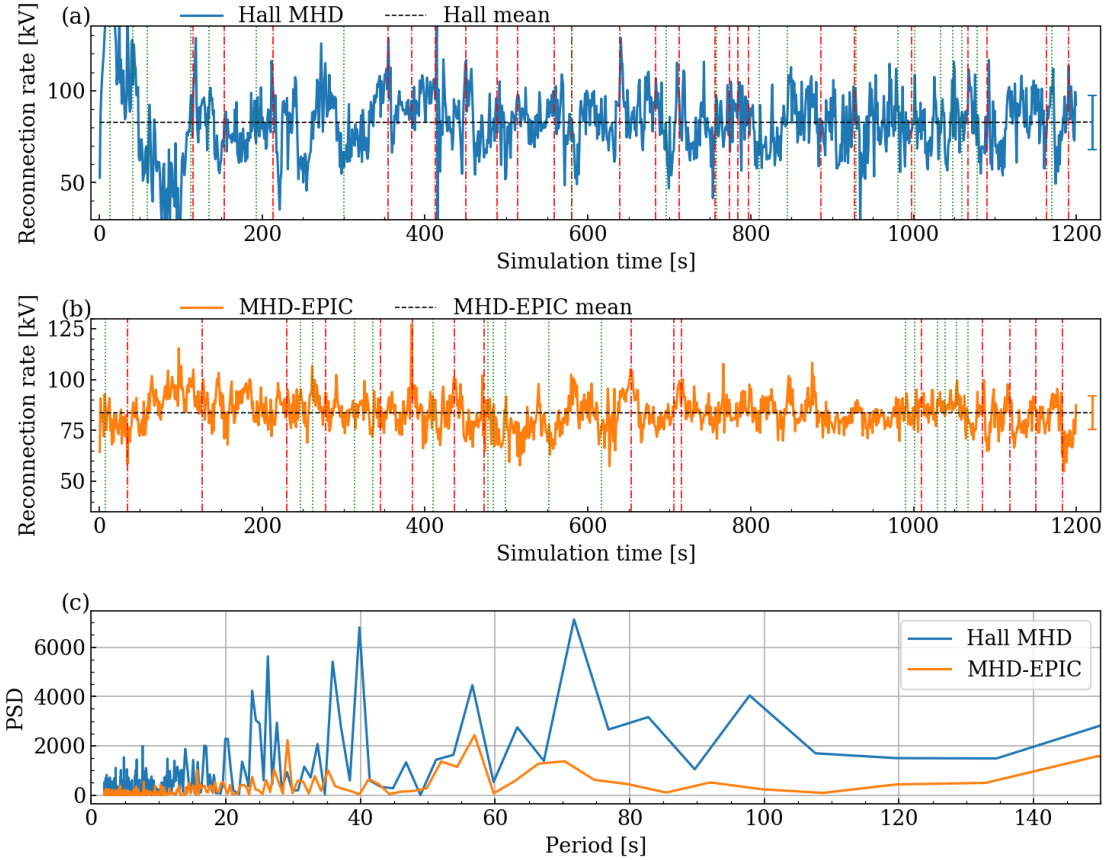


Figure 4.11: Global upstream reconnection rates from (a) Hall MHD in blue and (b) MHD-EPIC in orange throughout the 20-minute simulations. The average rate is 82.9 ± 18.3 kV for Hall MHD and 83.6 ± 8.2 kV for MHD-EPIC. Note that MHD-EPIC starts from $t = 300$ s of the Hall MHD run. The black dashed lines represent the means of the reconnection rate, with standard deviation bar on the right. The red and green dash-dotted lines represent sample FTEs identified from large thermal pressure perturbations on the magnetopause in Figure 4.6 at $z = \pm 0.5 R_G$, respectively. (c) Periodograms of the global upstream reconnection rate from Hall MHD in blue and MHD-EPIC in orange. x axis is the period and y axis is the power spectrum density.

efficiency from both models do not show any dominant periodicity, although there are multiple, relatively weak peaks around the 1 minute period (for Hall MHD, peaks at 26 s, 40 s, 55 s, 72 s and 110 s; for MHD-EPIC, peaks at 29 s and 57 s).

4.5 Wave Detection

4.5.1 Ultralow Frequency Wave

The 1 s output cadence allows us to study plasma waves in the ultralow frequency range. There are three kinds of MHD waves in a uniform plasma: the Alfvén

wave, fast and slow magnetosonic waves. Let \mathbf{v} , \mathbf{B} , ρ and θ be the plasma velocity, magnetic fields, plasma density and the angle between velocity and magnetic field, respectively. The background quantity magnitudes are represented with subscript 0 and the perturbed quantities are denoted with δ . The perturbed quantities of Alfvén waves follow the relation:

$$\frac{\delta \mathbf{v}}{v_A} = \pm \frac{\delta \mathbf{B}}{B_0} \quad (4.13)$$

where v_A is the Alfvén velocity, the "+" is for Alfvén waves propagating along positive \mathbf{B} direction, and "-" for waves propagating along negative \mathbf{B} direction.

For the slow and fast waves, the perturbed plasma and magnetic pressure follow

$$\delta P_B = \frac{v_A^2}{v_s^2} \left(1 - \frac{k^2 v_s^2 \cos^2 \theta}{\omega^2} \right) \delta P_t \quad (4.14)$$

where v_s is the sound speed, v_A is the Alfvén speed, \mathbf{k} is the wave number vector and θ is the angle between \mathbf{B}_0 and \mathbf{k} . For fast waves, the phase velocity $v_p = \omega/k$ will be larger than $v_s \cos \theta$, so δP_B and δP_t will have the same sign; for slow wave, $v_p < v_s \cos \theta$, so δP_B and δP_t will have the opposite sign.

Therefore Alfvén waves can be identified by the correlations between velocity and magnetic field perturbations, and the fast and slow waves can be identified by the negative (for slow waves) or positive (for fast waves) correlations between the thermal pressure and magnetic pressure perturbations. Even though the magnetospheric plasma are no longer homogeneous, these simple relations can still give us hints on the existence of ULF waves in a global simulation.

From the G8 flyby simulations, we do detect possible ultralow frequency (ULF) wave signatures on the magnetopause and in the upstream magnetosphere. We extract the states along sampled points along the Galileo G8 flyby trajectory, and study the correlations between perturbed quantities. Figure 4.12 shows the detection

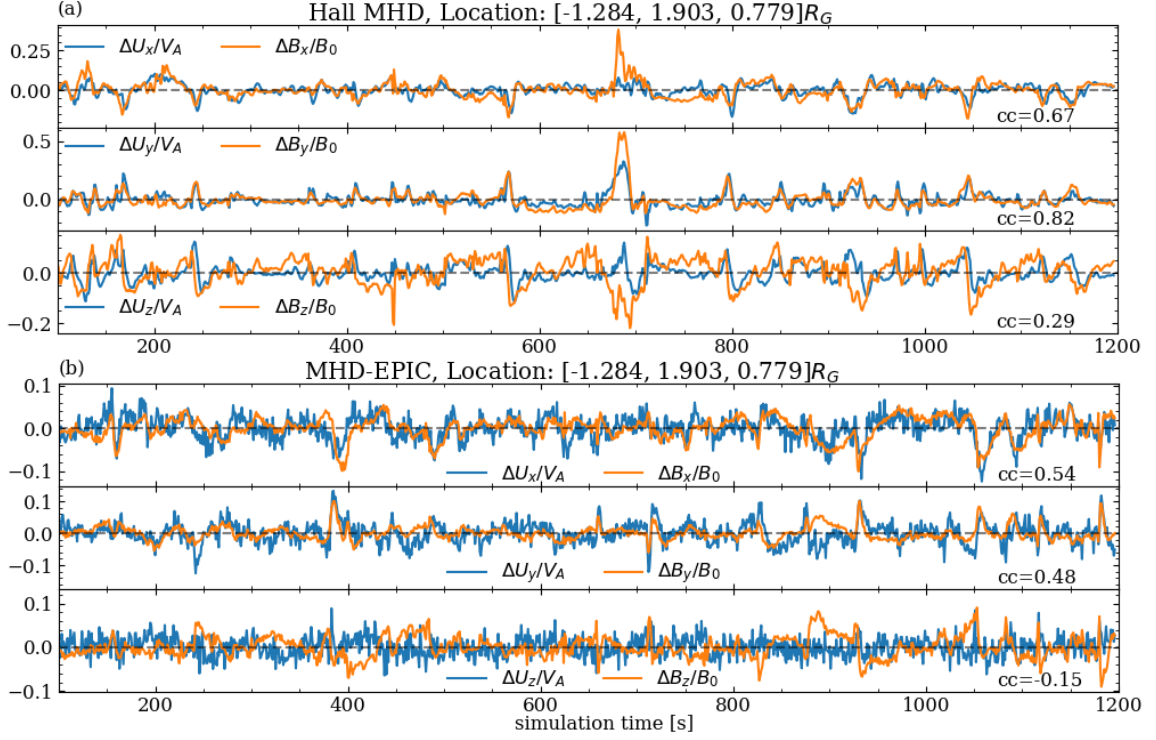


Figure 4.12: Alfvén wave detection with plasma velocity and magnetic field perturbations at static location $[-1.284, 1.903, 0.779]R_G$ outside the magnetosphere near the outbound crossing in (a) Hall MHD and (b) MHD-EPIC.

of Alfvén wave above the equatorial plane slightly outside the magnetopause near the outbound crossing. This is a location where the magnetic field mostly lies in the z direction, and we find quite strong positive correlations between the x and y components of the perturbed velocity and magnetic field. The normalized quantities also have roughly the same magnitude, which is what we expect from Equation 4.13. Thus it is highly likely to be Alfvén waves propagating upward from low latitudes. The influence of large flux ropes can also be seen in the static satellite perturbations inside or near the magnetosphere in Figure 4.12. During the intervals of large FTEs ($\sim 700s$ in Hall MHD, $\sim 400s$ in MHD-EPIC), the velocity and magnetic field perturbations deviate more from the linear correlations in Figure 4.12, which makes the ULF wave signatures less clear as in the quiet times. Further away from the magnetosphere, we get very clear signatures of fast wave mode shown in Figure 4.13a-

b with positively correlated thermal pressure and magnetic pressure perturbation. As for the slow wave mode, we find possible negative correlated signatures near the magnetopause crossings. Examples during the outbound crossing is shown in Figure 4.13c-d for Hall MHD and MHD-EPIC, respectively.

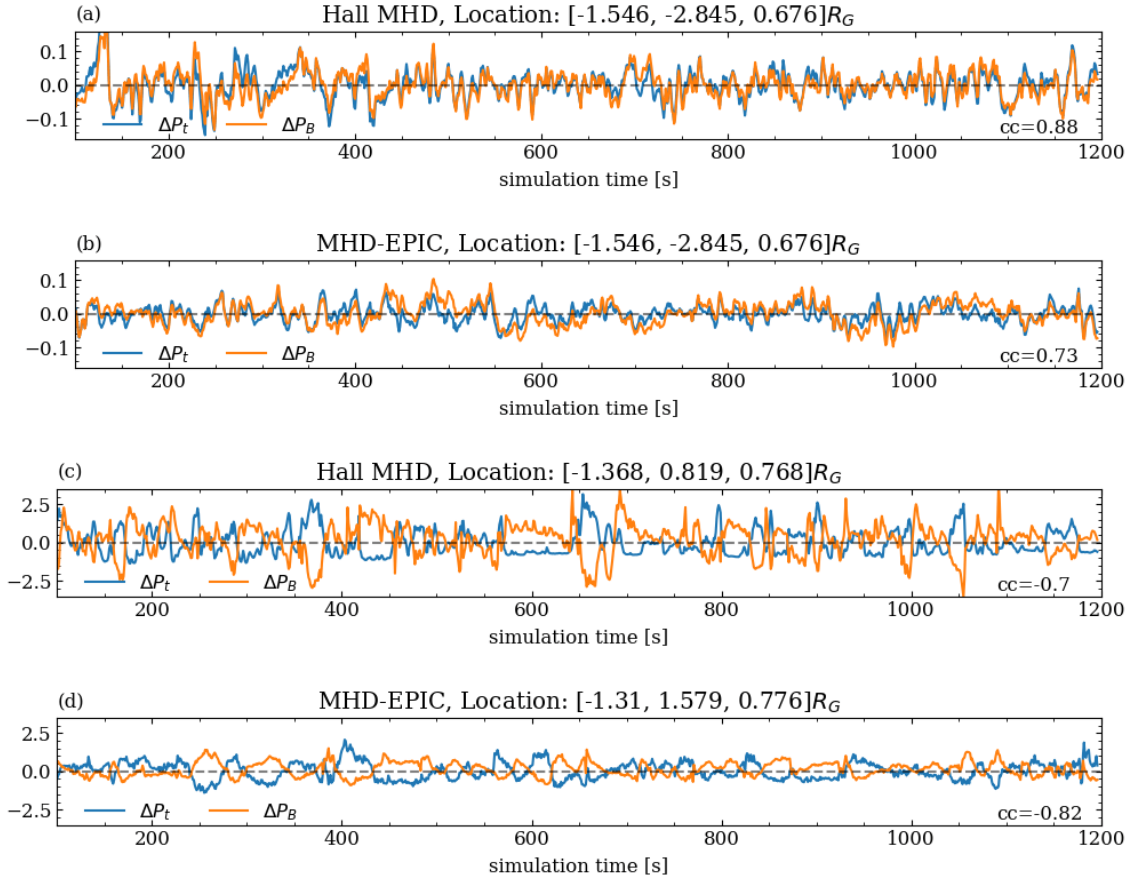


Figure 4.13: Fast (a-b) and slow (c-d) wave detection at static locations in GPhiO coordinates along the G8 flyby Galileo trajectory near Ganymede’s magnetosphere in Hall MHD and MHD-EPIC simulations. The plasma thermal pressure perturbations are shown in blue and the magnetic pressure perturbations are shown in orange. Location in (a-b) is on the anti-Jovian side outside of the magnetosphere; locations in (c-d) are near the outbound crossing. The correlation coefficient are marked on the bottom right for each panel.

The spatial distribution of the wave detection represented by the correlation coefficients of the perturbed quantities from MHD-EPIC simulation along the Galileo G8 flyby trajectory is summarized in Figure 4.14. Due to the fact that during the G8 flyby magnetosphere crossing the z coordinate changed very slowly, we choose

to plot the $z = 0.76R_G$ plane contoured with the total current density to represent the magnetosphere structure. From both Hall MHD and MHD-EPIC models, fast waves are clearly identified over a large range of locations outside the magnetosphere, while slow waves and Alfvénic perturbations are identified only at certain locations near the magnetopause crossings along the Galileo G8 flyby trajectory. This shows that even under constant upstream driving, the plasma interaction at the magnetopause can still trigger ULF waves. However, the correlation test between magnetic and velocity perturbation at static locations does not show evidence of shear Alfvén wave inside the magnetosphere, which contradicts the previously interpretation by [Kivelson et al. \[1998\]](#), [Volwerk et al. \[1999, 2013\]](#) of field line resonance from careful detrended spectrum analysis for the closest 3 min encounter.

The traditional way of studying field line resonance phenomena from numerical simulation is to enforce a periodic external pressure perturbation and track the magnetosphere responses. In our current simulation setup with fixed upstream conditions, it is a challenge to see the resultant shear Alfvén waves in the closed field line regions in two aspects: firstly the numerical diffusion caused by the order and type of schemes and grid resolution may damp the waves, and secondly there may be other physical processes not included in the current model that can diffuse away the energy.

4.5.2 Lower Hybrid Drift Instability

The correlation between magnetic pressure perturbation and thermal pressure perturbation is widely detected in the $z = 0.76R_G$ cut plane of the upstream magnetopause inside the PIC region. Figure 4.15 shows the perturbations in one snapshot $t = 106s$ from the MHD-EPIC run. We can see both the positive and negative correlated perturbations at different locations across the magnetopause. There are some

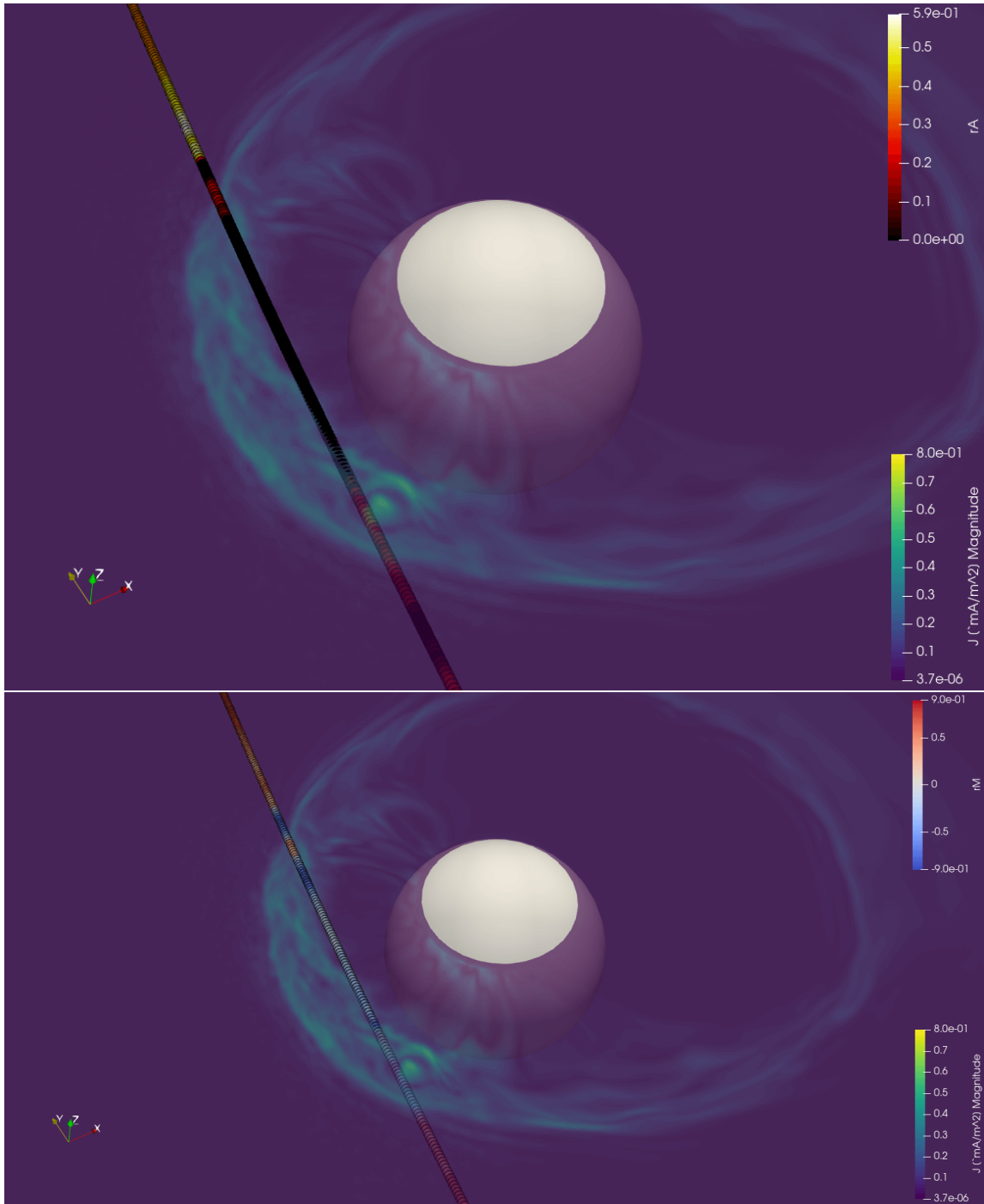


Figure 4.14: ULF wave distribution from MHD-EPIC simulation along the Galileo G8 flyby trajectory, colored with the linear correlation coefficient r_M between ΔP_t and ΔP_B and r_A between $\Delta \mathbf{V}$ and $\Delta \mathbf{B}$. A contour of the current density on the $z = 0.75R_G$ plane is overplotted for showing the magnetosphere.

interesting wave-like signatures on the positive y side of this plane with $kr_e \sim 0.4$, where $k = 2\pi/\lambda$ is the wave number, and r_e is the electron gyroradius. This lies in the regime of lower hybrid drift instability (LHDI). [Chen et al. \[2017\]](#) has reported the findings of LHDI in the dayside reconnection simulation of Earth under some scaling approximations. Whether or not LHDI can be found at Ganymede’s magnetosphere remains to be confirmed by future missions.

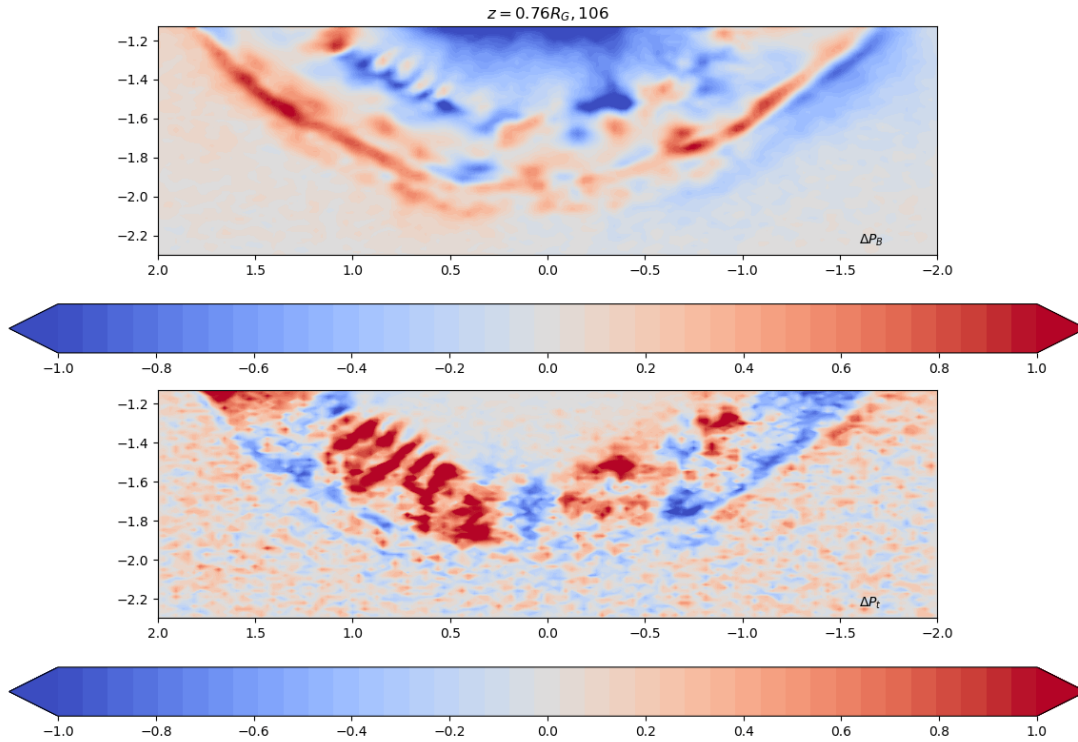


Figure 4.15: $t = 106$ s snapshot of normalized magnetic and thermal pressure perturbations at $z = 0.76R_G$ cut plane in the PIC region near the upstream magnetopause. The background pressure is taken as the average of ± 100 s time period.

4.5.3 Discussion

Recently [Carnielli et al. \[2019, 2020\]](#) used a test particle Monte-Carlo approach to build an ionosphere model for Ganymede that provides the spatial distribution of multiple ion species originating from Ganymede’s ionosphere. The magnetosphere models presented here used a relatively simplified approach to treating the iono-

sphere in that uniform, fixed plasma density and temperature are prescribed at the simulation boundary near Ganymede’s surface [Zhou et al. \[2019\]](#). In order to better understand the coupling between the magnetosphere and ionosphere, we may consider incorporating a realistic ionosphere model, such as that presented by [Carnielli et al. \[2020\]](#), into our global magnetosphere simulations in the future.

In the earlier study using a resistive MHD model with anomalous resistivity [Jia et al. \[2010\]](#), essentially the same FTE occurrence rate of 20-50 seconds was predicted as in the Hall MHD and MHD-EPIC models. These three different models all show that reconnection is non-steady under steady upstream conditions, and the characteristic timescale for FTE formation is on the order of tens of seconds. Putting all these results together does seem to suggest that this may be an intrinsic timescale to Ganymede’s magnetosphere dictated by the spatial size of the magnetosphere and the upstream plasma properties. However, developing quantitative relations still require further theoretical guidance and a series of carefully designed simulation runs to confirm.

The two models presented in this work predict a global reconnection efficiency of $\sim 60\%$ with flux ropes of $\sim R_G$ in length forming roughly 3 per minute at Ganymede’s upstream magnetopause under the conditions of the Galileo G8 flyby. Compared with the other Galileo flybys, G8 is the only one that occurred when Ganymede was inside Jupiter’s central plasma sheet. Outside of the central plasma sheet, the Jovian plasma density is usually smaller and the ambient magnetic field strength is larger, which result in smaller β and larger Alfvén velocity for the ambient plasma than for the G8 flyby. Because the ambient plasma and field conditions change periodically through each synodic rotation, it is of interest to examine how the properties of Ganymede’s magnetopause reconnection vary depending on the location of the moon

relative to Jupiter’s plasma sheet. We have performed simulations for other relevant scenarios with different upstream Alfvén Mach number and external field orientation. Results from our preliminary runs suggest that larger Alfvén velocity and/or larger magnetic shear at the magnetopause boundary tend to produce larger reconnection efficiency. Detailed investigation of the dependence of reconnection-driven dynamics on the upstream conditions is beyond the scope of this paper, but will be conducted in our future work.

Despite the similarities of many quantities between the two models, there are also some differences worth mentioning. While the total plasma pressures in the Hall MHD and MHD-EPIC models are about the same, the perturbations of pressure and the upstream reconnection rate and FTE counts are larger in the Hall MHD results under the grid resolution we use. [Tóth et al. \[2016\]](#) compared the Hall MHD simulations with two different resolutions and found that the solution became much more dynamic at high grid resolution ($\sim 1/64 R_G$) than the low grid resolution ($\sim 1/32 R_G$), while PIC behaves more similar between the two resolutions. There are several possible reasons for the differences between our Hall MHD and MHD-EPIC results.. The larger plasma pressure perturbations as well as reconnection rate perturbations in Hall-MHD come from the fact that Hall-MHD produces more patchy perturbations at the magnetopause than MHD-EPIC, especially in small scale oscillations. The small perturbations in PIC compared to Hall MHD can be the intrinsic feature of either the physical models or the numerical solvers (i.e. semi-implicit Hall-MHD solver and the GL-ECSIM solver). However, we have not done a systematic study on this effect. Additionally, the comparison of electron and ion pressures between the Hall MHD and MHD-EPIC models in Figure 4.2 clearly shows that Hall MHD cannot handle the energy partition/conversion between the two

species as in the kinetic PIC model, let alone the kinetic electron physics. The Hall MHD equations [Zhou et al. \[2019\]](#) contain no explicit terms for controlling electron pressure besides adiabatic heating.

From the particle distributions in phase space, we can see that kinetic physics only becomes important near the reconnection sites at the magnetopause boundary. In principle we can greatly speed up the simulation by embedding PIC regions only close to the magnetopause in the global Hall MHD runs. Many of the different measures for identifying the diffusion region are potentially useful for placing local PIC regions. However, this requires a more flexible configuration of the PIC domain, which will be the goal of future model development.

CHAPTER V

Hybrid Parallelization

The requirement of high fidelity numerical results promotes the development of massively hybrid parallelized models. In this chapter, we are going to present the parallel model development with BATS-R-US, including the incorporation of OpenMP with MPI [Zhou and Tóth, 2020] and the progress in the GPU-portable model.

5.1 OpenMP in BATS-R-US

One of the key features of BATS-R-US is its excellent scalability on supercomputers. Previous benchmarks [Tóth et al., 2012] with pure MPI parallelization have shown good strong scaling up to 8,192 cores and weak scaling up to 16,384 cores within the memory limit of the testing platform. However, the grid-related pre-calculated information replicated on every MPI process for simplifying the refinement algorithm have generated an unavoidable memory redundancy on computational nodes. To increase the scalability to even larger sizes, we need to reorganize the code and come up with a more advanced solution. As of now, for the existing processors targeted at supercomputers, most cores support at least two threads per core. For instance, the Xeon Phi X100 accelerator supports four threads per core; the IBM POWER9 core, which runs on Summit [Laboratory, 2020], comes in two variants: a four-way multithreaded one called SMT4 and an eight-way one called

SMT8. Future high performance computing architectures will inevitably support more hardware threads, where the well-designed hybrid parallel models will surely fit the hardwares and achieve optimal computing efficiency.

5.1.1 Hybrid Parallelization Strategy

BATS-R-US was originally designed for pure MPI parallelization and did not take advantage of the rapid development of shared-memory multithreading programming starting from late 1990s [Dagum and Menon, 1998, Chandra et al., 2001]. Even though MPI is generally observed to give better parallel scaling than OpenMP due to forced data locality, one obvious shortcoming of the pure MPI implementation is wasteful memory usage. In BATS-R-US, we support 1, 2, and 3 dimensional block-adaptive grids, where each block contains $n_1 \times n_2 \times n_3$ cells. Each block is surrounded by n_G layers of ghost cells. The topology of a 3D block-adaptive grid is described by an integer array with 18 integers per grid node (active nodes are at the leaves of the tree corresponding to the grid blocks) containing the following information: the current status (used, unused, to be refined, to be coarsened), the processor and local block indexes for active nodes, the minimum allowed, maximum allowed and current refinement levels, the three integer coordinates with respect to the whole grid, the global indexes of the parent and eight children nodes. This tree structure, together with a few more arrays related to load balancing and adaptive mesh refinement, is replicated on every MPI process, which simplifies the algorithms and saves inter-processor communication, but is clearly wasteful for current many-core computer nodes. If the total number of grid blocks reaches about 10 million, the total tree related storage will be close to 2 GB, so it will use up all the typical 2 GB/core memory available on the computing nodes. One option is to distribute the grid information over the MPI processes, but that will make the AMR algorithm

more complicated and potentially less efficient. Another option is to make the grid blocks larger (more cells per block), but that will make the AMR less flexible and require more grid cells to achieve the required resolution in the regions of interest. Our preferred approach is to use OpenMP multithreading. If there are N threads sharing the tree information per MPI process, then the fraction of node memory used to store the tree is reduced by a factor of N . On current architectures with many cores per node, this is a significant improvement. Of course, it is only acceptable, if the performance of the code remains close to that of the original pure MPI version.

There are other potential benefits of using hybrid parallelization. In some applications BATS-R-US uses large lookup tables with pre-calculated values for sake of efficiency (for equation of state, radiative cooling, opacity etc.). These tables can be quite large, and using shared memory access from multiple threads is clearly beneficial for saving memory use. Having fewer MPI processes for a given problem size may in principle improve communication (fewer messages are passed) and I/O performance (fewer files are used), but these potential benefits may or may not be realized and/or significant on a given computer platform.

The idea of hybrid programming with OpenMP and MPI arises naturally with the architecture of modern supercomputers, where each node containing several multi-core chips is connected through the network (e.g. InfiniBand). The potential advantages and challenges have been thoroughly described in the literature, e.g. [Smith and Bull, 2001, Drosinos and Koziris, 2004, Rabenseifner et al., 2009b]. With careful design of the code, researchers have found a relatively satisfactory performance on modern clusters [Jost et al., 2003, Jin et al., 2011].

There are four levels of communication supported by the current versions of MPI libraries with respect to thread safety:

1. Single: MPI calls outside PARALLEL regions only
2. Funneled: MPI calls on master thread only
3. Serialized: MPI calls on multiple threads serially
4. Multiple: concurrent MPI calls on multiple threads

For multi-threaded communication, different tags have to be used for distinguishing messages from different threads. We decided to use the default support level `MPI_THREAD_SINGLE` to minimize the work of rewriting the source code.

Two types of hybrid parallelization styles have been widely used [[Drosinos and Koziris, 2004](#)]:

1. the fine-grained model, which applies the multi-threaded parallelism on the innermost loops
2. the coarse-grained model, which applies the multi-threaded parallelism at a relatively high level

For BATS-R-US the two approaches correspond to the choice of parallelization over grid cells (fine-grained) or grid blocks (coarse-grained). As explained in detail above, BATS-R-US spreads the blocks among MPI processes and then loops through cells inside each block for calculating the face values, face fluxes and source terms for the update in each time step. We decided to follow the coarse-grained parallelization approach. A clear advantage is that there are fewer block loops than cell loops, so there are fewer directives to add and less overhead in entering/exiting the multi-threaded regions. In addition, the block loops cover a larger fraction of the computational cost than the cell loops, so we can expect better parallel performance. A potential issue with parallelizing the block loops is the complexity of the code within the parallel region, but this problem can be handled, as we will discuss below. In the end, we have

successfully added OpenMP parallel directives to most block-loops in BATS-R-US. The high level parallelized structure is shown in Figure 5.1.

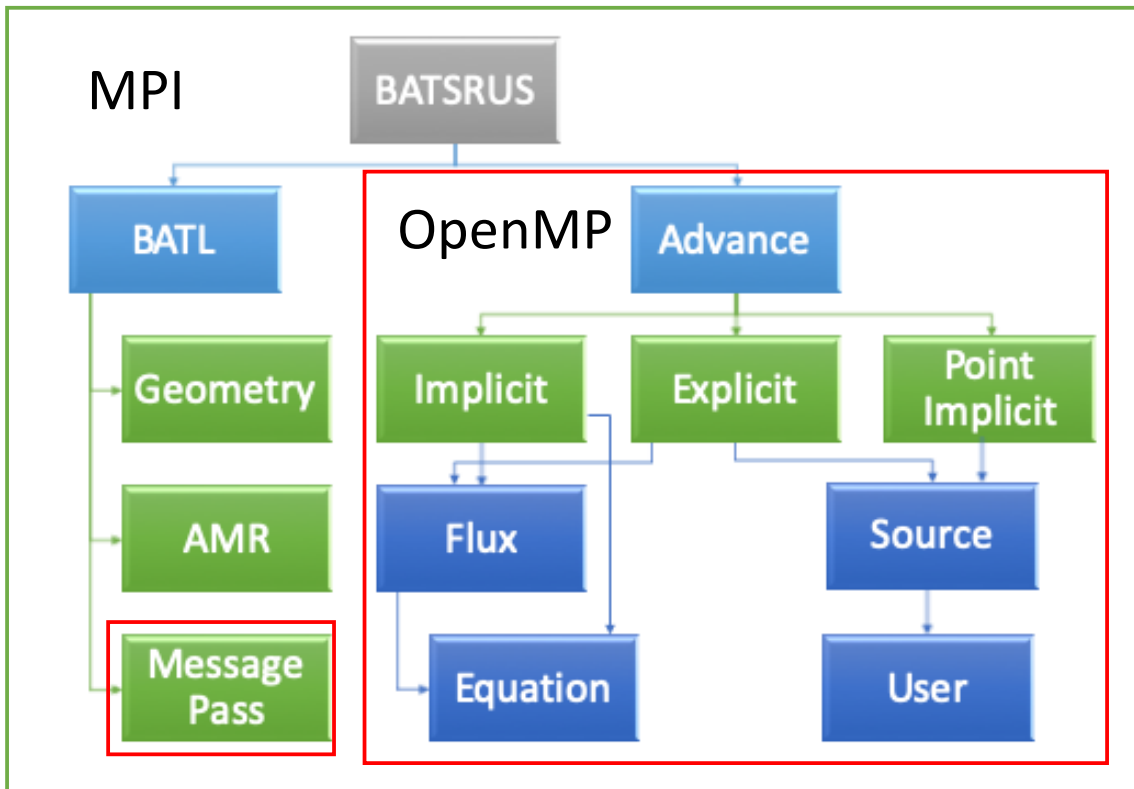


Figure 5.1: The high level structure of MPI+OpenMP BATS-R-US. The AMR library BATL is used for mesh generation, refinement and message passing. Explicit, fully-implicit, semi-implicit, part-implicit and point-implicit schemes can be used for time advance. Multiple versions of the equation and user modules containing the equation variable definitions and the application specific codes are available. The whole code is parallelized with MPI, while the region covered in red boxes are also parallelized with OpenMP.

Because of the functional programming structure of BATS-R-US, the usage of module variables in the multi-threaded regions requires special attention. Fortunately OpenMP provides the `threadprivate` directive to make global scope variables local and persistent to a thread through the execution of multiple parallel regions. This is a key feature of OpenMP that makes the hybrid parallelization possible for BATS-R-US without major modifications of the source code.

Similar hybrid approaches have also been implemented in block based AMR codes

like FLASH [ope, b], MPI-AMRVAC [Keppens et al., 2020], Athena [White et al., 2016], and GAMER2 [Schive et al., 2018]. FLASH is an AMR astrophysical code with OpenMP implemented for multiple solvers in both coarse-grain and fine-grain methods. In the small scale testings [ope, b], the coarse-grain approach consistently outperform the fine-grain approach. MPI-AMRVAC is written in Fortran with LASX preprocessor [Tóth, 1997], with coarse-grain multi-threading loop implemented in the explicit time advance, source term calculation, and error checking. Nice strong scaling with up to 2,000 cores and weak scaling with up to 30,000 cores has been shown for pure MPI runs. The latest version of Athena is written in C++, with multi-threading loop implemented mostly for the first dimension spatial loop in generalized coordinates among 80+ source files, and nice MPI scaling up to 6,144 cores. Finally, GAMER2 is a recently developed GPU-accelerated AMR code in astrophysics with MPI/OPENMP/GPU involved, with an impressive speedup compared with Athena++ [Schive et al., 2018]. It is also worth mentioning that a block based AMR library AMReX [Zhang et al., 2019b] natively supports MPI/OPENMP/OPENACC/GPU implementation. However, for all the mentioned codes above we have not found publications on their OpenMP performance. Another distinguishing feature of BATS-R-US is the parallel implicit time advance scheme. Here we present detailed performance results of the hybrid parallelization strategy for both explicit and implicit schemes up to hundreds of thousands of cores.

5.1.2 Implementation

Overview

There are two high level goals while modifying and improving the code:

1. Backward compatibility: the code should still work correctly and efficiently without the OpenMP compilation flag.

2. Minimize work effort and code changes as much as possible.

BATS-R-US is able to solve the system of partial differential equations with a mixture of explicit and implicit timestepping blocks distributed among the MPI processes. We treat first the explicit and then the implicit modules and add OpenMP directives incrementally to make sure the correctness and efficiency for each individual part. Since OpenMP does not allow controlling the scope of variables that are not explicitly listed in the function arguments, we have to rely on the default rules and settings about the variable scopes, especially for the variables in function and subroutine calls within the multi-threaded regions.

Current standard of the default OpenMP Fortran variable scope rules are listed as follows:

1. Variables explicitly showing up in the `omp parallel` section follow the common OpenMP rules for the shared or private attributes.
2. Variables with `save` attribute (e.g. module variables by default) are shared.
3. Variables being initialized at declaration time are shared.
4. Otherwise all variables are private.

Based on our choice of multi-thread parallelism on the block level, there is a simple yet general rule to decide whether a module variable should be declared as private inside the OpenMP parallel regions: *if an array does not have an index looping over blocks, it should be declared as threadprivate as long as there are writing operations.* Read-only variables do not need to be private to each thread. Note that Rule 3 listed above in practice gave us the most innocent looking bugs during testings. Dynamic threadprivate variables also need to be allocated for each thread, as opposed to only

being allocated for the main thread, otherwise this will produce "array not defined" run time errors.

In total we add 609 OpenMP directive lines to the 246,728 pure source code lines (excluding comments and empty lines), which is only 0.25% addition. These directives appear in declaring threadprivate variables in modules and subroutines, parallelizing block loops in both explicit and implicit schemes, reduction operations in error check and krylov iterative solver, and message passing. Together with the minor modifications and adjustments for the algorithms described below, the hybrid parallelization can be achieved with no extra tedious work.

Explicit Schemes

Due to the independency of blocks (with ghost cells) during one timestep update, it is rather straightforward to add multi-threading to the block loop over all the explicitly updated blocks. The kernel code is shown in List V.1. Note that we only present the kernel interfaces for brevity. At the final stage of each timestep, message passing is done through the BATL library for updating the ghost cell values.

```

1 STAGELOOP: do iStage = 1, nStage
2   ! Multi-block solution update.
3   !$omp parallel do
4   do iBlock = 1, nBlock
5     if(Unused_B(iBlock)) CYCLE
6     call calc_face_value(iBlock)
7     if(IsBoundaryBlock(iBlock)) &
8       call set_face_boundary(iBlock, Time_Simulation)
9     call calc_face_flux(iBlock)
10    call calc_source(iBlock)
11    call update_state(iBlock)
12    call calc_timestep(iBlock)
13  end do
14  !$omp end parallel do
15  if(iStage < nStage) call exchange_messages
16 end do STAGELOOP ! Multi-stage solution update loop.

```

Listing V.1: Explicit multi-stage update block loop pseudo code

All the high level functions (e.g. `calc_face_value`, `calc_face_flux`) are imported from their corresponding modules. Inside each module, the `threadprivate` clause is used to distinguish local variables for each block, e.g. the cell-based face

value arrays and the cell-based face flux arrays.

Implicit Schemes

Different kinds and combinations of implicit schemes are implemented in BATS-R-US. It provides the options of using point-implicit numerical scheme for handling stiff source terms and semi-implicit schemes for handling Hall and ohmic resistivity, radiative transfer and heat conduction terms. The point-implicit scheme follows a blockwise implementation which is similar to the explicit source term calculation in the sense of multi-threading. The semi-implicit schemes, which only part of the unknowns (e.g. magnetic field) are solved implicitly, can be treated as a special case for the full implicit schemes, therefore we will only discuss about the full implicit scheme parallelization. Lastly, the part implicit scheme means that only the unknowns in the selected blocks are updated implicitly, otherwise they are updated explicitly. [Tóth et al., 2006]

The mostly used implicit schemes in BATS-R-US convert the non-linear PDEs into a linearized form $\mathbf{A}\vec{x} = \vec{b}$ and solve iteratively in the Krylov subspace. Because of the explicit storage of variables based on the grid blocks, the implementation of full/part implicit schemes requires a transformation between the explicit and implicit storage. A common pattern in the original implicit solver module is a counter n inside the implicit block loop for indexing over the 1D arrays of the unknowns or the right-hand-side of the linear equation, where the backward-dependencies prevent a direct OpenMP parallelization. However, in most cases the indexes of the starting blocks can be pre-calculated directly from the grid-block structure, which then allows a block level multi-threading parallelization. A typical implicit block loop example is given in List V.2. Here after solving the unknowns \vec{x} from the the linear equations we use the pre-calculated initial value for the incremental counter n for each block outside the

cell and variable loops so that the threads can be performed independently. Besides, the block-based preconditioners can also benefit from multi-threading.

```

1 !$omp parallel do private( n )
2 do iBlock=1,nBlock
3   n = (iBlock-1)*nI*nJ*nK*nVar
4   do k=1,nK; do j=1,nJ; do i=1,nI; do iVar=1,nVar
5     n = n + 1
6     Impl_VGB(iVar, i, j, k, iBlock) = Impl_VGB(iVar, i, j, k, iBlock)
7     + x_I(n)*Norm_V(iVar)
8   enddo; enddo; enddo; enddo;
9 enddo
10 !$omp end parallel do

```

Listing V.2: Implicit block loop pseudo code: solution update. nI , nJ , nK , $nVar$ represent the number of cells in a grid block in the three spatial dimensions and the number of variables, respectively.

Profiling on the implicit solver suggest that it is not optimal to add OpenMP parallel regions as much as possible. Small loops with too few iterations won't benefit from OpenMP because of thread launching overhead; some intrinsic functions in Fortran for linear algebra already have built-in optimizations such that further multi-threading has no gain in timing. For instance, it is tempting to add `workshare` directive to operations like dot products in the Krylov iterative solver, but it turns out to have no improvement with such highly optimized Fortran intrinsic functions. We also avoided the usage of nested OpenMP parallel regions for loops over cells.

Message Passing

Message passing fills in the ghost cells of the blocks based on information taken from the neighboring blocks. With mesh refinement involved, there are three possible cases for communication with a neighboring block: 1. block at the same refinement level; 2. block at a finer refinement level; 3. block at a coarser refinement level.

The original message passing algorithm inside the BATL library can do both cellwise and facewise message exchange between layers of ghost cells and neighboring physical cells. Message passing on the faces are implemented for the adaptive grid where the numerical fluxes need to be conserved between the coarse and fine cell

faces on different resolution level. For most of the current applications, this part usually takes insignificant amount of time, so OpenMP directives are not added considering the amount of work of rewriting the code and the potential threading overhead detrimental to performance. In the case of cellwise message passing, ghost cells are filled with information from neighboring blocks. Our old scheme combines the local exchange (message passing between blocks on the same processor) and the remote exchange (message passing between blocks on the remote processors). This turns out to be difficult for adding efficient multi-threading parallelization, which is now rewritten to fully separate the local and remote part for an easy implementation of OpenMP and allow for possible overlap of communication and computation. The pseudo code is shown in List V.3. First the cell values to be communicated to remote MPI processes are collected into packed 1D send buffers. Next the MPI asynchronous communication functions `iSend/iRecv` are called. Then the *local* copying of cell values between blocks residing on the same MPI process are done in a multi-threaded loop. This is followed by the `waitall` barriers so that the communication overhead can be hidden by the time spent on the local copies. Finally the received linear buffer values are saved into the ghost cells of the blocks. This implementation only requires the thread safety level up to `MPI_THREAD_SINGLE` since the MPI calls occur outside the multi-threaded loop. Our performance tests presented below suggest that this simple strategy works satisfactorily, especially for the runs with more threads. Higher thread level support with more complex communication scheme could be added if this becomes a significant bottleneck in future applications.

```

1 ! Prepare the buffer for message passing to another processor
2 do iBlockSend = 1, nBlock
3     call message_pass_block(iBlockSend, DoRemote=.true.)
4 end do
5 ! Post sends
6 iRequestS = 0
7 do iProcRecv = 0, nProc-1
8     iRequestS = iRequestS + 1
9     call MPI_isead(...)

```

```

10 end do
11 ! Post requests
12 iRequestR = 0
13 do iProcSend = 0, nProc-1
14     iRequestR = iRequestR + 1
15     call MPI_irecv(...)
16 end do
17 ! Exchange ghost cell info on the same processor
18 !$omp parallel do
19 do iBlockSend = 1, nBlock
20     call message_pass_block(iBlockSend, DoRemote=.false.)
21 end do
22 !$omp end parallel do
23 ! Wait for all requests to be completed
24 if(iRequestR > 0) call MPI_waitall(...)
25 ! Wait for all sends to be completed
26 if(iRequestS > 0) call MPI_waitall(...)
27 ! Unpack the 1D buffer array to full variable arrays
28 call buffer_to_state

```

Listing V.3: Message passing demonstration

Data Locality

Current clusters rely heavily on the idea of Non-uniform Memory Access (NUMA). On Blue Waters, for example, the 32 core XE6 computing node contains two sockets, each with 2 8-core NUMA domain. Access time on shared memory in different NUMA domain/socket is significantly longer than within the same NUMA domain/socket, so we expect a performance drop from 8 to 16 threads and 16 to 32 threads on one node. On Stampede2 SKX compute nodes with 48 cores per node on 2 sockets, each NUMA domain consists of 12 cores. So similarly, we expect performance drop beyond 12 threads.

Timing Utility

The original timing library we used in BATS-R-US only works for single-threaded MPI runs. This new OpenMP extension of the code requires a modification of the current library. Now we have a new subroutine which only record the timings from the master thread for each MPI process. It is also possible, but not necessarily needed, to record the timings for each thread.

5.1.3 Testing and Debugging

Nightly Tests

For a comprehensive quality check and verification of the SWMF and the physics models contained (including BATS-R-US), we have built an automated **nightly test suite** for testing the code with various setups on various platforms. The latest version of the code is checked out from a central Git repository and 100+ tests are performed on multiple platforms with different compilers, compiler flags and number of cores. The test results are monitored every day and have been archived since 2009. These nightly tests were used heavily for the OpenMP development. The BATS-R-US tests helped us identify code features that did not perform the same way with and without OpenMP parallelization. Gradually more-and-more tests passed, and now OpenMP can be turned on and it passes for essentially all 100+ tests. Newly added code can work seamlessly as long as they carefully follow the Fortran variable scope standard listed above. This has to be verified, of course, by adding new functionality tests that cover the new code features to the test suite.

Resolving Race Conditions

Race conditions are very hard to debug. The code does not crash, but the results are incorrect, and may change randomly depending on the order of the thread executions. This can even happen when each thread uses the variable with the same values, but the value changes inside the multi-threaded region. An example, that demonstrates the issue is shown in Listing V.4.

```

1  subroutine calc_corotation_velocity(Xyz_D, uRot_D)
2     ! Calculates rotation velocity uRot_D at location Xyz_D
3     use CON_axes,           ONLY: get_axes
4     use ModCoordTransform, ONLY: cross_product
5     real, intent(in) :: Xyz_D(3)
6     real, intent(out):: uRot_D(3)
7
8     real, save :: Omega_D(3)
9     !$omp threadprivate( Omega_D )
10    logical :: IsUninitialized = .true.

```



```

11
12   if(IsUninitialized)then
13       call get_axes(TimeSimulation, RotAxisGseOut_D=Omega_D)
14       Omega_D = OmegaBody * Omega_D
15       IsUninitialized = .false.
16   end if
17   ! The corotation velocity is  $u = \Omega \times R$ 
18   uRot_D = cross_product(Omega_D, Xyz_D)
19 end subroutine calc_corotation_velocity

```

Listing V.4: Race condition example in subroutine

Although all threads perform the same calculation to obtain the angular velocity vector `Omega_D`, a race condition is still present because its value is first obtained with `get_axes` and then multiplied by `OmegaBody`, and different threads will go through these steps nearly, but not perfectly, simultaneously. Since the variable is declared with the `save` attribute in line 8, OpenMP will regard the variable as shared by default, so the value of the variable will be essentially random. One possible fix is adding the `!$omp threadprivate` directive in line 9 (an alternative would be adding a new variable for the rotational axis vector). Finding errors like this in hundreds of thousands of lines of source code spanning dozens of files is extremely challenging. We tried several different approaches (careful code inspection, checking which variable changes in a loop, etc.) but it became clear that we need help from a debugger. We tried several debuggers, but most of them either did not report the issues, or produced an overwhelming number of false alarms. In the end, we found the Intel Inspector to be an extremely useful tool for detecting thread-related issues. In our experience with Fortran, Inspector can successfully locate dead-locks, false sharing and race conditions. The race condition in the example problem above was found and fixed with the help of Intel Inspector.

5.1.4 Performance

We have performed some standard Brio-Wu MHD shock tube tests on various platforms. A 3D Cartesian grid is chosen, and dynamic AMR has not been employed.

The magnetic monopole is controlled by hyperbolic cleaning [Tóth et al., 2012] and 8-wave scheme [Powell et al., 1999]. Despite its simplicity, this is a fairly representative test for various applications in terms of computational cost per grid cell, as well as exercising the most important parts of the BATS-R-US code.

Platform Information

Table 5.1 lists all the platforms used for testing. We have performed parallel scaling studies on the Blue Waters Supercomputer using the XE nodes with AMD 6276 Interlagos processor and on the Stampede2 supercomputer using the SKX nodes with Intel Xeon Platinum 8160 ("Skylake") processor. [Stanzione et al., 2017].

Table 5.1: Platform Information

Name	Blue Waters	Stampede2
Processor	AMD 6276 Interlagos	Intel Xeon Platinum 8160
Sockets/Node	2	2
Cores/Node	32*	48
Threads/Core	1	2**
Clock rate (GHz)	2.3	2.1***
Memory (GB)	2	4
Fortran compiler used	gfortran	ifort

* The Interlagos processor has 8 Bulldozer cores, but viewed as 16 "processors" by the Linux system. These "processors" are the schedulable integer cores that work with the floating point unit.

** Hyperthreading is on by default.

*** 1.4-3.7GHz depending on instruction set and number of active cores, 2.1GHz is the nominal value.

Strong Parallel Scaling

Figure 5.2 shows the strong scalings on Blue Waters from 1 to 2048 nodes (32 cores/node). The MHD equations with hyperbolic cleaning [Tóth et al., 2012, Dedner et al., 2002] are solved with the second order explicit Linde scheme [Linde, 1998] and Koren's limiter [Koren, 1993] on a 3D Cartesian grid. The grid contains about 33.5 million grid cells in 65,536 grid blocks. Each block contains $8 \times 8 \times 8$ cells (with additional 2 layers of ghost cells on each side), which is a typical block size for our

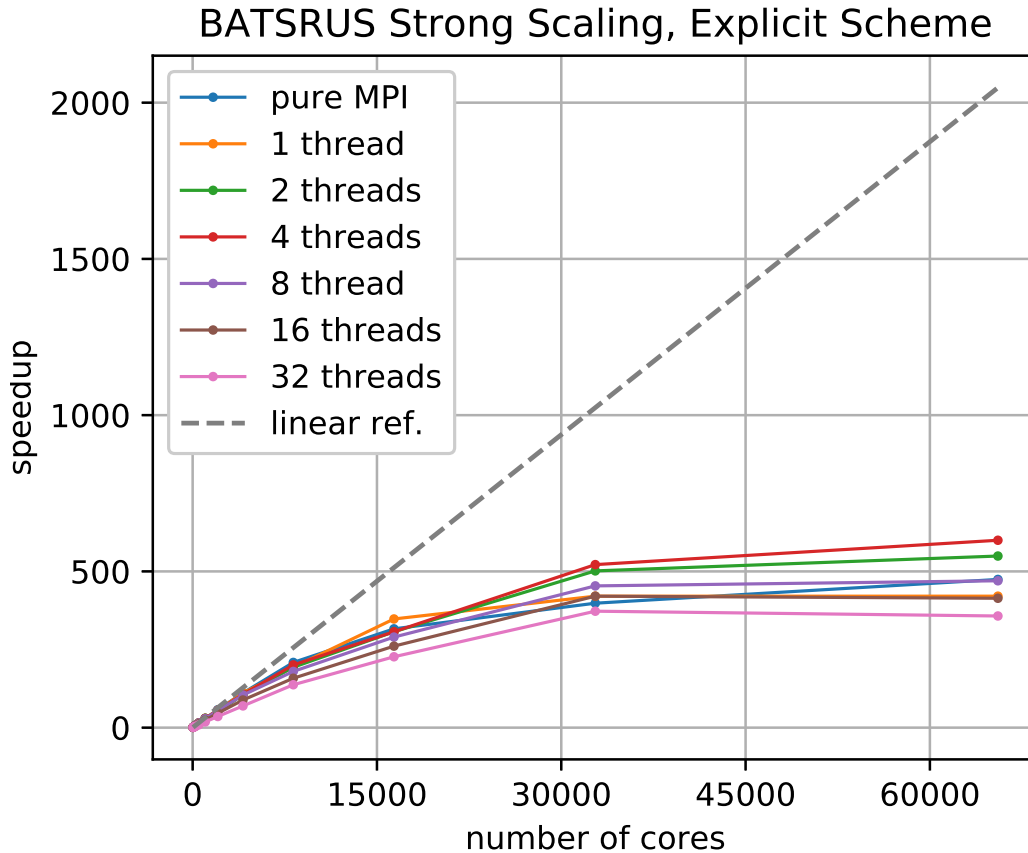


Figure 5.2: Hybrid strong scaling of BATS-R-US on Blue Waters. A 3D shocktube problem is solved by the second order explicit Linde scheme [Linde, 1998] on a uniform grid with a total of 33 million cells using $8 \times 8 \times 8$ blocks. Each computing node has two AMD 6276 Interlagos processors, and a single XE node has 32 “integer” cores. The x-axis represents the total number of physical cores used, and the y-axis represents the speedup compared to 1 node (32 cores) run compiled without OpenMP. The number of blocks per MPI process ranges from 2048 to 1 for single thread runs. The dashed line indicates ideal linear scaling. gfortran 7.3.0 is used for compilation.

common simulations. In the figure, the different colors represent different number of threads used in the runs. The points lying on the same vertical line have the same number of cores but a different combination of MPI processes and OpenMP threads. At 2048 nodes (65,536 cores), we only have 1 block per core and we obtain $\sim 25\%$ efficiency compared to the ideal linear scaling starting from 1 node shown by the dashed line. At 1024 nodes (32,768 cores), the parallel efficiency is around 50%. Over 1000 nodes, the multi-threaded runs show an advantage over the pure MPI

runs, but there is no apparent trend on the optimal number of threads being used. Based on the asymptotic behavior of the scaling curves, the parallel portion of the code is estimated to be $\sim 99.8\%$ both with pure MPI and hybrid MPI + OpenMP parallelization.

There is an interesting non-monotonic behavior of the speedup between different number of threads beyond 16,384 cores. For example, the 1 thread run performs the best for 16,384 cores but the second worst for 32,768 cores. The 2 threads and 4 threads run out-perform the rest for 65,536 cores. Given the same number of cores, more threads means more blocks per MPI process. The few number of blocks for the last few points on the lines may enhance the influence of random noises and block level overhead. Theoretically 4 threads should be the optimal choice given the NUMA domain architecture, as is the case shown beyond 30,000 cores.

Weak Parallel Scaling

Since our goal is to save memory with OpenMP parallelization, the weak scaling, with a fixed problem size per core, is the most relevant. We solve the same MHD problem as in the strong scaling test but now the number of grid cells is proportional to the number of cores and the available memory: there are 256 grid blocks with a total of 131,072 grid cells per core. BATS-R-US's capability of mixing explicit and implicit numerical schemes requires good scaling for both time stepping schemes. Here they are tested individually. For the explicit algorithm test, we select the second order Roe flux [Roe, 1981] with Koren's third order limiter [Koren, 1993]; for the implicit time stepping test, we select the second order Linde flux [Linde, 1998] with fixed (for sake of meaningful timings) 20 iterations of the BiCGSTAB [Van der Vorst, 1992] scheme. Figures 5.3 and 5.4 show the performances for the explicit and implicit schemes, respectively. We note that while the explicit runs have ~ 10

times more cell updates per second compared to the implicit runs, the explicit time step (limited by the CFL condition) is much smaller than the implicit time step. For a general problem, the performance of the implicit schemes highly depends on the stiffness of the equation, the preconditioners, and the iterative methods being applied. A series of runs with 1, 2, 4, 8, 16, and 32 threads are shown with different colored lines. Note that for a fair comparison we fix the number of blocks per MPI process per thread, $n_{\text{Block}}/(n_{\text{MPI}} * n_{\text{Thread}}) = \text{constant}$, and with only 1 thread (red line), 2 threads, and 4 threads we ran out-of-memory at 2^{15} , 2^{16} , 2^{17} cores and beyond, respectively. The left panel of Figure 5.4 shows that the 2-thread runs even beat the linear scaling calculated from the 1-thread base case.

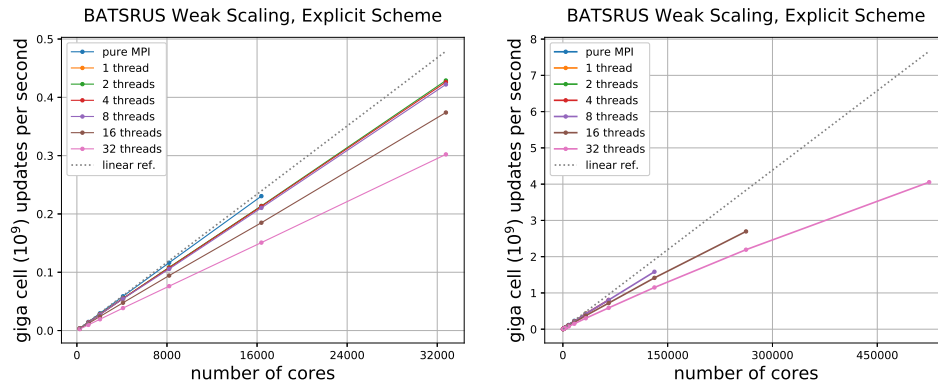


Figure 5.3: Hybrid weak scaling of BATS-R-US on Blue Waters. Left shows the cell updates per second for 1 to 32 threads runs up to 32,768 cores and right shows for the runs up to 524,288 cores. A 3D shocktube problem is solved by the second order explicit Roe scheme on a uniform grid with 131,072 grid cells per core using $8 \times 8 \times 8$ blocks. The number of blocks allocated per MPI process varies from 256 to 8192 depending on the number of threads. The dashed line indicates ideal linear scaling starting from 256 cores.

The right panels of Figures 5.3 and 5.4 show that the maximum number of cores the code can scale to strongly depends on the number of threads. The weak scaling is limited by the memory needed for storing the tree structure of the grid blocks, which is replicated for each MPI process to simplify the AMR algorithm. With 256 blocks per core, the total number of blocks reaches about 8.4 million when running the test

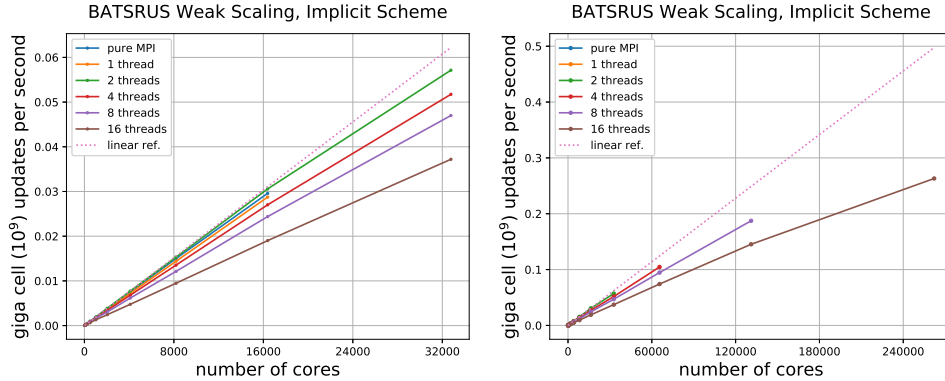


Figure 5.4: Hybrid weak scaling of BATS-R-US on Blue Waters. Left shows the cell updates per second for 1 to 16 threads runs up to 32,768 cores and right shows for the runs up to 262,144 cores. A 3D shocktube problem is solved by the second order implicit Linde scheme with BiCGSTAB method on a uniform grid. There are 131,072 grid cells per core, and the block size is $8 \times 8 \times 8$. The number of blocks per MPI process varies from 256 to 8192 depending on the number of threads. The dashed line indicates ideal linear scaling starting from 256 cores.

on 32,768 cores. Storing the corresponding tree information, including extra storage for the parent blocks, requires ~ 2 GB of memory for each MPI process. If the code is run with only one thread per MPI process, the tree information will exhaust the available 2GB/core memory of the XE nodes of Blue Waters, so we cannot scale beyond about 16 thousand cores with pure MPI parallelization. By taking advantage of the shared memory access allowed by multi-threading, we are now able to scale up to $\sim 5 \times 10^5$ and $\sim 2.5 \times 10^5$ cores with less than 50% loss compared to the ideal linear scaling with the explicit and implicit time stepping schemes, respectively. This means that more than an order of magnitude larger problem sizes have become doable thanks to the hybrid parallelization.

For the largest simulation we have done, the total number of grid cells is about 70 billion. With 9 output variables and single precision output format, the total size of data is about 2 TB. We exclude all the I/O related performance issues from this paper and leave them as a future target.

In both the explicit and implicit scheme tests, there is a significant performance

drop from 8 threads to 16 threads. This is probably caused by the connection speed bottleneck between NUMA domains for Blue Waters XE6 node architecture. There is another performance drop from 16 threads to 32 threads due to the connection speed between the sockets.

Performance Details

We took a closer look at the 256 core run from the scaling tests, and plotted the absolute and relative timings for the main modules in Figure 5.5 and 5.6. The *Advance* module is calling the other modules, so the timings for *Advance* include the timings for the other modules. Figure 5.5 shows that for the explicit scheme 8 threads with 32 MPI processes performs almost the same as 1 thread with 256 MPI processes, but 32 threads with 8 MPI is much worse, mainly because of the update solution check for reducing timesteps for vastly changing variables and the message pass for ghost cell fillings. The timings show that modules without global communication (*FaceValue*, *FaceFlux*, *Update*, etc.) scale very well, while modules with global communications (reduction in *UpdateCheck*, ghost cell filling in *MessagePass* etc.) scale poorly with more threads than what can fit on a NUMA domain.

Similar module performance for the implicit scheme is shown in Figure 5.6. As explained above, the current design of the iterative solver splits the block loops into several pieces, which requires individual multi-threading regions and unavoidable implicit barriers. The effect of thread launching overhead and barriers shows up even with 4 threads, and becomes quite significant with 16 threads.

Compiler and Platform Dependence

Many factors can influence the efficiency of the model besides the computational cost of the mathematical algorithm. The hardware specifications of different plat-

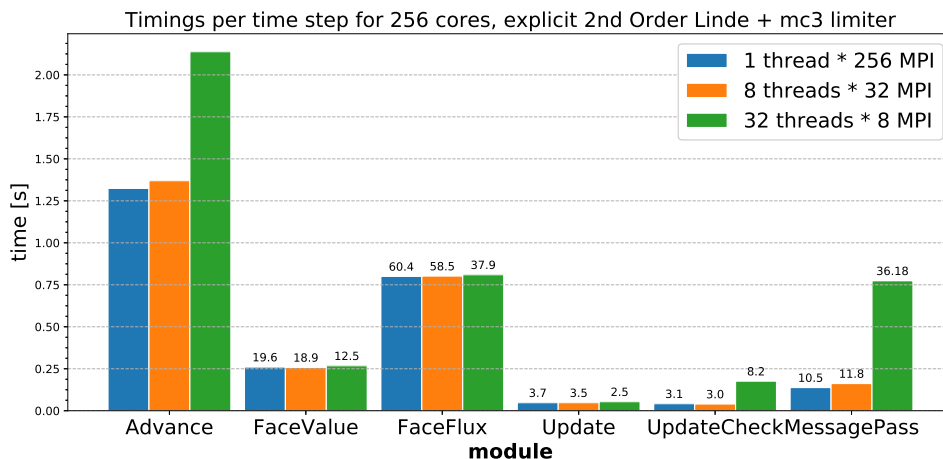


Figure 5.5: Timings for the Brio-Wu MHD shock tube test per time step on Blue Waters XE nodes. Second order explicit Linde scheme [Linde, 1998] is applied on a uniform grid with a total of 33 million cells using $8 \times 8 \times 8$ blocks. *Advance* represents the total time marching part of BATS-R-US, including all the other timings on the right; *FaceValue* is the reconstruction of left/right states; *FaceFlux* is the approximate Riemann solver; *Update* is the solution update; *UpdateCheck* is the timestep adjustment in case of vastly changing variables; *MessagePass* is the ghost cell filling. The numbers on top of the bar plot is the percentage each module takes in the total timings of the simulation.

forms and the compiling environments play a crucial role. As has been reported in the literature [Keyes et al., 2000], a typical CFD application utilizes less than 20% of the available computing power due to the heavy memory usage with relatively few computations done to each grid cell. With all the optimizations we have done so far, BATS-R-US can achieve 18% computing efficiency in our MHD test that is representative of many BATS-R-US applications.

Besides the Blue Waters Cray computer with AMD cores, we also tested the performance on Stampede2 with Intel Skylake cores. The results of weak scaling are shown in Figure 5.7. On this machine, the queuing system allows a maximum of 868 nodes per job, so we chose to go up to 512 nodes for the tests. As one can see from the comparisons, there is a much larger jump in performance between single and multi-threaded runs; then there is another jump when going beyond 12 threads, which is probably related to the NUMA domain architecture on Stampede2. Based on the

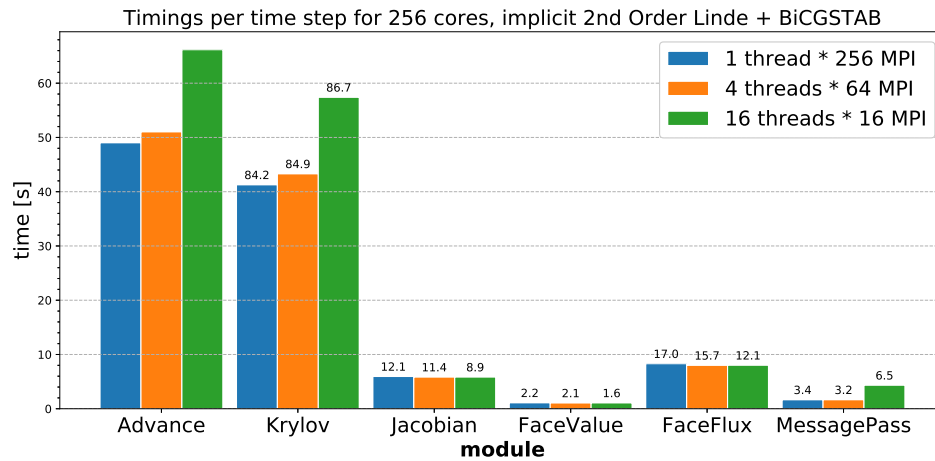


Figure 5.6: Timings for the Brio-Wu MHD shock tube test per time step on Blue Waters XE nodes. Second order implicit Linde scheme [Linde, 1998] with BiCGSTAB method is applied on a uniform grid with a total of 33 million cells using $8 \times 8 \times 8$ blocks. *Advance* represents the total time marching part of BATS-R-US. *Krylov* represents the iterative solver which contains part of the face value, flux calculations and ghost cell fillings. *Jacobian* represents calculating the preconditioner. The numbers on top of the bar plot is the percentage each module takes in the total timings of the simulation.

profiling results of each individual module of BATS-R-US, the most complicated face flux calculation and the message passing are degraded in performance as we increase the number of threads, while the timings for the less complicated face value and source term calculations remain more or less the same. This may indicate that the Intel Fortran compiler has some issues in optimizing a large multi-threaded parallel region: the inlining of multiple levels of function calls, vectorizations of the innermost loops, clever memory management within functions may be restricted by the compiler due to the complexity of the source code.

We also observed a bizarre and significant slowdown of the code when running on a single OpenMP thread compared to the runs compiled without OpenMP on the Blue Waters XE nodes and Stampede2 SKX compute nodes using the Intel Fortran compiler. Similar issues have also been reported elsewhere [Dinesh K. Kaushik and Smith, Rabenseifner et al., 2009a]. Different combinations of the processor and compiler also show different behaviors. For example, we have performed tests on

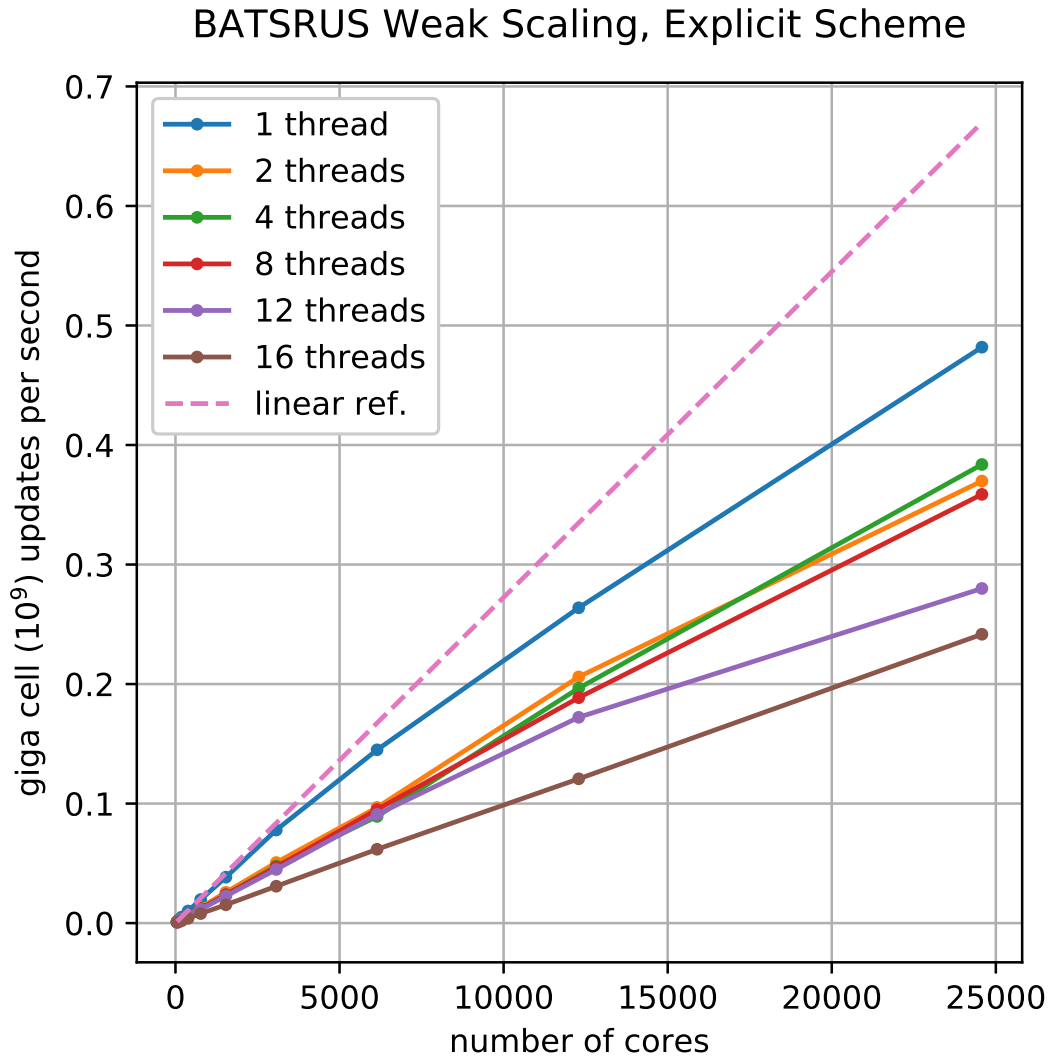


Figure 5.7: Hybrid weak scaling of BATSRUS on Stampede2 from 1 node (48 cores) to 512 nodes (24,576 cores). A 3D shocktube problem is solved by the second order explicit Roe scheme on a uniform grid with 131,072 grid cells per core using $8 \times 8 \times 8$ blocks. The number of blocks allocated per MPI process varies from 256 to 4096 depending on the number of threads. The dashed line indicates ideal linear scaling starting from 48 cores.

Blue Waters with the Cray, gfortran, ifort and pgf90 compilers. All four compilers has similar efficiency (within $\sim 15\%$ difference) when compiled without OpenMP, but there are drastic differences in timing when compiled with OpenMP. For the single thread runs, the Cray and gfortran compilers show almost no degradation in performance, while on the other hand ifort becomes ~ 2 times slower and pgf90 becomes ~ 3 times slower. This is an overlooked issue as the 1-thread case is often the baseline for all the hybrid MPI-OpenMP scaling tests, as opposed to pure MPI compilation.

To demonstrate the problem in a simple manner, we take the standard shock tube test from the overnight test suite, and plot the timings for a series of compilation setup on 3 different platforms in Figure 5.8. It scales well on the latest Frontera supercomputer with gfortran 9.0 and on our local Linux platform with gfortran 4.8, while significant degradation in performances are observed for the rest combinations.

Previous studies [[Dinesh K. Kaushik and Smith](#)] tended to blame the overhead of managing the threads, however, it is puzzling for us that this overhead would make a difference on the order of seconds for our test runs when the thread startup overhead is often claimed to be on the order of ~ 100 ms. Two side-by-side test runs on Stampede2 compiled without any optimizations with OpenMP flags on and off suggest that this overhead alone could cost a 20% slowdown, which is one of the four kinds of OpenMP overhead summarized in [[Lindberg](#)]. The difference is more significant with aggressive optimizations (-O2/-O3), and the slowdowns tend to relate to the problem size being solved. Sequential consistencies might be an issue in compiler optimizations. OpenMP defines consistency as a variant of weak consistency: it is not allowed to reorder synchronization operations with read or write operations on the same thread. In addition to that, a flush operation (which guarantees the con-

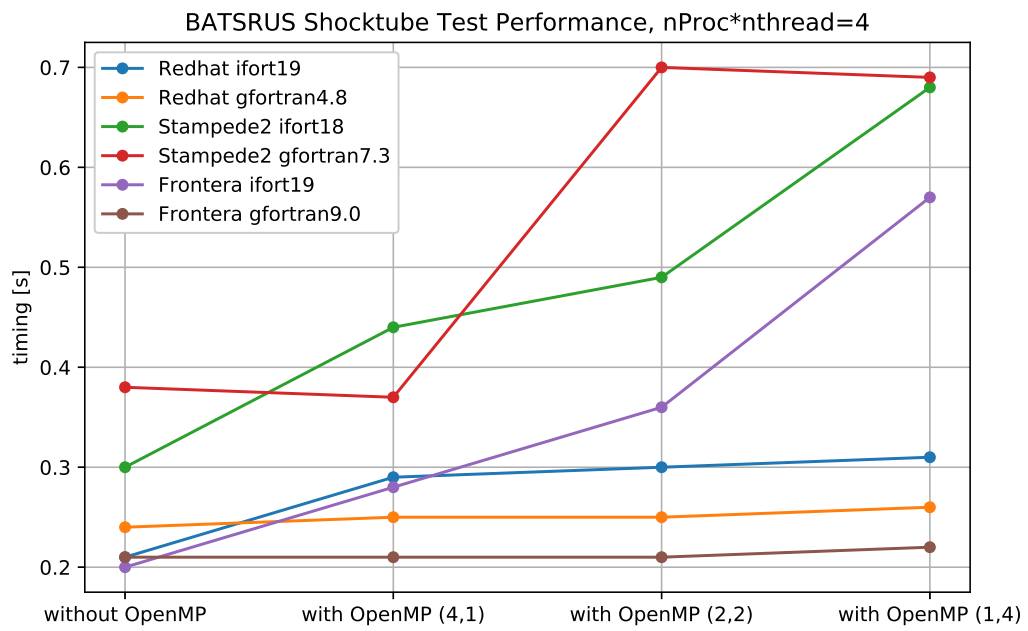


Figure 5.8: Comparisons of BATSURUS compiling with or without OpenMP. Four vertical lines represent 4 MPI compiling without OpenMP, 4 MPI and 1 thread, 2 MPI and 2 threads, and 1 MPI and 4 threads. Three testing platforms with different compiler choices are present: Redhat Linux desktop, Stampede2 supercomputer, and Frontera supercomputer.

sistencies of shared variables) is implied by OpenMP synchronizations at entry/exit of parallel regions, implicit/explicit barriers, entry/exit of critical regions, and locks. The detailed hidden implementations to ensure sequential consistencies for different compilers is a possible explanation to the performance discrepancies in our tests.

5.2 GPU Offloading

In recent years, there is a huge boost in GPU computing that comes with the success in algebraic-intensive areas. People came up with different strategies in taking advantage of GPUs. First, we can write CPU codes under certain rules and restrictions and convert them automatically to the GPU version [Benítez-Llambay and Masset, 2016]. This approach limits the style of CPU programming, and the efficiency from the auto-generated CUDA codes is usually less than the direct CUDA implementation. Secondly, we can write both the CPU and GPU version of code, such as [Schive et al., 2018]. Even as a popular choice, this will potentially lead to low productivity in the long run due to the redundancy in the code base, and the workload of maintaining both versions at the same time.

As an MHD code with a long history, BATS-R-US has more than 240,000 lines of pure Fortran source code (excluding comments and empty lines). We are looking for an approach that requires minimal coding effort while produces decent speedups, such as KOKKOS for C++ [Grete et al., 2019]. One tempting choice is the **OpenACC** (for open accelerators), which allows GPU offloading from the same code by switching on the directives. As in OpenMP, programmers can annotate C, C++ and Fortran source code to identify the areas that should be accelerated using compiler directives and additional functions, without writing specific CUDA code for the GPU. We have made some initial attempts to offload the computational-intensive work in BATS-R-

US to GPU through OpenACC.

As a starting point to run BATS-R-US on GPU, we developed a simple skeleton code that imitates the structure of the original code. The simplified version inherits the OpenMP pragmas from our previous hybrid model development, and the basic MPI functions but without actual communications. There are no real physical calculations for the cell values and face fluxes: we replace the computations with loops of square root operations to represent the different workload for different numerical schemes. The idea is to see if the current code is compatible with the OpenACC pragmas and estimate the potential problems we may encounter. Currently, PGI compiler is the only one available that provides full support for OpenACC. The test results are shown in Table 5.2. We ran the CPU version on a single core IBM POWER9 node and the GPU version on a NVIDIA Tesla V100 on Summit [Laboratory, 2020]. The speedup ratio of GPU runs over a single core CPU depends on the block sizes (n_{Cell}^3), the workload per cell (n_{Work}), and the number of blocks (n_{Block}). Given that each Summit node contains 2 POWER9 CPUs and 6 V100 GPUs (1 CPU and 3 GPUs per socket), and each of the POWER9 CPUs have 21 physical cores, each with 4 hardware threads, the speedup ratio must exceed $7/28$ with/without multithreading to gain the efficiency advantage. GPU architecture favors more homogeneous computations, with typically larger block sizes and more work per cell. If there are not enough computations to be done, the GPU code won't have the efficiency advantage over the parallel CPU code.

In the future, these preliminary results can shed light upon the continuing development of GPU-portable BATS-R-US. As promised in the OpenMP 5.0+ standard [ope, a], all the currently available OpenACC features will be incorporated into the OpenMP, with support from all major computer hardware and software vendors.

We can easily transfer to the OpenMP standard and take advantage of the flexible offloading options.

	nCell*	nWork**	nBlock	time/nIter	speedup**
CPU	8	10	100	0.0124	
GPU	8	10	100	0.0095	1.3
CPU	8	10	1000	0.125	
GPU	8	10	1000	0.094	1.3
CPU	8	10	10000	1.27	
GPU	8	10	10000	0.95	1.3
CPU	8	50	100	0.0886	
GPU	8	50	100	0.0104	8.5
CPU	8	100	10	0.0198	
GPU	8	100	10	0.0013	15.2
CPU	8	100	100	0.1978	
GPU	8	100	100	0.012	16.5
CPU	16	10	10	0.0091	
GPU	16	10	10	0.0012	7.6
CPU	16	10	100	0.0914	
GPU	16	10	100	0.0105	8.7
CPU	16	100	10	0.1494	
GPU	16	100	10	0.0014	106.7
CPU	16	10000	1	0.2168	
GPU	16	10000	1	0.0026	83.4
CPU	32	10	10	0.07	
GPU	32	10	10	0.002	35.0
CPU	32	10	100	1.98	
GPU	32	10	100	0.022	90.0
CPU	32	100	1	0.116	
GPU	32	100	1	0.00037	313.5
CPU	32	100	10	1.16	
GPU	32	100	10	0.0023	504.3
CPU	64	100	10	9.133	
GPU	64	100	10	0.012	761.1

* Number of cells per dimension in the 3D setup.

** Number of square root operations inside the kernel for imitating real computation.

*** Speedup over single core CPU using IBM POWER9 for CPU and Tesla V100 for GPU.

Table 5.2: Timings for the skeleton BATSRUS for porting to GPU with OpenACC

CHAPTER VI

Visualization and Data Analysis Package: VisAna

In order to process the observation and simulation data easily and efficiently, we need better tools for visualization and data analysis. Because our ideas keep changing, for productivity reasons, people often rely on dynamic programming languages for data processing instead of static languages. This chapter describes novel ways of efficient data processing, especially within the Space Weather Modeling Framework (SWMF).

6.1 Brief History of Dynamic Programming Languages

Programming languages can be classified into compiled and interpreted languages. While compiled languages such as C/C++ and Fortran are heavily used for the kernels of numerical physics models for speed, interpreted languages like IDL, Matlab and Python are more widely used in data processing and visualization because of their portability and productivity.

IDL has a long history of application in the space physics community since the 1980s. Matlab, which is another popular generic high level programming language known for linear algebra, is also commonly used in science and teaching. In recent years, we have seen an overwhelming popularity of Python in all categories thanks to its simplicity and readability. However, when it comes to numerical computing

and big data analysis, Python itself as a “glue” language often shows its lack of performance. The common tricks for speeding up the process in high level interpreted languages are:

1. to provide APIs to lower level libraries and functions written in C/C++ or Fortran and compiled into machine codes;
2. to incorporate Just-In-Time (JIT) compilation technique for converting scripts into bytecodes or machine codes to a certain degree.

Early attempts [[Behnel et al., 2011](#), [Foundation](#)] have shown that the first approach is not trivial to implement from a programmer’s perspective, and the second approach is hard to achieve for an existing language that is not designed with the ideas in mind.

This is where the latest competitor, Julia [[Bezanson et al., 2017](#)], comes into play. As a new language born for numerical computing and high performance, it has gradually gained attention from the scientific world and now unveils its potential to machine learning and generic programming. The key feature of Julia is its type inference system that allows a generic program to generate efficient machine code from LLVM [[LLVM](#)] for specific types with multiple dispatch [[Wiki, b](#)] at run times. As the founders of Julia claimed, this will be a fresh new approach to tackle the “two-language” problem, which is a trade-off that developers typically make when choosing a language — it can either be relatively easy for humans to write, or relatively easy for computers to run, but not both.

As a fun fact, the languages built with scientific programming in mind from the start, such as Mathematica, R, Matlab, Fortran and Julia, share one thing in common: they are all 1-based indexing languages.

6.2 VisAna: A Community Package for Simulation Data Processing

The open source **VisAna** package [Zhou, 2020] is written completely in Julia and provides functionalities of

- simulation data reading
- visualizing
- data format converting
- language interoperability
- data analysis in space physics

The data loader **SWMF.jl** inherits the ideas and code structures from the IDL post-processing scripts developed by Gábor Tóth and my original Matlab scripts. It currently supports all the output formats from SWMF, with benchmarks comparing with **SpacePy**, IDL, and Matlab shown in Figure 6.1. Taking advantage of the multiple dispatch design, the same code base works for all types of floating point precision. Note that in the Matlab version, all the final outputs are converted to double precision for satisfying the requirement of some intrinsic functions, which potentially increases the timings. In Julia there is no such restriction.

For example, one typical binary output file at a given snapshot from the iPIC3D 3D output for Ganymede’s magnetosphere simulation with grid resolution of one electron inertial length is about 2.4 GB. Julia performs ~ 5 times faster than IDL and ~ 10 times faster than Matlab, which is a significant boost of processing data. Further advantage from JIT compilation will be gained by manipulating the data as the user wants afterwards, and quickly exploring ideas and revising codes. Note that the comparison here neglects the JIT compilation time during the first time

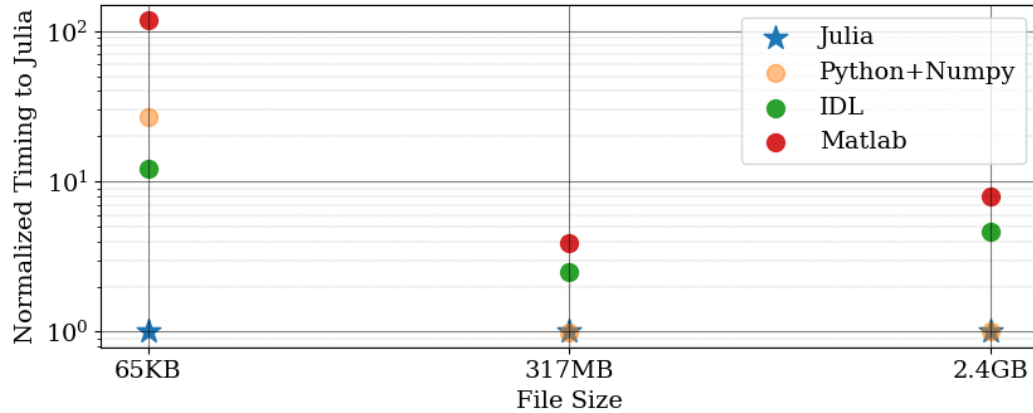


Figure 6.1: Benchmarks for the data loading of three different file sizes in KB, MB, and GB, respectively. All the timings are normalized to the Julia execution time.

execution. In practice we only need fast performance for large scale repeated tasks, where Julia is obviously apt for.

6.2.1 Field Tracing

Field lines are a family of curves that are instantaneously tangent to the vector of the field. They show the direction in which a massless element will travel at any point in time. In CFD and MHD simulations, streamlines/field lines are heavily used to visualize velocity and electromagnetic field structure and identify connectivity.

Currently two types of stream tracing methods are implemented in VisAna:

1. tracing in a uniform grid,
2. tracing in an unstructured grid.

In a uniform grid, it is straightforward to find the nearby cell-center values around the current position. A bilinear interpolation is used together with 2nd order forward Euler and 4th order Runge-Kutta time stepping as described in Algorithm 1.

Demonstrations of the 2D uniform grid tracing are shown in Figures 6.2 and 6.3 for an analytic asymptotic field and dipole field, respectively. Corresponding fixed timesteps are chosen according to the grid resolution. RK4 produces more accurate

Algorithm 1: Field tracing in uniform grid

Result: An array of points representing the trajectory

```

1 Create unit vectors from full vector field
2 for  $i = 1:maxstep$  do
3   Find surrounding points
4   if leave the domain then
5     Break
6   end
7   Interpolate unit vectors to current location
8   if detect NaNs in function values then
9     Break
10  end
11  Update location
12 end

```

tracing than the 2nd order Euler scheme under the same resolution and timestep. Therefore it is chosen as the default scheme unless efficiency is needed for large amounts of data.

Tracing in the unstructured grid output, e.g., the `.plt` format from AMR simulations, requires a different approach. Besides relying on ParaView or Tecplot's built-in function through Python API calls, VisAna provides a basic function for quickly scanning through data. The brute force approach is outlined in Algorithm 2.

Algorithm 2: Field tracing in unstructured grid, brute force

Result: An array of points representing the trajectory

```

1 find the grid cell you are currently in
2 for  $i = 1:maxstep$  do
3   move along the vector direction until you hit the edge of that cell
4   find the neighbour cell who shares the same edge
5   use the vector direction in the next cell and move along that direction
6   if reach MaxLength or reach domain boundary then
7     Break
8   end
9   Update location
10 end

```

This simple scheme may not perform as bad as one think. Finite volume method intrinsically only have one constant value in each cell, so tracing in this way reflect

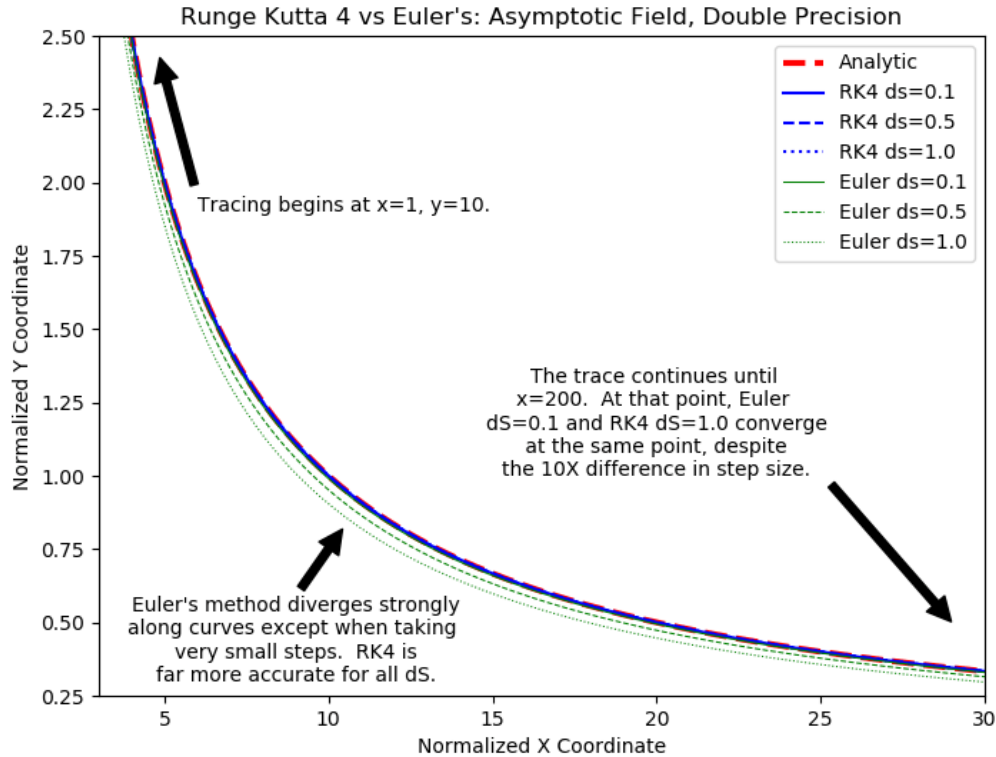


Figure 6.2: Demonstration of stream tracing in an asymptotic field. Integration using RK4 and forward Euler method is colored in blue and green, respectively. Different line styles represent different timesteps. The analytic solution is colored in red.

the true discrete numerical solution. Given that the purpose of this tracing is not only for visualization, but also for checking the connectivity of field regions, e.g., the open/closed magnetic field lines in certain locations, the brute force algorithm fulfills the need adequately.

Demonstrations of the 2D unstructured grid tracing are shown in Figure 6.4. Comparing the analytically defined stream function denoted by the blue lines with the stream tracing output of connected dots as footprints inside each cell.

Both of these tracing schemes can be extended to 3D. The tracing on the uniform grid in 3D, on one hand, is a natural extension to the 2D functions, which has been already implemented but not fully tested. The tracing on the unstructured grid in

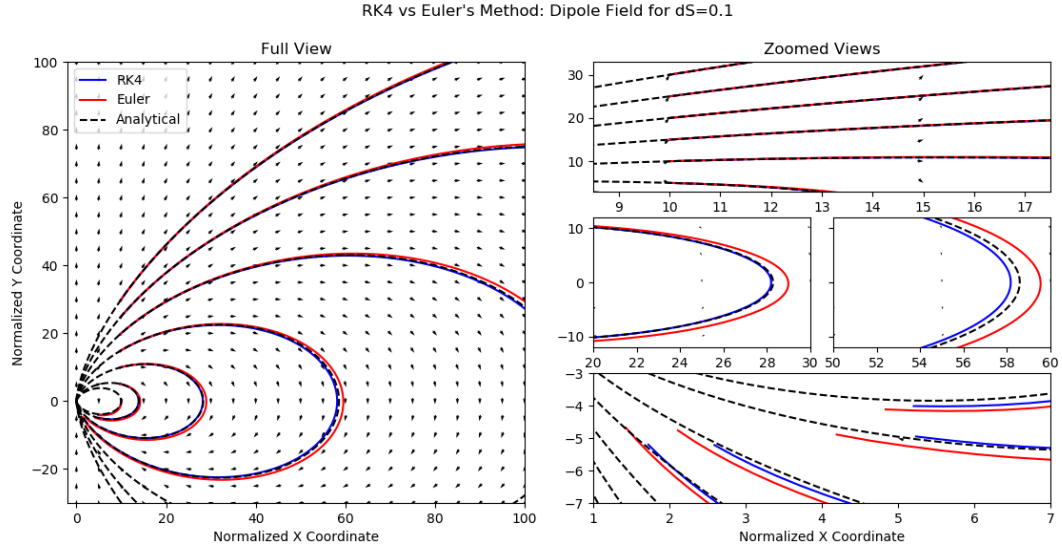


Figure 6.3: Demonstration of stream tracing in a dipole field originates at $(0,0)$. Integration using RK4 and forward Euler method is colored in blue and red, respectively.

3D, on the other hand, requires the support of a robust unstructured grid type and some geometry detection scheme in 3D, which is still a work in progress.

6.2.2 Particle Tracing

Particle tracing aims at solving the motion of particles according to Newton's second law. In MHD and PIC simulations, we have

$$\frac{d\mathbf{x}_P(t)}{dt} = \mathbf{v}(\mathbf{x}_P(t), t) \quad (6.1)$$

$$\frac{d\mathbf{v}(\mathbf{x}_P)}{dt} = \frac{q_P}{m_P} \left(\mathbf{E} + \mathbf{v}(\mathbf{x}_P) \times \mathbf{B} \right) \quad (6.2)$$

where \mathbf{x} and \mathbf{v} are the particle position and velocity, respectively, m is the particle mass, q is the particle electric charge, and \mathbf{E}, \mathbf{B} are the EM field vectors. The subscript P indicates that we are following the motion of a particle.

The particle tracing done in the post-processing phase is usually known as the *test particle simulation*. It is different from the field tracing in that particles have mass and the field may not be steady. It is also different from the self-consistent PIC simulation because particles do not affect the EM field, under the assumption that a

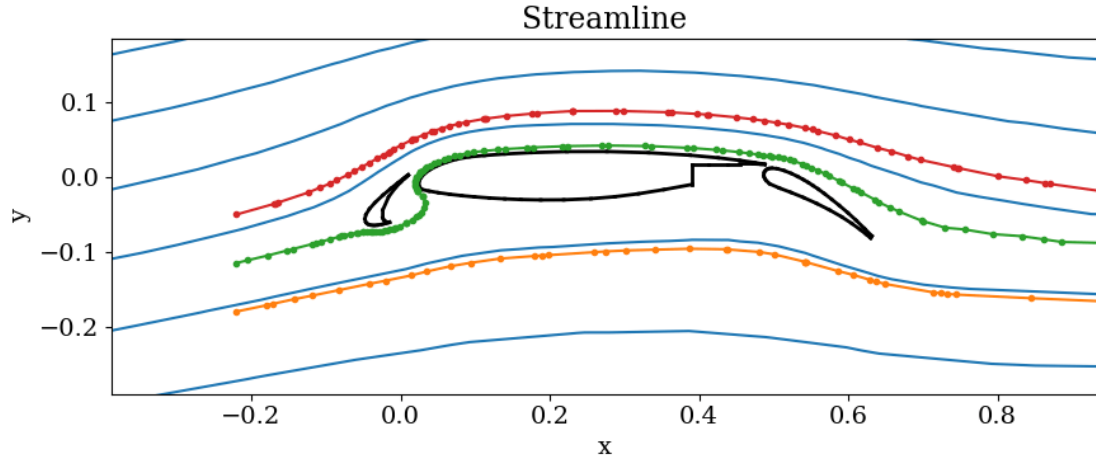


Figure 6.4: Demonstration of stream tracing around an airfoil in an unstructured triangular grid. The three traced lines are shown with colored lines with dots representing the footprints in each cell. The solid blue lines are contours from the analytically defined stream functions, which are calculated numerically.

small number of test particles have tiny effects on the field structure. The resulting \mathbf{x}_P forms the pathline, which describes the trajectory of the traced particle.

Mathematically this is an ordinary differential equation (ODE) problem that can be solved numerically. Thanks to the fast development of ODE solver in Julia [Rackauckas and Nie, 2017], it can be easily achieved with proper choice of scheme. Figure 6.5 shows an example of 50 MeV ion trajectory traced in the Earth’s dipole field, which reflects the three basic motions of charged particles: gyration around the magnetic field line, bounce along an L shell, and drift in the azimuthal direction. Tsitouras 5/4 Runge-Kutta method [Tsitouras, 2011] is used for this tracing.

6.2.3 File Format Conversion

One common task in scientific computing is the conversion between file formats for specific visualization purposes. For historical reasons, SWMF supports Tecplot `.plt` format, but not the open source VTK formats that is commonly used for ParaView [Ayachit, 2015] and VisIt [Childs, 2012]. VisAna provides a useful functionality of

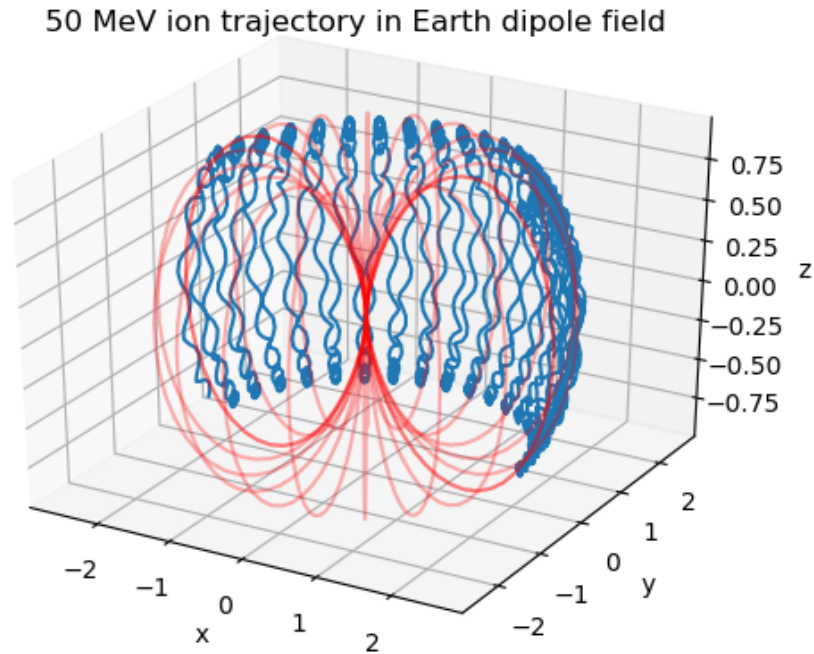


Figure 6.5: Demonstration of particle tracing inside Earth’s dipole field. The magnetic field lines are shown in red. Trajectory of an ion with 50 MeV energy is shown in blue. The lengths are normalized to Earth radius $R_E = 6378$ km.

converting ASCII and binary outputs from SWMF into structured and unstructured VTK formats with compression. The actual compression ratio depends on the original data, with ~ 5 times smaller than the ASCII output and ~ 2 times smaller than the binary output. The generated VTK files can then utilize the advanced techniques in ParaView and VisIt for rendering and 3D visualization. One example is shown in Figure 4.14.

Besides, the AMR tree file generated by BATL can be accessed by VisAna, which allows further optimizations of block-based AMR visualization.

6.2.4 Parallel Execution

Post-processing for current massively parallel numerical models commonly involves more than 100 GB of data. Unlike physical models which require frequent information change between processors, data analyses are usually embarrassingly parallel jobs, which means there are few inter-dependencies between different parts of data. VisAna makes use of the current available parallelism of Julia to boost the efficiency. Coroutines, multi-threading and multi-core features are implemented at different levels for large data processing. Our tests show that even though all patterns work for a small number of cores (< 10), channels work more stable for larger numbers of cores due to its minimum usage of memory. For example, the Ganymede MHD-EPIC high resolution simulation generates 1200 VTK unstructured files of size 2.4GB each. Every single output file takes about 20s to trace field lines and plot contour surfaces. In serial execution this will take about 6.7 hours on Frontera supercomputers. With parallel channels, all the work can be finished within 30 mins.

6.2.5 Programming Language Interoperability

Besides doing work solely in Julia, VisAna offers convenient ways to call functions from Python, Matlab and IDL through multiple external packages. Future development can easily take advantage of the existing fruitful packages and functions among multiple commonly used languages. The similar syntax and several script auto-conversion tools also help in quickly building an efficient workflow.

6.2.6 Test Tracking

To aid the tracking and analysis of SWMF simulations, VisAna provides basic support for extracting timing information from runlog files. Sustainable model development requires frequent testing. Based on the SWMF [overnight test suite](#), we

deploy a script for requesting runlog data online, extracting pass/failure information and timings, and visualize the results in a time series plot. Users can conveniently select the testing time range, type, and platform in the input parameter file and compare the timings, which is useful for debugging and code optimization. An example monitoring output of the MHD shock tube test is shown in Figure 6.6.

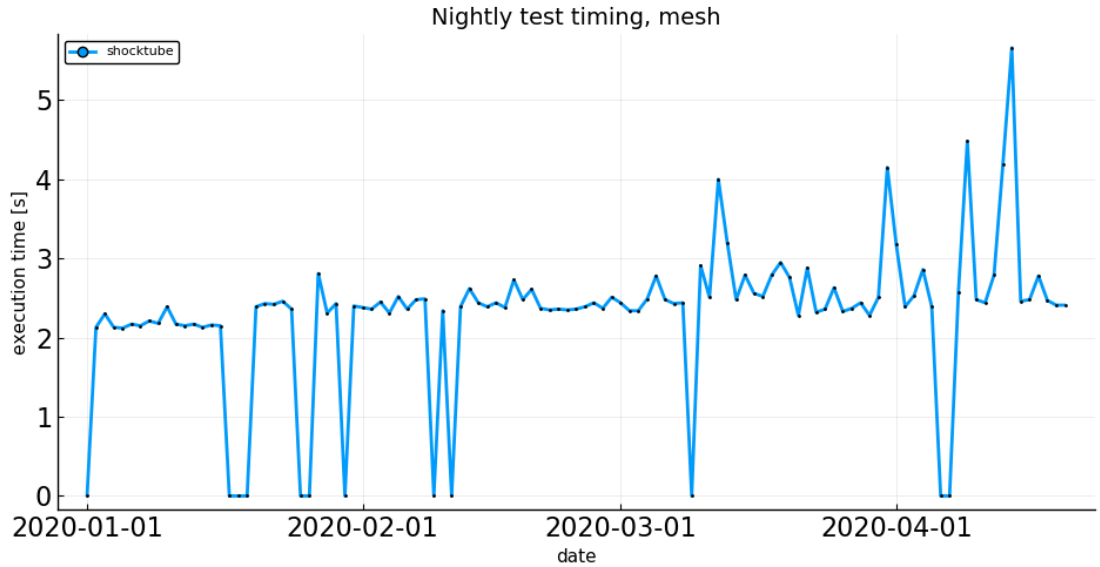


Figure 6.6: Profiling of MHD shock tube test from the [overnight test suite](#) during the period 2020-01-01 to 2020-04-01.

More Features

Currently one big missing part of VisAna is the built-in unit system and the conversion between SI, Gaussian, planetary, and user defined units. With the increasing need of efficient data processing and visualization, the flexibility and great performance of Julia-based packages will attract more attention in the space physics numerical computing community.

CHAPTER VII

Conclusions

7.1 Summary of Results

7.1.1 Embedded Kinetic Simulation of Ganymede's Magnetosphere

In the first study of the thesis, we have presented an improved MHD-EPIC model of Ganymede's magnetosphere. With an adaptive stretched spherical MHD grid, an incorporated magnetic diffusion solver down to the core region, and a new energy-conserving PIC scheme, the simulations show a dynamic magnetosphere with reconnection-related features. Key findings from this set of simulations are:

1. Single fluid MHD-EPIC model is able to capture the global structure of the magnetic field in Ganymede's sub-Alfvénic magnetosphere, shown by the quasi-steady comparison over all six Galileo flybys. Intermittent FTEs are generated even under steady upstream conditions, signatures of which can be potentially captured by in-situ magnetometer observations.
2. Ion-scale physics including non-gyrotropic and non-isotropic effects are fully resolved in the PIC domain covering the majority of the upstream magnetopause. Crescent-shape ion distribution is found on the magnetospheric side of the upstream reconnection.
3. The energetic flux densities from upstream reconnection obtained directly from

the PIC simulation are about 10^{-7} W/cm². A quantitative estimation on the auroral emission shows that upstream reconnection can contribute for up to about half of the peak brightness.

4. The G2 flyby comparisons of plasma bulk flow velocities with the PLS data support the interpretation that the observed plasma outflow from Ganymede's ionosphere is mainly composed of oxygen ions.

7.1.2 Reconnection-driven Dynamics at Ganymede's Upstream Magnetosphere

In the subsequent Ganymede's magnetosphere study, we have presented the results and predictions from Hall MHD and MHD-EPIC simulation of upstream reconnection dynamics. We find that under steady upstream conditions, magnetopause reconnection occurs in a non-steady manner. Flux ropes of about $1 R_G$ in length form on the magnetopause at a rate about 2/minute and produce spatiotemporal variations in plasma and field properties. Upon reaching grid resolution comparable to the electron inertial length, the MHD-EPIC model can resolve both electron and ion kinetics at the magnetopause and show localized non-gyrotropic behavior inside the diffusion region. We have developed a general and robust method to calculate the global reconnection rate that works for a highly dynamic reconnection process as present in Ganymede's upstream magnetosphere. The estimated global reconnection rate from the models is about 80 kV with 60% efficiency, and there is weak evidence of about 1 minute periodicity from the global reconnection efficiency fluctuation from the simulations.

The global Hall MHD and MHD-EPIC simulations presented in this paper allow us to study in detail how magnetic reconnection occurs at Ganymede's upstream magnetopause. Our simulation results provide predictions regarding the unsteadiness

of reconnection, generation of FTEs, and the particle and field characteristics of the diffusion region around the X-lines. These predictions can be tested through and also be used to interpret new observations from future space missions, especially the upcoming Jupiter Icy Moon Explorer (JUICE) mission [[Grasset et al., 2013](#)].

7.1.3 Hybrid Parallel MHD Model Development

The requirement of high fidelity numerical results stimulates the development of massively hybrid parallelized model. We have successfully extended our finite volume/difference MHD code BATS-R-US from pure MPI to MPI+OpenMP hybrid implementation, with only 0.25% modification to the $\sim 250,000$ lines of source code. Good weak scaling performances are obtained up to $\sim 500,000$ cores with explicit time stepping and up to $\sim 250,000$ cores with implicit time stepping. Using the hybrid parallelization, we are now able to solve problems more than an order of magnitude larger than before thanks to the usage of shared memory for large grid arrays. We opt to use coarse-grained multithreading applied to the loops over grid blocks, because it provides more opportunity for parallelism than the fine-grained approach applied to loops over grid cells. The main challenge with the coarse-grained approach in a large and complex code is to find and eliminate race conditions, which can be solved with well-organized code structure and the help of debuggers like Intel Inspector.

We also found that the compiler's capability to efficiently optimize a multi-threaded code varies significantly from compiler to compiler on various platforms. It is important to check the performance of the single-threaded execution of the code compiled with and without the OpenMP library.

7.1.4 Data Visualization and Analysis

To suit the needs of processing large data from simulations, I have developed an open source, high performance and user friendly post-processing package VisAna in the Julia programming language. A whole new workflow of space physics-related data processing and analysis incorporating the latest techniques in computer science has been established. VisAna provides the capabilities of visualizing, parallel data processing, file format conversion, and function calling from different programming languages through the open source community support.

7.2 Future Work

7.2.1 Ganymede's Magnetosphere Modeling

Even though the magnetic field comparison between simulation and Galileo observation data is fairly good, our MHD-EPIC model clearly cannot reproduce every detailed feature observed by Galileo. Most of the discrepancies appear in the magnetopause crossings, and in general the simulation results look more smoother than observations. This may indicate that the second order schemes we applied to MHD and PIC are not fully capable of capturing the sharp changes at the feasible grid resolution, or the prescribed boundary conditions, especially the inner boundary conditions, cannot faithfully represent the influence coming from the interior. In the future, the nonsymmetric resistivity distribution, a putative ocean with high conductivity located between 150 km and 250 km depth [Kivelson et al., 2002, Saur et al., 2015] and new knowledge about the interior structure may intrigue the development of a more realistic resistivity model. Recently [Carnielli et al., 2019, 2020] used a test particle Monte-Carlo approach to build an ionosphere model for Ganymede that provides the spatial distribution of multiple ions species originating from Ganymede's

ionosphere. The magnetosphere models presented in this thesis research used a relatively simplified approach to treating the ionosphere in that uniform, fixed plasma density and temperature are prescribed at the simulation boundary near Ganymede's surface [Zhou et al., 2019]. In order to better understand the coupling between the magnetosphere and ionosphere, we may consider incorporating a realistic ionosphere model, such as that presented by Carnielli et al. [2020], into our global magnetosphere simulations in the future.

Additionally, because of the intrinsic differences between Hall MHD and PIC models, a coupled PIC region produces different results from the Hall MHD model. As can be seen in Figure 3.5a and Figure 3.6a, the magnetic field comparisons are not as good as in the stand-alone Hall MHD simulations (Figure 3.4e,f). From plasma observations [Kivelson et al., 2004], we know there are at least two ion populations in the ambient Jovian environment: thermal and hot components. For a single fluid model like we have shown in this thesis, we have to make assumptions in prescribing a single upstream ion temperature that results in a total particle pressure consistent with the observations. A more realistic simulation left for future work should be separating the ion populations using a multi-fluid MHD model as well as introducing more particle species inside the PIC region.

We may also include an even more consistent coupling between multi-fluid MHD, or even higher moments MHD, and the PIC model. From the particle distributions in phase space, we can see that kinetic physics only becomes important near the reconnection sites at the magnetopause boundary. Therefore in principle we can greatly speed up the simulation by embedding PIC regions only close to the magnetopause in the global Hall MHD runs. Many of the different measures for identifying the diffusion region (e.g. Figure 4.7) are potentially useful for placing local PIC regions

with physics-based criteria. However, this requires a more flexible configuration of the PIC domain, which will be the goal of future adaptive PIC model development.

Compared with the other Galileo flybys, G8 is the only one that occurred when Ganymede was inside Jupiter’s central plasma sheet. Outside of the central plasma sheet, the Jovian plasma density is usually smaller and the ambient magnetic field strength is larger, which results in smaller β and larger Alfvén velocity for the ambient plasma than for the G8 flyby. Because the ambient plasma and field conditions change periodically through each synodic rotation, it is of interest to examine how the properties of Ganymede’s magnetopause reconnection vary depending on the location of the moon relative to Jupiter’s plasma sheet. We have performed simulations for other relevant scenarios with different upstream Alfvén Mach number and external field orientation. Results from our preliminary runs suggest that larger Alfvén velocity and/or larger magnetic shear at the magnetopause boundary tend to produce larger reconnection efficiency. Detailed investigation of the dependence of reconnection-driven dynamics on the upstream conditions is beyond the scope of this paper, but will be conducted in our future work.

7.2.2 Hybrid Parallelization

Our massively-parallel model development using hybrid MPI+OpenMP reported in Chapter V excludes the updates on parallel I/O, which is potentially important in practice. In the current version of BATS-R-US, each MPI process writes its own files, and the individual pieces are collected together during the post-processing phase. Such strategies don’t scale well and are likely to stress the HPC file system like Lustre by burdening the metadata servers because of the number of files created. A better approach is the genuinely parallel MPI-based I/O, with MPI processes collectively read/write to a single target.

The GPU-offload attempt as described in Chapter 5.2, can be a useful guidance for the next step parallel code development. OpenMP 5.0+ provides potential capability of seamlessly converting existing OpenACC directives to the OpenMP compatible version. By cleaning up the legacy modules and coming up with cleaner interfaces between BATL, the numerical solver and the equation sets, we will be able to build the cross platform version of BATS-R-US.

Bibliography

- Space weather modeling framework. <http://csem.engin.umich.edu/>. Accessed: 2020-05-20.
- Openmp api specification: Version 5.0. <https://www.openmp.org/spec-html/5.0/openmp.html>, a.
- Multithreaded flash. http://flash.uchicago.edu/~jbgallag/2012/flash4_ug/node42.html#SECTION01033100000000000000, b. Accessed: 2019-11-07.
- JD Anderson, EL Lau, WL Sjogren, G Schubert, and WB Moore. Gravitational constraints on the internal structure of ganymede. *Nature*, 384(6609):541, 1996.
- K Arzner and M Scholer. Kinetic structure of the post plasmoid plasma sheet during magnetotail reconnection. *Journal of Geophysical Research: Space Physics*, 106(A3):3827–3844, 2001.
- Maha Ashour-Abdalla, Jean P Berchem, Jörg Büchner, and Lev M Zelenyi. Shaping of the magnetotail from the mantle: Global and local structuring. *Journal of Geophysical Research: Space Physics*, 98(A4):5651–5676, 1993.
- Nicolas Aunai, Michael Hesse, and Maria Kuznetsova. Electron nongyrotropy in the context of collisionless magnetic reconnection. *Physics of Plasmas*, 20(9):092903, 2013.

- Utkarsh Ayachit. *The paraview guide: a parallel visualization application*. Kitware, Inc., 2015.
- Stefan Behnel, Robert Bradshaw, Craig Citro, Lisandro Dalcin, Dag Sverre Seljebotn, and Kurt Smith. Cython: The best of both worlds. *Computing in Science & Engineering*, 13(2):31–39, 2011.
- Pablo Benítez-Llambay and Frédéric S Masset. Fargo3d: a new gpu-oriented mhd code. *The Astrophysical Journal Supplement Series*, 223(1):11, 2016.
- Naoki Bessho, L-J Chen, and Michael Hesse. Electron distribution functions in the diffusion region of asymmetric magnetic reconnection. *Geophysical Research Letters*, 43(5):1828–1836, 2016.
- Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98, 2017.
- Bruce G Bills. Free and forced obliquities of the galilean satellites of jupiter. *Icarus*, 175(1):233–247, 2005.
- JU Brackbill and DW Forslund. An implicit method for electromagnetic plasma simulation in two dimensions. *Journal of Computational Physics*, 46(2):271–308, 1982.
- AL Broadfoot, BR Sandel, DE Shemansky, JC McConnell, GR Smith, JB Holberg, SK Atreya, TM Donahue, DF Strobel, and JL Bertaux. Overview of the voyager ultraviolet spectrometry results through jupiter encounter. *Journal of Geophysical Research: Space Physics*, 86(A10):8259–8284, 1981.
- JM Broll, SA Fuselier, and KJ Trattner. Locating dayside magnetopause recon-

- nection with exhaust ion distributions. *Journal of Geophysical Research: Space Physics*, 122(5):5105–5113, 2017.
- JL Burch and TD Phan. Magnetic reconnection at the dayside magnetopause: Advances with mms. *Geophysical Research Letters*, 43(16):8327–8338, 2016.
- Gianluca Carnielli, Marina Galand, François Leblanc, Ludivine Leclercq, Ronan Modolo, Arnaud Beth, Hans LF Huybrighs, and Xianzhe Jia. First 3d test particle model of ganymede’s ionosphere. *Icarus*, 330:42–59, 2019.
- Gianluca Carnielli, M Galand, François Leblanc, Ronan Modolo, A Beth, and X Jia. Constraining ganymede’s neutral and plasma environments through simulations of its ionosphere and galileo observations. *Icarus*, page 113691, 2020.
- Michael H Carr, Michael JS Belton, Clark R Chapman, Merton E Davies, Paul Geissler, Richard Greenberg, Alfred S McEwen, Bruce R Tufts, Ronald Greeley, Robert Sullivan, et al. Evidence for a subsurface ocean on europa. *Nature*, 391(6665):363–365, 1998.
- PA Cassak and MA Shay. Scaling of asymmetric magnetic reconnection: General theory and collisional simulations. *Physics of Plasmas*, 14(10):102114, 2007.
- Gaël Cessateur, Jean Lilensten, Mathieu Barthélémy, Thierry Dudok de Wit, Cyril Simon Wedlund, Guillaume Gronoff, Hélène Ménager, and Matthieu Kretschmar. Photoabsorption in ganymede’s atmosphere. *Icarus*, 218(1):308–319, 2012.
- Rohit Chandra, Leo Dagum, David Kohr, Ramesh Menon, Dror Maydan, and Jeff McDonald. *Parallel programming in OpenMP*. Morgan kaufmann, 2001.

- Yuxi Chen and Gábor Tóth. Gauss's law satisfying energy-conserving semi-implicit particle-in-cell method. *Journal of Computational Physics*, 386:632–652, 2019.
- Yuxi Chen, Gábor Tóth, and Tamas I Gombosi. A fifth-order finite difference scheme for hyperbolic equations on block-adaptive curvilinear grids. *Journal of Computational Physics*, 305:604–621, 2016.
- Yuxi Chen, Gábor Tóth, Paul Cassak, Xianzhe Jia, Tamas I Gombosi, James A Slavin, Stefano Markidis, Ivy Bo Peng, Vania K Jordanova, and Michael G Henderson. Global three-dimensional simulation of earth's dayside reconnection using a two-way coupled magnetohydrodynamics with embedded particle-in-cell model: Initial results. *Journal of Geophysical Research: Space Physics*, 122(10):10–318, 2017.
- Yuxi Chen, Gabor Toth, Xianzhe Jia, James Slavin, Weijie Sun, Stefano Markidis, Tamas Gombosi, and Jim Raines. Studying dawn-dusk asymmetries of mercury's magnetotail using mhd-epic simulations. *arXiv preprint arXiv:1904.06753*, 2019.
- Hank Childs. Visit: an end-user tool for visualizing and analyzing very large data. 2012.
- Glyn Collinson, William R Paterson, Christopher Bard, John Dorelli, Alex Gloer, Menelaos Sarantos, and Rob Wilson. New results from galileo's first flyby of ganymede: Reconnection-driven flows at the low-latitude magnetopause boundary, crossing the cusp, and icy ionospheric escape. *Geophysical Research Letters*, 45(8): 3382–3392, 2018.
- Leonardo Dagum and Ramesh Menon. Openmp: An industry-standard api for shared-memory programming. *Computing in Science & Engineering*, (1):46–55, 1998.

- Lars KS Daldorff, Gábor Tóth, Tamas I Gombosi, Giovanni Lapenta, Jorge Amaya, Stefano Markidis, and Jeremiah U Brackbill. Two-way coupling of a global hall magnetohydrodynamics model with a local implicit particle-in-cell model. *Journal of Computational Physics*, 268:236–254, 2014.
- Darren L De Zeeuw, Tamas I Gombosi, Clinto PT Groth, Kenneth G Powell, and Quentin F Stout. An adaptive mhd method for global space weather simulations. *IEEE Transactions on Plasma Science*, 28(6):1956–1965, 2000.
- Richard Joseph DeCoster and Louis A Frank. Observations pertaining to the dynamics of the plasma sheet. *Journal of Geophysical Research: Space Physics*, 84(A9):5099–5121, 1979.
- Andreas Dedner, Friedemann Kemm, Dietmar Kröner, C-D Munz, Thomas Schnitzer, and Matthias Wesenberg. Hyperbolic divergence cleaning for the mhd equations. *Journal of Computational Physics*, 175(2):645–673, 2002.
- William D. Gropp Dinesh K. Kaushik, David E. Keyes and Barry F. Smith. Using memory performance to understand the mixed mpi/openmp model. <https://www.mcs.anl.gov/research/projects/petsc-fun3d/Talks/pcfd00.pdf>. Accessed: 2019-06-13.
- Jack J Dongarra, Piotr Luszczek, and Antoine Petit. The linpack benchmark: past, present and future. *Concurrency and Computation: practice and experience*, 15(9):803–820, 2003.
- John C Dorelli, Alex Gloer, Glyn Collinson, and Gábor Tóth. The role of the hall effect in the global structure and dynamics of planetary magnetospheres: Ganymede as a case study. *Journal of Geophysical Research: Space Physics*, 120(7):5377–5392, 2015.

- JF Drake, M Swisdak, TD Phan, PA Cassak, MA Shay, ST Lepri, RP Lin, E Quataert, and TH Zurbuchen. Ion heating resulting from pickup in magnetic reconnection exhausts. *Journal of Geophysical Research: Space Physics*, 114(A5), 2009.
- SD Drell, HM Foley, and MA Ruderman. Drag and propulsion of large satellites in the ionosphere: An alfvén propulsion engine in space. *Journal of Geophysical Research*, 70(13):3131–3145, 1965.
- Nikolaos Drosinos and Nectarios Koziris. Performance comparison of pure mpi vs hybrid mpi-openmp parallelization models on smp clusters. In *18th International Parallel and Distributed Processing Symposium, 2004. Proceedings.*, page 15. IEEE, 2004.
- Stefan Duling, Joachim Saur, and Johannes Wicht. Consistent boundary conditions at nonconducting surfaces of planetary bodies: Applications in a new ganymede mhd model. *Journal of Geophysical Research: Space Physics*, 119(6):4412–4440, 2014.
- J Egedal, A Le, W Daughton, B Wetherton, PA Cassak, L-J Chen, B Lavraud, RB Torbert, J Dorelli, DJ Gershman, et al. Spacecraft observations and analytic theory of crescent-shaped electron distributions in asymmetric magnetic reconnection. *Physical review letters*, 117(18):185101, 2016.
- Aharon Eviatar, Darrell F Strobel, Brian C Wolven, Paul D Feldman, Melissa A McGrath, and Donald J Williams. Excitation of the ganymede ultraviolet aurora. *The Astrophysical Journal*, 555(2):1013, 2001a.
- Aharon Eviatar, Vytenis M Vasyliūnas, and Donald A Gurnett. The ionosphere of ganymede. *Planetary and Space Science*, 49(3-4):327–336, 2001b.

- S Fatemi, AR Poppe, KK Khurana, M Holmström, and GT Delory. On the formation of ganymede's surface brightness asymmetries: Kinetic simulations of ganymede's magnetosphere. *Geophysical Research Letters*, 43(10):4745–4754, 2016.
- Paul D Feldman, Melissa A McGrath, Darrell F Strobel, H Warren Moos, Kurt D Retherford, and Brian C Wolven. Hst/stis ultraviolet imaging of polar aurora on ganymede. *The Astrophysical Journal*, 535(2):1085, 2000.
- Harley Flanders. Differentiation under the integral sign. *The American Mathematical Monthly*, 80(6):615–627, 1973.
- Python Software Foundation. A foreign function library for python. <https://docs.python.org/3.8/library/ctypes.html>.
- LA Frank, WR Paterson, KL Ackerson, and SJ Bolton. Low-energy electron measurements at ganymede with the galileo spacecraft: Probes of the magnetic topology. *Geophysical research letters*, 24(17):2159–2162, 1997a.
- LA Frank, WR Paterson, KL Ackerson, and SJ Bolton. Outflow of hydrogen ions from ganymede. *Geophysical research letters*, 24(17):2151–2154, 1997b.
- W. R. Paterson Frank L. A. and M. G. Kivelson. Observations of nonadiabatic acceleration of ions in earth's magnetotail. *Journal of Geophysical Research: Space Physics*, 1994.
- Galileo Galilei. *Sidereus Nuncius, or the sidereal messenger*. University of Chicago Press, 2016.
- Olivier Grasset, MK Dougherty, A Coustenis, EJ Bunce, C Erd, D Titov, M Blanc, A Coates, P Drossart, LN Fletcher, et al. Jupiter icy moons explorer (juice): An

- esa mission to orbit ganymede and to characterise the jupiter system. *Planetary and Space Science*, 78:1–21, 2013.
- Philipp Grete, Forrest W Glines, and Brian W O’Shea. K-athena: a performance portable structured grid finite volume magnetohydrodynamics code. *arXiv preprint arXiv:1905.04341*, 2019.
- DA Gurnett, WS Kurth, A Roux, SJ Bolton, and CF Kennel. Evidence for a magnetosphere at ganymede from plasma-wave observations by the galileo spacecraft. *Nature*, 384(6609):535, 1996.
- Donald A Gurnett and Amitava Bhattacharjee. *Introduction to plasma physics: with space and laboratory applications*. Cambridge university press, 2005.
- DT Hall, PD Feldman, Ml A McGrath, and DF Strobel. The far-ultraviolet oxygen airglow of europa and ganymede. *The Astrophysical Journal*, 499(1):475, 1998.
- William K Hartmann and Ron Miller. *The grand tour: a traveler’s guide to the Solar System*. Workman Publishing, 2005.
- M Hoshino, T Mukai, T Yamamoto, and S Kokubun. Ion dynamics in magnetic reconnection: Comparison between numerical simulation and geotail observations. *Journal of Geophysical Research: Space Physics*, 103(A3):4509–4530, 1998.
- YQ Hu, XC Guo, and Chi Wang. On the ionospheric and reconnection potentials of the earth: Results from global mhd simulations. *Journal of Geophysical Research: Space Physics*, 112(A7), 2007.
- Zhenguang Huang, Gábor Tóth, Bart van der Holst, Yuxi Chen, and Tamas Gombosi. A six-moment multi-fluid plasma model. *Journal of Computational Physics*, 2019.

- Wing-Huen Ip and Andreas Kopp. Resistive mhd simulations of ganymede's magnetosphere 2. birkeland currents and particle energetics. *Journal of Geophysical Research: Space Physics*, 107(A12):SMP-42, 2002.
- P Janhunen, M Palmroth, T Laitinen, I Honkonen, L Juusola, G Facskó, and Tuija I Pulkkinen. The gumics-4 global mhd magnetosphere-ionosphere coupling simulation. *Journal of Atmospheric and Solar-Terrestrial Physics*, 80:48-59, 2012.
- Xianzhe Jia, Raymond J Walker, Margaret G Kivelson, Krishan K Khurana, and Jon A Linker. Three-dimensional mhd simulations of ganymede's magnetosphere. *Journal of Geophysical Research: Space Physics*, 113(A6), 2008.
- Xianzhe Jia, Raymond J Walker, Margaret G Kivelson, Krishan K Khurana, and Jon A Linker. Properties of ganymede's magnetosphere inferred from improved three-dimensional mhd simulations. *Journal of Geophysical Research: Space Physics*, 114(A9), 2009.
- Xianzhe Jia, Raymond J Walker, Margaret G Kivelson, Krishan K Khurana, and Jon A Linker. Dynamics of ganymede's magnetopause: Intermittent reconnection under steady external conditions. *Journal of Geophysical Research: Space Physics*, 115(A12), 2010.
- Xianzhe Jia, James A Slavin, Tamas I Gombosi, Lars KS Daldorff, Gabor Toth, and Bart Holst. Global mhd simulations of mercury's magnetosphere with coupled planetary interior: Induction effect of the planetary conducting core on the global interaction. *Journal of Geophysical Research: Space Physics*, 120(6):4763-4775, 2015.
- Haoqiang Jin, Dennis Jespersen, Piyush Mehrotra, Rupak Biswas, Lei Huang, and

- Barbara Chapman. High performance computing using mpi and openmp on multi-core parallel systems. *Parallel Computing*, 37(9):562–575, 2011.
- Gabriele Jost, Hao-Qiang Jin, Dieter anMey, and Ferhat F Hatay. Comparing the openmp, mpi, and hybrid programming paradigm on an smp cluster. 2003.
- Nawapat Kaweeyanun, Adam Masters, and Xianzhe Jia. Favorable conditions for magnetic reconnection at ganymede’s upstream magnetopause. *Geophysical Research Letters*, page e2019GL086228, 2020.
- Rony Keppens, Jannis Teunissen, Chun Xia, and Oliver Porth. Mpi-amrvac: a parallel, grid-adaptive pde toolkit. *Computers & Mathematics with Applications*, 2020.
- David E Keyes, Dinesh K Kaushik, and Barry F Smith. Prospects for cfd on petaflops systems. In *Parallel Solution of Partial Differential Equations*, pages 247–277. Springer, 2000.
- Margaret G Kivelson, KK Khurana, FV Coroniti, S Joy, CT Russell, RJ Walker, J Warnecke, L Bennett, and C Polanskey. The magnetic field and magnetosphere of ganymede. *Geophysical Research Letters*, 24(17):2155–2158, 1997.
- Margaret G Kivelson, Fran Bagenal, William S Kurth, Fritz M Neubauer, Chris Paranicas, and Joachim Saur. Magnetospheric interactions with satellites. *Jupiter: The planet, satellites and magnetosphere*, pages 513–536, 2004.
- MG Kivelson, KK Khurana, CT Russell, RJ Walker, et al. Discovery of ganymede’s magnetic field by the galileo spacecraft. *Nature*, 384(6609):537, 1996.
- MG Kivelson, J Warnecke, L Bennett, S Joy, KK Khurana, JA Linker, CT Rus-

- sell, RJ Walker, and C Polanskey. Ganymede's magnetosphere: Magnetometer overview. *Journal of Geophysical Research: Planets*, 103(E9):19963–19972, 1998.
- MG Kivelson, KK Khurana, and M Volwerk. The permanent and inductive magnetic moments of ganymede. *Icarus*, 157(2):507–522, 2002.
- Andreas Kopp and Wing-Huen Ip. Resistive mhd simulations of ganymede's magnetosphere 1. time variabilities of the magnetic field topology. *Journal of Geophysical Research: Space Physics*, 107(A12):SMP–41, 2002.
- Barry Koren. A robust upwind discretization method for advection, diffusion and source terms. In *Numerical methods for advection-diffusion problems*, pages 117–138. Vieweg, 1993.
- Oak Ridge National Laboratory. Summit user guide. <https://docs.olcf.ornl.gov>, 2020.
- Giovanni Lapenta. Exactly energy conserving semi-implicit particle in cell formulation. *Journal of Computational Physics*, 334:349–366, 2017.
- Giovanni Lapenta, JU Brackbill, and Paolo Ricci. Kinetic approach to microscopic-macroscopic coupling in space and laboratory plasmas. *Physics of plasmas*, 13(5):055904, 2006.
- Giovanni Lapenta, J Berchem, M Zhou, RJ Walker, M El-Alaoui, ML Goldstein, WR Paterson, BL Giles, CJ Pollock, CT Russell, et al. On the origin of the crescent-shaped distributions observed by mms at the magnetopause. *Journal of Geophysical Research: Space Physics*, 122(2):2024–2039, 2017.
- Paul Lindberg. Performance obstacles for threading: How do they affect openmp code? <https://software.intel.com/en-us/articles/>

performance-obstacles-for-threading-how-do-they-affect-openmp-code. Accessed: 2019-06-13.

Timur Jaan Linde. *A Three-Dimensional Adaptive Multi uid MHD Model of the Heliosphere*. PhD thesis, The University of Michigan, 1998.

LLVM. The llvm compiler infrastructure. <https://llvm.org/>.

Yingjuan Ma, Christopher T Russell, Gabor Toth, Yuxi Chen, Andrew F Nagy, Yuki Harada, James McFadden, Jasper S Halekas, Rob Lillis, John EP Connerney, et al. Reconnection in the martian magnetotail: Hall-mhd with embedded particle-in-cell simulations. *Journal of Geophysical Research: Space Physics*, 123(5):3742–3763, 2018.

Stefano Markidis, Giovanni Lapenta, et al. Multi-scale simulations of plasma with ipic3d. *Mathematics and Computers in Simulation*, 80(7):1509–1519, 2010.

Rodney J Mason. Implicit moment particle simulation of plasmas. *Journal of Computational Physics*, 41(2):233–244, 1981.

Melissa A McGrath, Xianzhe Jia, Kurt Retherford, Paul D Feldman, Darrell F Strobel, and Joachim Saur. Aurora on ganymede. *Journal of Geophysical Research: Space Physics*, 118(5):2043–2054, 2013.

PM Molyneux, JD Nichols, NP Bannister, EJ Bunce, JT Clarke, SWH Cowley, J-C Gérard, Denis Grodent, SE Milan, and C Paty. Hubble space telescope observations of variations in ganymede’s oxygen atmosphere and aurora. *Journal of Geophysical Research: Space Physics*, 123(5):3777–3793, 2018.

Susanna Musotto, Ferenc Varadi, William Moore, and Gerald Schubert. Numerical simulations of the orbits of the galilean satellites. *Icarus*, 159(2):500–504, 2002.

- Fritz M Neubauer. The sub-alfvénic interaction of the galilean satellites with the jovian magnetosphere. *Journal of Geophysical Research: Planets*, 103(E9):19843–19866, 1998.
- Jonathan Wei Xiang Ng. *Fluid closures for the modelling of reconnection and instabilities in magnetotail current sheets*. PhD thesis, Princeton University, 2019.
- TOP500 Organization. November 2019 top 500 list. <https://www.top500.org/lists/2019/11/>, 2019.
- C Paranicas, WR Paterson, AF Cheng, BH Mauk, RW McEntire, LA Frank, and DJ Williams. Energetic particle observations near ganymede. *Journal of Geophysical Research: Space Physics*, 104(A8):17459–17469, 1999.
- C Paty, W Paterson, and R Winglee. Ion energization in ganymedes magnetosphere: Using multifluid simulations to interpret ion energy spectrograms. *Journal of Geophysical Research: Space Physics*, 113(A6), 2008.
- Carol Paty and Robert Winglee. Multi-fluid simulations of ganymede’s magnetosphere. *Geophysical research letters*, 31(24), 2004.
- Carol Paty and Robert Winglee. The role of ion cyclotron motion at ganymede: Magnetic field morphology and magnetospheric dynamics. *Geophysical research letters*, 33(10), 2006.
- AP Payan, CS Paty, and KD Retherford. Uncovering local magnetospheric processes governing the morphology and variability of ganymede’s aurora using three-dimensional multifluid simulations of ganymede’s magnetosphere. *Journal of Geophysical Research: Space Physics*, 120(1):401–413, 2015.

TD Phan, G Paschmann, C Twitty, FS Mozer, JT Gosling, JP Eastwood, M Øieroset, H Reme, and EA Lucek. Evidence for magnetic reconnection initiated in the magnetosheath. *Geophysical Research Letters*, 34(14), 2007.

Kenneth G Powell, Philip L Roe, Timur J Linde, Tamas I Gombosi, and Darren L De Zeeuw. A solution-adaptive upwind scheme for ideal magnetohydrodynamics. *Journal of Computational Physics*, 154(2):284–309, 1999.

Rolf Rabenseifner, Georg Hager, and Gabriele Jost. Hybrid mpi/openmp parallel programming on clusters of multi-core smp nodes. In *2009 17th Euromicro international conference on parallel, distributed and network-based processing*, pages 427–436. IEEE, 2009a.

Rolf Rabenseifner, Georg Hager, and Gabriele Jost. Hybrid mpi/openmp parallel programming on clusters of multi-core smp nodes. In *2009 17th Euromicro international conference on parallel, distributed and network-based processing*, pages 427–436. IEEE, 2009b.

Christopher Rackauckas and Qing Nie. Differentialequations. jl—a performant and feature-rich ecosystem for solving differential equations in julia. *Journal of Open Research Software*, 5(1), 2017.

Joachim Raeder, Douglas Larson, Wenhui Li, Emil L Kepko, and Timothy Fuller-Rowell. Opengcm simulations for the themis mission. *Space Science Reviews*, 141(1-4):535–555, 2008.

Philip L Roe. Approximate riemann solvers, parameter vectors, and difference schemes. *Journal of computational physics*, 43(2):357–372, 1981.

Yousef Saad and Martin H Schultz. Gmres: A generalized minimal residual algorithm

for solving nonsymmetric linear systems. *SIAM Journal on scientific and statistical computing*, 7(3):856–869, 1986.

Joachim Saur, Stefan Duling, Lorenz Roth, Xianzhe Jia, Darrell F Strobel, Paul D Feldman, Ulrich R Christensen, Kurt D Retherford, Melissa A McGrath, Fabrizio Musacchio, et al. The search for a subsurface ocean in ganymede with hubble space telescope observations of its auroral ovals. *Journal of Geophysical Research: Space Physics*, 120(3):1715–1737, 2015.

Hsi-Yu Schive, John A ZuHone, Nathan J Goldbaum, Matthew J Turk, Massimo Gaspari, and Chin-Yu Cheng. Gamer-2: a gpu-accelerated adaptive mesh refinement code—accuracy, performance, and scalability. *Monthly Notices of the Royal Astronomical Society*, 481(4):4815–4840, 2018.

Gerald Schubert, Keke Zhang, Margaret G Kivelson, and John D Anderson. The magnetic field and internal structure of ganymede. *Nature*, 384(6609):544, 1996.

Jack Scudder and William Daughton. Illuminating electron diffusion regions of collisionless magnetic reconnection using electron agyrotropy. *Journal of Geophysical Research: Space Physics*, 113(A6), 2008.

MA Shay, TD Phan, CC Haggerty, Masaki Fujimoto, JF Drake, K Malakit, PA Casak, and M Swisdak. Kinetic signatures of the region surrounding the x line in asymmetric (magnetopause) reconnection. *Geophysical Research Letters*, 43(9):4145–4154, 2016.

Adam P Showman and Renu Malhotra. Tidal evolution into the laplace resonance and the resurfacing of ganymede. *Icarus*, 127(1):93–111, 1997.

- Adam P Showman and Renu Malhotra. The galilean satellites. *Science*, 286(5437): 77–84, 1999.
- Lorna Smith and Mark Bull. Development of mixed mode mpi/openmp applications. *Scientific Programming*, 9(2-3):83–98, 2001.
- MF Smith and DJ Rodgers. Ion distributions at the dayside magnetopause. *Journal of Geophysical Research: Space Physics*, 96(A7):11617–11624, 1991.
- F Sohl, T Spohn, D Breuer, and K Nagel. Implications from galileo observations on the interior structure and chemistry of the galilean satellites. *Icarus*, 157(1): 104–119, 2002.
- I.V. Sokolov, E.V. Timofeev, J. Sakai, and K. Takayama. On shock-capturing schemes using artificial wind. *Shock Waves*, 9(6):423–427, Dec 1999. ISSN 1432-2153. doi: 10.1007/s001930050173. URL <https://doi.org/10.1007/s001930050173>.
- Dan Stanzione, Bill Barth, Niall Gaffney, Kelly Gaither, Chris Hempel, Tommy Minyard, Susan Mehringer, Eric Wernert, H Tufo, D Panda, et al. Stampede 2: the evolution of an xsede supercomputer. In *Proceedings of the Practice and Experience in Advanced Research Computing 2017 on Sustainability, Success and Impact*, page 15. ACM, 2017.
- Quentin F Stout, Darren L De Zeeuw, Tamas I Gombosi, Clinton PT Groth, Hal G Marshall, and Kenneth G Powell. Adaptive blocks: A high performance data structure. In *Supercomputing, ACM/IEEE 1997 Conference*, pages 57–57. IEEE, 1997.

- M Swisdak. Quantifying gyrotopry in magnetic reconnection. *Geophysical Research Letters*, 43(1):43–49, 2016.
- Gábor Tóth. The lasy preprocessor and its application to general multi-dimensional codes. 1997.
- Gábor Tóth, Igor V Sokolov, Tamas I Gombosi, David R Chesney, C Robert Clauer, Darren L De Zeeuw, Kenneth C Hansen, Kevin J Kane, Ward B Manchester, Robert C Oehmke, et al. Space weather modeling framework: A new tool for the space science community. *Journal of Geophysical Research: Space Physics*, 110(A12), 2005.
- Gábor Tóth, Darren L De Zeeuw, Tamas I Gombosi, and Kenneth G Powell. A parallel explicit/implicit time stepping scheme on block-adaptive grids. *Journal of Computational Physics*, 217(2):722–758, 2006.
- Gábor Tóth, Yingjuan Ma, and Tamas I Gombosi. Hall magnetohydrodynamics on block-adaptive grids. *Journal of Computational Physics*, 227(14):6967–6984, 2008.
- Gábor Tóth, Bart Van der Holst, Igor V Sokolov, Darren L De Zeeuw, Tamas I Gombosi, Fang Fang, Ward B Manchester, Xing Meng, Dalal Najib, Kenneth G Powell, et al. Adaptive numerical algorithms in space weather modeling. *Journal of Computational Physics*, 231(3):870–903, 2012.
- Gábor Tóth, Xianzhe Jia, Stefano Markidis, Ivy Bo Peng, Yuxi Chen, Lars KS Daldorff, Valeriy M Tenishev, Dmitry Borovikov, John D Haiducek, Tamas I Gombosi, et al. Extended magnetohydrodynamics with embedded particle-in-cell simulation of ganymede’s magnetosphere. *Journal of Geophysical Research: Space Physics*, 121(2):1273–1293, 2016.

- Gábor Tóth, Yuxi Chen, Tamas I Gombosi, Paul Cassak, Stefano Markidis, and Ivy Bo Peng. Scaling the ion inertial length and its implications for modeling reconnection in global simulations. *Journal of Geophysical Research: Space Physics*, 122(10), 2017.
- Ch Tsitouras. Runge–kutta pairs of order 5 (4) satisfying only the first column simplifying assumption. *Computers & Mathematics with Applications*, 62(2):770–775, 2011.
- Henk A Van der Vorst. Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems. *SIAM Journal on scientific and Statistical Computing*, 13(2):631–644, 1992.
- Vytenis M Vasyliūnas and Aharon Eviatar. Outflow of ions from ganymede: A reinterpretation. *Geophysical research letters*, 27(9):1347–1349, 2000.
- John P Verboncoeur. Particle simulation of plasmas: review and advances. *Plasma Physics and Controlled Fusion*, 47(5A):A231, 2005.
- M Volwerk, X Jia, C Paranicas, WS Kurth, MG Kivelson, and KK Khurana. Ulf waves in ganymede’s upstream magnetosphere. In *Annales Geophysicae*, volume 31, page 45. Copernicus GmbH, 2013.
- Mo Volwerk, Margaret G Kivelson, KK Khurana, and RL McPherron. Probing ganymede’s magnetosphere with field line resonances. *Journal of Geophysical Research: Space Physics*, 104(A7):14729–14738, 1999.
- Chi Wang, XiaoCheng Guo, Zhong Peng, BinBin Tang, TianRan Sun, WenYa Li, and YouQiu Hu. Magnetohydrodynamics (mhd) numerical simulations on the in-

- teraction of the solar wind with the magnetosphere: A review. *Science China Earth Sciences*, 56(7):1141–1157, 2013.
- Liang Wang, Kai Germaschewski, Ammar Hakim, Chuanfei Dong, Joachim Raeder, and Amitava Bhattacharjee. Electron physics in 3-d two-fluid 10-moment modeling of ganymede’s magnetosphere. *Journal of Geophysical Research: Space Physics*, 123(4):2815–2830, 2018.
- Christopher J White, James M Stone, and Charles F Gammie. An extension of the athena++ code framework for grmhd based on advanced riemann solvers and staggered-mesh constrained transport. *The Astrophysical Journal Supplement Series*, 225(2):22, 2016.
- Wiki. Ganymede (moon). [https://en.wikipedia.org/wiki/Ganymede_\(moon\)](https://en.wikipedia.org/wiki/Ganymede_(moon)), a.
- Wiki. Multiple dispatch. https://en.wikipedia.org/wiki/Multiple_dispatch, b.
- Hu You-Qiu, Guo Xiao-Cheng, Li Guo-Qiang, Wang Chi, and Huang Zhao-Hui. Oscillation of quasi-steady earth’s magnetosphere. *Chinese Physics Letters*, 22(10):2723, 2005.
- Seiji Zenitani, Michael Hesse, Alex Klimas, and Masha Kuznetsova. New measure of the dissipation region in collisionless magnetic reconnection. *Physical review letters*, 106(19):195003, 2011.
- Binzheng Zhang, Kareem A Sorathia, John G Lyon, Viacheslav G Merkin, Jeffrey S Garretson, and Michael Wiltberger. Gamera: A three-dimensional finite-volume mhd solver for non-orthogonal curvilinear geometries. *The Astrophysical Journal Supplement Series*, 244(1):20, 2019a.

Weiqun Zhang, Ann Almgren, Vince Beckner, John Bell, Johannes Blaschke, Cy Chan, Marcus Day, Brian Friesen, Kevin Gott, Daniel Graves, et al. Amrex: a framework for block-structured adaptive mesh refinement. 2019b.

Hongyang Zhou. Visana, a package for swmf data processor and analyzer. <https://henry2004y.github.io/VisAnaJulia/dev/>, 2020.

Hongyang Zhou and Gábor Tóth. Efficient openmp parallelization to a complex mpi parallel magnetohydrodynamics code. *Journal of Parallel and Distributed Computing*, 2020.

Hongyang Zhou, Gábor Tóth, Xianzhe Jia, Yuxi Chen, and Stefano Markidis. Embedded kinetic simulation of ganymede's magnetosphere: Improvements and inferences. *Journal of Geophysical Research: Space Physics*, 2019.

Hongyang Zhou, Gábor Tóth, Xianzhe Jia, and Yuxi Chen. Reconnection-driven dynamics at ganymede's upstream magnetosphere: 3d global hall mhd and mhd-epic simulations. *Journal of Geophysical Research: Space Physics*, accepted, 2020. doi: 10.1002/essoar.10502974.1. URL <https://doi.org/10.1002/essoar.10502974.1>.