UNIVERSITY OF CALIFORNIA SAN DIEGO

High-performance computational techniques for x-ray imaging and micromagnetic analysis of periodic arrays

A Dissertation submitted in partial satisfaction of the requirements
for the degree Doctor of Philosophy

in

Electrical Engineering (Applied Physics)

by

Fangzhou Ai

Committee in charge:

Professor Vitaliy Lomakin, Chair
Professor Yeshaiahu Fainman
Professor Eric E. Fullerton
Professor Vlado A. Lubarda
Professor Daniel F. Sievenpiper

2024

The Dissertation of Fangzhou Ai is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2024

DEDICATION

To my family and friends.

TABLE OF CONTENTS

DISSERTATION APPROVAL PAGE iii

DEDICATION iv

TABLE OF CONTENTS v

LIST OF FIGURES vii

LIST OF TABLES xi

ACKNOWLEDGEMENTS xii

VITA xiv

ABSTRACT OF THE DISSERTATION xv

INTRODUCTION 1

Chapter 1 Introduction to parallel computing 4

1.1 Open Multi-Processing: parallel computing on multi-core CPU 5

1.2 Open Message Passing Interface: parallel computing on multiple CPUs 6

1.3 Compute Unified Device Architecture: parallel computing on GPU 6

1.4 Hybrid methods on heterogenous architectures 7

Chapter 2 Brief overview of micromagnetics 9

2.1 Landau-Lifshitz-Gilbert equation 9

2.2 Principal interactions in continuum representation 10

2.3 Spin waves in magnetic materials 18

Chapter 3 Real time 3D coherent X-ray diffraction imaging 23

3.1 Formulation of algorithm 25

3.2 Validation and performance 32

3.2 Summary 39

Acknowledgements 40

Chapter 4 Spin-wave assisted synchronization in 2D arrays of spin torque oscillators 41

4.1 Problem formulation 42

4.2 Device structure design...44

4.3 Summary 51

Acknowledgements...52

Chapter 5 Fast periodic interpolation method for superposition sums in a periodic unit cell .53

5.1 Problem formulation 55

5.2 FFT-PIM algorithm...59

5.3 Numerical results 69

5.4 Summary 77

Acknowledgements...80

Chapter 6 Periodic micromagnetic finite element method 81

6.1 Problem formulation 82

6.2 Implementation 86

6.3 Results...95

6.4 Summary 102

Acknowledgements...103

Chapter 7 Periodic phase diagrams in micromagnetics with an eigenvalue solver 104

7.1 Problem formulation 105

7.2 Results...108

7.3 Conclusion 111

Acknowledgements...112

Chapter 8 Conclusion...113

REFERENCES 115

LIST OF FIGURES

Figure 2.1: Illustration of how $\mathbf{H}_{\mathrm{eff}}$ (blue arrow) interacts with $\mathbf{M}$, the precession term $\mathbf{M}\times\mathbf{H}_{\mathrm{eff}}$ is the green arrow that keeps the $\mathbf{M}$ revolving, and the damping term $\mathbf{M}\times\mathbf{M}\times\mathbf{H}_{\mathrm{eff}}$ is the red arrow that tries to align the $\mathbf{M}$ with $\mathbf{H}_{\mathrm{eff}}$. For most practical magnetic materials damping effect always exists. 11

Figure 2.2: Illustration of magnetization with different J_{ex}. (a) Ferromagnetism with $J_{\mathrm{ex}} > 0$, (b) anti-ferromagnetism with $J_{\mathrm{ex}} < 0$, the net magnetization is equal to 0 and (c) ferrimagnetism with $J_{\mathrm{ex}} < 0$, the net magnetization is larger than 0. 13

Figure 3.1: (A) A sample holder containing a cell rotated to different angles with coherent X-ray beam impinges on this cell, diffraction patterns are measured. (B) A three-dimensional diffraction pattern was assembled from two-dimensional diffraction patterns. The three-dimensional structure of the cell (C) was iteratively reconstructed. 23

Figure 3.2: Illustration of our routine. The green region represents the step A, pre-reconstruct one projected object. The dark region representing step B, reconstruct multiple episodes sequentially. The yellow region is step C, we repeated step A and B times and merge some best of them. The pink region is step D, where we interpolate spectral contents to reconstruct the object. 28

Figure 3.3: Resolution determined by PRTF(A) and FSC(B) of CSWA and CPRA. The dashed lines here are the determined resolutions of CSWA with $N_i^{CSWA} = 500$ and 1000 respectively, the solid lines are determined resolutions of CPRA with different N_e and N_{ei}. CPRA's resolution from FSC are all equal or above CSWA except $N_e = 1,\ N_{ei} = 1$ configuration. 33

Figure 3.4: Volume density of original object (A), reconstructed object from CSWA (B), CPRA method with $N_e = 1,\ N_{ei} = 2$ (C) and $N_e = 1,\ N_{ei} = 10$ (D). (D) exhibits detached density cloud in the top right corner while (B) and (C) are not. Here (C) is slightly better than (B) since the connection of two density cloud is weaker. 34

Figure 3.5: Central slices through x-y), y-z , z-x plane, (A) original object, (B) reconstructed object from CSWA, (C) reconstructed object from CPRA with $N_e = 1,\ N_{ei} = 2$, (D) reconstructed object from CPRA with $N_e = 1,\ N_{ei} = 10$. (D) looks most like (A) with many similar features, and (C) is still slightly better than (B) for there're clear boundaries of density. 35

Figure 3.6: Reconstruction of S. aureus cell. (A-C) Reconstructed 2D object. (D) PRTF of all 2D reconstructions (gray lines), mean values (blue line), minimum values (red line) and maximum values (green line). The horizontal line is the 1/e criterion. (D) Reconstructed surface morphology of S. aureus. Black circles indicate the two representative depressions. 36

Figure 3.7: Performance of CSWA and CPRA method on different devices. (A) computational time of different methods on CPU and GPU, (B) acceleration ratio across different methods on CPU and GPU, defined by CSWA time consumption over that of CPRA. 38

Figure 4.1: Illustration of the model without high damping region for (a) $N = 3$ (b) $N = 12$ and with high damping region (c) $N = 12$. Grey substrate here is of low damping constant, black congruent triangles are of high damping constant, and the small dots spin torque oscillators. The corresponding *y*-magnetization are shown below. 45

Figure 4.2: (a) The simulated total output power under 0K (blue dashed line) and 300K (red dashed line) vs theoretical maximum (black solid line), at both temperature the result indicates global in-phase synchronization is achieved. (b) Snapshot of y-magnetization under 300K, 48 STNOs in total. Phase shift can be observed via visual inspect. 47

Figure 4.3: Calculated synchronization efficiency η of the whole array under different applied field $\mathbf{H}_{app}$. A sharp transition from low output power to nearly 100% output power indicates the emergence of global in-phase synchronization. And the synchronization efficiency η is maintained well above 90% after this transition. 49

Figure 4.4: Synchronization with mean value $\mu = 6.37\times10^5\,\text{A/m}$ and deviation $\sigma = 0,1,5,10\%$ for saturation magnetization $M_s = \mu \pm \sigma / 2$ and $N = 48$. (a) The synchronization efficiency η slightly drops from 98.5% to 96.5. (b) A snapshot of average of m_y over all 48 STNOs vs m_y from two well-separated corners with $\sigma = 10\%$ 50

Figure 4.5: Magnetization along y-direction of synchronized (a) ring and (b) concentric loops. These two structures are built by turning off some STNOs on the model of size $N = 27$. 51

Figure 5.1: Illustration for a 2D periodic problem consisting of an infinite 2D array of cubes. The central red cube is the zeroth unit cell. Surrounding it along the x and y axes are the 1^{st} order image cubes in green and the 2^{nd} order image cubes in blue. 54

Figure 5.2: Convergence of PGFs, $L_x, L_y, L_z = 1$, $x = y = z = 0.5$. (a) The relative error of the sum of the first m terms from Eq. (5.5), Eq. (5.6) and Eq. (5.7), with $k_0 = -1 - j$ and $k_{x0} = 1 - j, k_{y0} = 1 + j, k_{z0} = -1 + j$, (b) The relative error of the sum of the first m terms from Eq. (5.8), Eq. (5.9) and Eq. (5.10). Exponential convergence is achieved in both cases. 59

Figure 5.3: Illustration of the source (black) and observer (red) grids for the far-zone PSP component. Black and red circles are source/observer grid points on the Cartesian lattices. With on-grid black/red circles and random black/red dots. Black/red/green arrows are projections, interpolations and interactions among grid sources. 62

Figure 5.4: The error correction step 4 for the near-zone PSP component. The black dot is a source point and red dot is an observer point. The green region is the error-correction region Ω^{ER} . The black circles are grid points. Direct calculation within this region is done by subtracting the grid interaction inside the green region and adding the direct interactions. 68

Figure 5.5: Potential on the x axis of a coaxial structure. (a) Coaxial structure unit cell ($L = 1$) and potential of the non-periodic unit cell. Inner radius $r_1 = 1$ with negative line charge density $\rho_1 = -1$, outer radius $r_2 = 2$ with positive line charge density $\rho_2 = 1/2$. (b) PSP with a 1D periodicity along the x axis with $L_x = 1$ for $k_0 = k_{x0} = 0$ and a $k_0 = 1$, $k_{x0} = 1 - j$ 70

Figure 5.6: Relative error of the far-zone PSP for a NPSP case with a 1D periodicity along the x -axis. (a) Relative error for 1st, 3rd, and 6th order interpolation with varying uniform grid sizes with $i_d = 1$. (b) Relative error of cubic interpolation with 1000 sparse grid points and different numbers of subtracted near-zone unit cells. 72

Figure 5.7: Prepossessing time for the non-periodic case, NPSP case with 3D periodicity, and for their difference. 73

Figure 5.8: Execution time versus N of evaluating the non-periodic potential and NPSP PSP with 3D periodicity on (a) eight-core CPU and (b) on GPU. 74

Figure 5.9: Execution time versus the relative error for the near-zone PSP component on CPU and GPU for a 3D NPSP case with $N = 53K$ and $N = 510K$. 76

Figure 5.10: (a)Preprocessing time and (b) execution time versus N of evaluating the PSP for a 3D NPSP and high-frequency dynamic cases for the relative errors of $1e-3$ and $1e-5$. 77

Figure 6.1: Categories of 1D PBC unit cell (solid black) and its nearest images (shadowed), (a) non-touching, non-protruding case, (b) touching, non-protruding case, (c) non-touching, protruding case and (d) touching, protruding case. 83

Figure 6.2: Illustration of (a) Protruding unit cell; (b) its geometry center (black dot) and its protruding parts in green and blue; (c) Regular unit cell after shifting the protruding parts. 96

Figure 6.3: Hysteresis loop along $\hat{z}-$ direction of infinite long periodic rod along $\hat{z}-$direction, the inset is the unit cell. The loop is of square shape and coercive field is around 3050 Oe, close to theoretical value 3085 Oe. 96

Figure 6.4: Example of 2D T-NP-PBC case. (a) Equilibrium vortex state without PBC and (b) the uniform state when with 2D T-NP-PBC. The yellow dashed line is the location of the line excitation. (c) Angles between wave vector of propagating spin wave and magnetization. (d) Simulated wavelength (circle marks) and theoretical prediction (blue dashed line). 98

Figure 6.5: Example of 3D T-NP-PBC case. (a) Multi-grain structure (the unit cell is marked as a yellow cube); (b) multi-grain structure folded into the unit cell; (c) hysteresis loop for the periodic multi-grain structure. The unit cell size is $6\mu m$, the average grain size is $2.5\mu m$. The structure was meshed in a tetrahedral mesh with 70 million elements and 12 million nodes. 99

Figure 6.6: Performance comparison of calculation of magnetostatic field between the PM-FEM on 3D T-NP-PBC case and non-periodic case with original method on single-thread CPU and GPU 101

Figure 7.1: Illustration for a periodic problem consisting along $\hat{x}$ (yellow), $\hat{y}$ (blue) and $\hat{z}$ (green) direction with 0^{th} unit cell (red) and its periodic images (grey). 104

Figure 7.2: The left figure is the phase diagram (dispersion relationship) calculated from theoretical equation (black circles), LLG time-domain solver (green circles), eigenvalue solver (red line) and its relative error with respect to the theoretical values. The right figure is the y component of the magnetization perturbation $\mathbf{v}$ from the eigenstate of $k_x = \pi / L_x$. 109

Figure 7.3: The left figure is the 1D phase diagram of the film with a hole in the middle from the periodic LLG eigen value solver (red dashed line). The right figure is the y-component of the magnetization perturbation $\mathbf{v}$ from the eigen state of $k_{x0} = \pi / L_x$. 110

Figure 7.4: The left figure is the 2D phase diagram of the film with a hole in the middle from the periodic LLG eigenvalue solver (red dashed line). The right figure is the y component of the magnetization perturbation $\mathbf{v}$ from the eigen state of $k_{x0} = \pi / L_x$. 111

LIST OF TABLES

Table 5.1: Grid size for near-zone PSP component evaluation. The near-zone grid is much larger than the far-zone grid when the problem size N is large. Second-order projection/interpolation for near-zone evaluation is used. 75

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor, Professor Vitaliy Lomakin, for his invaluable guidance, patience, and unwavering support throughout my PhD journey. I am profoundly thankful for his mentorship and for the opportunities I have been given to grow as a researcher and as a person under his guidance. This thesis would not have been possible without his exemplary guidance and encouragement.

From my undergraduate studies in Zhejiang University, I would like to thank Professor Yi Yin, Dr. Yuan Zheng, Dr. Ying Fei and Dr. Kunliang Bu for being my advisor and offering guidance of how to do research to prepare for the upcoming doctoral studies.

From my internships in Baidu USA and TikTok, I would like to thank Dr. Tianyi Gao, Youlong Chen, Xuan Zou, Peng Wu and Hanzhi Zhou for offering precious experiences from the industry.

From Professor Vitaliy Lomakin's group I would like to thank Dr. Iana Volvach for the help on micromagnetic modelling techniques. I would like to thank Xueyang Wang for the help on understanding the reconstruction algorithms of coherent X-ray diffraction imaging. I would like to thank Dr. Zhuonan Lin for the help on integrating the Eigenvalue solver with periodic boundary condition. I would also like to thank Jiawei Duan for the help on refining the micromagnetic code. I thank all the members of my thesis committee for reading this manuscript and providing valuable feedback.

Chapter 3, in full, is a reprint of the material as it appears in F. Ai, O. Shpyrko and V. Lomakin "Real time 3D coherent X-ray diffraction imaging", which is currently under review in Phys. Rev. Lett. The dissertation author was the primary researcher and author of this paper.

Chapter 4, in full, is a reprint of the material as it appears in F. Ai and V. Lomakin, “Spin-wave assisted synchronization in 2D arrays of spin torque oscillators”, which is currently under review in Phys. Rev. Appl. The dissertation author was the primary researcher and author of this paper.

Chapter 5, in full, is a reprint of the material as it appears in F. Ai and V. Lomakin, “Fast Fourier Transform periodic interpolation method for superposition sums in a periodic unit cell”, Comp. Phys. Comm., Volume 304, 109291 (2024). The dissertation author was the primary researcher and author of this paper.

Chapter 6, in full, is a reprint of the material as it appears in F. Ai, J. Duan and V. Lomakin, “Periodic micromagnetic finite element method”, which is currently under review in J. Magn. Magn. Mater. The dissertation author was the primary researcher and author of this paper.

Chapter 7, in full, is a reprint of the material as it appears in F. Ai, Z. Lin, J. Duan and V. Lomakin, “Periodic phase diagrams in micromagnetics with an eigenvalue solver”, which is currently under review in IEEE Trans. Magn. The dissertation author was the primary researcher and author of this paper.

Finally, I deeply appreciate my parents Shanshui Ai, Binghong Fang, my parents-in-law Liangfu Weng, Liqin Zhu and my beloved Shu Weng for their love, help and support.

VITA

2018 Bachelor of Science in Physics, Zhejiang University

2020 Master of Science in Electrical Engineering (Applied Physics), University of California San Diego

2024 Doctor of Philosophy in Electrical Engineering (Applied Physics), University of California San Diego

PUBLICATIONS

F. Ai, O. Shpyrko and V. Lomakin, “Real time 3D coherent X-ray diffraction imaging”, *Phys. Rev. Lett*., under review.

F. Ai and V. Lomakin, “Spin-wave assisted synchronization in 2D arrays of spin torque oscillators”, *Phys. Rev. Appl*., under review.

F. Ai and V. Lomakin, “Fast Fourier Transform periodic interpolation method for superposition sums in a periodic unit cell”, *Comp. Phys. Comm., Volume 304, 109291 (2024)*

F. Ai and V. Lomakin, “Periodic Green's functions overcoming singularities with applications to FFT-based superposition method”, in preparation.

F. Ai, J. Duan and V. Lomakin, “Periodic micromagnetic finite element method*”, J. Magn. Magn. Mater.*, under review

F. Ai, Z. Lin, J. Duan and V. Lomakin, “Periodic phase diagrams in micromagnetics with an eigenvalue solver”, *IEEE Trans. Magn.*, under review.

C. Liu, F. Ai, S. Reisbick, A. Zong, A. Pofelski, M. G. Han, F. Camino, C. Jing, V. Lomakin and Y. Zhu, “Correlated spin wave generation and domain-wall oscillation in a topologically textured magnetic film”, *Nat. Mater.*, accepted.

FIELD OF STUDY

Major Field: Electrical Engineering
Studies in Applied Physics
Professor Vitaliy Lomakin

ABSTRACT OF THE DISSERTATION

High-performance computational techniques for x-ray imaging and micromagnetic analysis of periodic arrays

by

Fangzhou Ai

Doctor of Philosophy in Electrical Engineering (Applied Physics)

University of California San Diego, 2024

Professor Vitaliy Lomakin, Chair

High-performance computing (HPC) has shown a great transformation in the capability to address complex scientific and engineering challenges. As hardware and software advances continue to push the boundaries of what is computationally feasible, the pursuit of enhanced performance and cost efficiency remains a driving force. In this dissertation, we develop advanced high-performance computational techniques for x-ray imaging and micromagnetic simulations, and present results related to our proposed approaches.

We begin by introducing fundamental concepts in parallel computing and micromagnetics, laying the groundwork for the subsequent methods and analyses. An efficient algorithm is then presented for reconstructing data in coherent X-ray diffraction imaging (CXDI), enabling real-time data processing and thereby facilitating a deeper understanding of experimental results. We subsequently examine a novel micromagnetic model comprising multiple spin-transfer-torque nano-oscillators (STNOs), demonstrating that collective synchronization across large arrays can be achieved through the propagation of spin waves.

Furthermore, building on these foundations, the dissertation introduces interpolation-based methods for handling periodicities in micromagnetics and related domains. These methods seamlessly extend existing non-periodic frameworks with minimal adjustments and only marginal computational overhead. Finally, by integrating these techniques with the Landau–Lifshitz–Gilbert (LLG) solver and eigenvalue solver, we address micromagnetic problems of solving for the magnetization dynamics both in time and frequency domains involving periodic conditions. The presented results demonstrate the versatility and efficiency of the presented approaches, offering new avenues for advancing our understanding of magnetic phenomena and their computational exploration.

INTRODUCTION

In recent decades, our understanding of complex physical systems has been propelled forward by increasingly sophisticated mathematical models and computational approaches. From micromagnetics, where the behavior of magnetic materials down to the nanoscale demands accurate and efficient numerical simulations, to the inversion of high-resolution experimental datasets in techniques like coherent X-ray diffraction imaging (CXDI), the interplay between advanced theory and cutting-edge computation has become essential. These problems often involve intricate differential equations, large parameter spaces, and subtle boundary conditions that challenge both mathematical ingenuity and computational capability.

While high-performance computing (HPC) techniques—such as exploiting massively parallel architectures, GPUs, and scalable algorithms—provide the raw computational power to handle these complex tasks, the core difficulty remains deeply rooted in mathematical and physical principles. Designing algorithms that can faithfully capture the underlying physics while maintaining accuracy, stability, and efficiency is at the heart of this dissertation. The work presented here emphasizes the theoretical foundations of these methods, demonstrating how careful formulation, discretization, and numerical optimization can yield simulations and data analyses that are both reliable and insightful.

In micromagnetics, understanding the magnetization dynamics governed by the Landau–Lifshitz–Gilbert (LLG) equation requires delicate balance: we must ensure accuracy in representing the physical interactions and also find computational strategies to handle large-scale models, complex geometries, and various time scales. Similarly, in CXDI, reconstructing images and extracting meaningful physical insights from raw diffraction data calls for algorithms that combine robust mathematical techniques with efficient implementations, ensuring real-time processing of massive datasets.

This dissertation brings these themes together, introducing algorithms and enhancements that bridge the gap between theory and experiment. We discuss how these algorithms can be implemented and optimized on modern computing platforms—employing HPC strategies, when necessary, to achieve performance gains. Yet, the narrative remains focused on the mathematical formulations and physical models that make these computational advances meaningful. By starting from fundamental principles, building accurate models, and then carefully tailoring computational strategies to the structure of the underlying equations, we aim to highlight not just the utility of HPC, but the fundamental mathematical and physical reasoning that drives progress in computational science. The dissertation is divided into 8 chapters.

Chapter 1 delivers a brief introduction for parallel computing, one of the most important components of HPC, with three major approaches, single Central Processing Unit (CPU), multiple CPUs, and Graphics Processing Unit (GPU). We rely on these approaches for the later implementation of our techniques.

Chapter 2 introduces the fundamentals of micromagnetics evolving around the Landau–Lifshitz–Gilbert equation. We also present the theory and simulation results of spin wave generation and propagation from the micromagnetic perspective.

Chapter 3 presents a high-performance algorithm that can be applied to processing CXDI experimental data at real-time speed. We include both formulation, implementation, and results with experimental data to verify the validity and performance.

Chapter 4 presents micromagnetic simulations of a structure that can synchronize a large array of spin-transfer torque nano-oscillators (STNOs).

Chapter 5 introduces a fast interpolation-based approach that can handle arbitrary types of periodicities using the finite element method (FEM) approach, that can be incorporated on top of current existing methods without major modification with a small computational overhead.

Chapter 6 combines the algorithms introduced in Chapter 5 with an LLG solver to handle periodic micromagnetic problems in the time domain.

Chapter 7 combines the algorithms introduced in Chapter 5 with an eigenvalue solver to handle linearized periodic micromagnetic problems in the frequency domain.

Finally, chapter 8 gives conclusion and remarks.

Chapter 1 Introduction to parallel computing

Parallel computing is a crucial component of high-performance computing (HPC), fundamentally transforming the way complex computations and data processing tasks are handled. At its core, parallel computing involves the simultaneous use of multiple processing elements to solve computational problems more efficiently than possible with a single processor. It is vital in HPC, where the goal is to achieve peak performance and handle vast amounts of data at high speeds. By dividing large tasks into smaller sub-tasks that can be processed concurrently, parallel computing significantly accelerates the computation time and enhances the overall performance of HPC systems. The ability to process data and perform calculations at high speeds opens new ways for research and development, making parallelization indispensable.

In this chapter we focus on three common methods of parallelization on different hardware architectures. It starts with the introduction of parallelization on a single multi-core CPU utilizing the Open Multi-processing (OpenMP). It then extends the topic to multiple CPUs via Open Message Passing Interface (OpenMPI), that can cooperate with OpenMP to form two level hierarchy for best hardware usage. Next, it addresses parallelization on GPU with Compute Unified Device Architecture (CUDA), the mainstream toolkit for GPU programming. Finally, it gives an overview of hybrid methods by blending all these techniques together for the best performance on different heterogenous architectures. These parallel techniques are the foundation of the high-performance algorithms in this dissertation.

1.1 Open Multi-Processing: parallel computing on multi-core CPU

OpenMP is an industry-standard application programming interface (API) that supports multi-platform shared-memory parallel programming in C, C++, and Fortran. Designed to facilitate the development of parallel applications, OpenMP provides a straightforward and flexible approach to harness the power of multi-core processors, allowing developers to write parallel code more efficiently and with greater portability. By offering a set of compiler directives, library routines, and environment variables, OpenMP enables programmers to parallelize their code incrementally, making it an ideal choice for both new applications and the parallelization of existing serial code.

OpenMP simplifies the creation and management of parallel regions in a program. It uses pragma directives to specify parallel regions, which are blocks of code that can be executed by multiple threads concurrently. These directives are inserted into the code, guiding the compiler on how to generate parallel executable code. This approach allows developers to maintain a single code base that can run on both parallel and serial systems, facilitating easier debugging and testing.

One of the key advantages of OpenMP is its ease of use and scalability. With minimal changes to existing code, developers can achieve significant performance improvements by leveraging the capabilities of modern multi-core processors. OpenMP supports a wide range of parallel programming features, including parallel loops, sections, tasking, and synchronization constructs. These features provide fine-grained control over thread creation, workload distribution, and data sharing, enabling developers to optimize performance and resource utilization effectively.

OpenMP's portability is another major benefit. It is supported by most major compilers and runs on a variety of platforms, from desktop computers to large-scale supercomputers. This widespread support ensures that OpenMP programs can be deployed across different systems with minimal modifications, making it a versatile tool for parallel programming.

1.2 Open Message Passing Interface: parallel computing on multiple CPUs

OpenMPI, short for Open Message Passing Interface, is a prominent Message Passing Interface (MPI) library specifically designed for high-performance computing (HPC). It facilitates efficient communication between nodes in a parallel computing environment, making it indispensable for tasks that require the coordination of multiple processors. OpenMPI supports a wide set of computer architectures and network interconnects, making it highly adaptable to different HPC setups.

The primary objective of OpenMPI is to provide a portable and scalable MPI standard that can be used on all sorts of computing systems, from single-node desktops to thousands of nodes in supercomputers. It implements the full MPI-3 standards and is continually updated to support advancements in computing technology and networking.

Key features of OpenMPI include its robust fault tolerance, its ability to support complex network topologies, and its efficient use of available hardware to optimize performance. It also provides various tools and diagnostic utilities to facilitate development and debugging of parallel applications.

For developers and researchers, OpenMPI offers an extensive toolkit for building and executing distributed applications that can leverage the power of parallel processing to handle complex computations and large data sets more effectively. Whether it is for scientific research, simulations, or data analysis, OpenMPI provides a foundation that enables users to maximize the capabilities of modern HPC environments.

1.3 Compute Unified Device Architecture: parallel computing on GPU

CUDA (Compute Unified Device Architecture) is a parallel computing platform and programming model developed by NVIDIA for harnessing the power of its GPU for general purpose computing. CUDA enables dramatic increases in computing performance by allowing developers to utilize the many cores of a GPU to perform general-purpose mathematical and computational tasks in a much faster way compared to traditional sequential programming on CPUs.

Introduced by NVIDIA in 2007, CUDA provides a comprehensive development environment and toolkit that includes a compiler, libraries, and debugging and optimization tools. This ecosystem allows developers to write software for GPUs in a variety of programming languages, including C, C++, and Fortran. The key advantage of CUDA is its ability to efficiently distribute computing tasks across thousands of GPU cores, enabling significant performance improvements for applications involving large-scale mathematical computations, such as simulations, analytics, and machine learning.

The CUDA model extends the C programming language with specific extensions that allow direct manipulation of GPU resources. It organizes the computation in terms of grids, blocks, and threads, which helps in scaling processing tasks across multiple cores effectively. By minimizing data transfer times between the CPU and GPU and maximizing concurrent execution, CUDA enables developers to achieve high computational throughput across a wide range of applications.

CUDA has been widely adopted in both academia and industry for research and development in fields such as bioinformatics, fluid dynamics, quantum chemistry, and machine learning, where large datasets and complex calculations are common. Its success has propelled the use of GPUs from strictly graphics rendering devices to mainstream

1.4 Hybrid methods on heterogenous architectures

In the domain of parallel computing, the adoption of hybrid computational methods across heterogeneous architectures stands as a strategy for achieving optimal efficiency and scalability. This approach integrates the distinct capabilities of OpenMP, OpenMPI, and CUDA to exploit the combined strengths of CPUs and GPUs within complex, multi-node environments. OpenMP is primarily usdsed to harness parallelism on shared-memory processors, effectively distributing compute-intensive tasks across the multiple cores of CPUs within a single node. CUDA targets the powerful parallel processing capabilities of GPUs, significantly enhancing performance for tasks that involve large-scale data computations, such as deep learning models and scientific simulations.

Further extending the computational framework, OpenMPI allows for inter-node communications in distributed systems, facilitating efficient data transfer and task coordination across a network of computers. To enhance this capability, particularly in GPU-accelerated environments, NVIDIA's NCCL (NVIDIA Collective Communications Library) plays an important role. NCCL optimizes communication patterns that are commonly found in deep learning models, providing highly efficient multi-GPU and multi-node communication. This allows OpenMPI to more effectively manage and synchronize GPU-centric tasks, ensuring operations across distributed systems.

By leveraging this hybrid approach, each component of the HPC ecosystem can be utilized to its fullest potential - OpenMP for intra-node efficiency, CUDA for intensive data processing on GPU, and OpenMPI combined with NCCL for robust inter-node communication and management. Blended combinations between these technologies enable developers to tailor their use of computing resources to match the specific requirements of their computational tasks. This not only maximizes performance but also optimizes resource utilization across diverse computing environments, making it possible to tackle more complex, data-intensive problems with greater speed and accuracy. As such, hybrid methods on heterogeneous architectures represent the forefront of current HPC capabilities, blending multiple forms of computational power to address the ever-growing demands of modern scientific and engineering challenges.

Chapter 2 Brief overview of micromagnetics

Micromagnetics [1] is a branch of physics that studies the magnetization behaviors at nanometer to close to sub-millimeter length scales. The length scales are large enough to average out the atomic structure of the material, but small enough to resolve complex magnetization states, such as domain walls or vortices. It is also a theoretical approach that describes the magnetization process at this intermediate scale between the quantum mechanical scale of individual atoms and the macroscale. It seeks to understand and predict the complex interactions between magnetic moments in a material, which are influenced by both intrinsic properties, such as the magnetic anisotropy, exchange stiffness, and extrinsic factors, such as geometrical constraints and external magnetic fields. Based on the theory, the corresponding micromagnetic simulations are essential tools for this field, providing detailed insights into the magnetization configurations and dynamics. These simulations help in designing and optimizing various magnetic devices, such as sensors, memory devices, and actuators, which are crucial components in modern technology.

This chapter presents the foundation of the micromagnetic theory. It first introduces the core concept, i.e. LLG equation [2], that governs the dynamics of magnetization. It then discusses the modeling of various kinds of interactions in the micromagnetic framework within a continuum representation. Finally, it describes the theory and micromagnetic model of spin waves as a practical example of micromagnetics.

2.1 Landau-Lifshitz-Gilbert equation

The Landau-Lifshitz-Gilbert (LLG) equation is a fundamental mathematical model used in the field of micromagnetics to describe the dynamics of magnetization in ferromagnetic materials. Developed initially by Lev Landau and Evgeny Lifshitz (1935), and later augmented by T. L. Gilbert, the LLG equation provides a comprehensive framework for understanding how the magnetization vector $\mathbf{M}$ (yellow arrow in Fig. 2.1) in a material responds to the total effective field $\mathbf{H}_{\mathrm{eff}}$ (blue arrow in Fig. 2.1). Mathematically, the LLG equation is expressed as:

$$\frac{\partial \mathbf{M}}{\partial t} = -\frac{\gamma}{1+\alpha^2}(\mathbf{M}\times\mathbf{H}_{\mathrm{eff}} + \frac{\alpha}{M_s}\mathbf{M}\times\mathbf{M}\times\mathbf{H}_{\mathrm{eff}}) \qquad (2.1)$$

The γ is the gyromagnetic ratio, M_s is the saturation magnetization and α is the phenomenological damping constant [3] determined by the material property. Here, Eq. (2.1) is often referred to as the explicit form of LLG equation.

There are two terms in the right-hand-side (RHS) of Eq. (2.1). The first term is $\mathbf{M}\times\mathbf{H}_{\mathrm{eff}}$ (green arrow in Fig. 2.1), and it describes a precessional torque that is perpendicular to both $\mathbf{M}$ and $\mathbf{H}_{\mathrm{eff}}$, leading to an energy non-dissipative precession motion of $\mathbf{M}$ precessing around the axis with the same direction of $\mathbf{H}_{\mathrm{eff}}$. The second term $\mathbf{M}\times\mathbf{M}\times\mathbf{H}_{\mathrm{eff}}$ (red arrow in Fig. 2.1) is a dissipative damping term pointing from the $\mathbf{M}$ to the $\mathbf{H}_{\mathrm{eff}}$, as the direction suggests this term is trying to align the $\mathbf{M}$ with the $\mathbf{H}_{\mathrm{eff}}$ by damping.

2.2 Principal interactions in continuum representation

The total field $\mathbf{H}_{\mathrm{eff}}$ in Eq. (2.1) is the net interaction coming from various interactions, expressed as

$$\mathbf{H}_{eff} = \mathbf{H}_{\mathrm{ms}} + \mathbf{H}_{\mathrm{ex}} + \mathbf{H}_{\mathrm{an}} + \mathbf{H}_{\mathrm{ap}} + ... \qquad (2.2)$$

The first four RHS interactions are magnetostatic interaction $\mathbf{H}_{\mathrm{ms}}$, exchange interaction $\mathbf{H}_{\mathrm{ex}}$, magneto-crystalline anisotropy interaction $\mathbf{H}_{\mathrm{an}}$, and Zeeman, also called applied field, interaction $\mathbf{H}_{\mathrm{ap}}$. These fields can be derived from the partial derivative of the magnetic energy E with respect to the $\mathbf{M}$

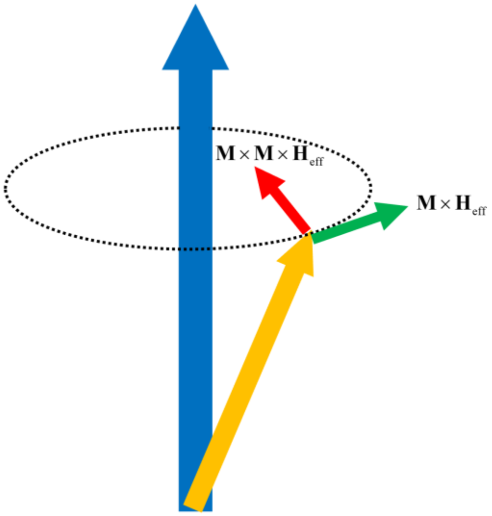

$\mathbf{M}\times\mathbf{M}\times\mathbf{H}_{\text{eff}}$
$\mathbf{M}\times\mathbf{H}_{\text{eff}}$

Figure 2.1: Illustration of how $\mathbf{H}_{\text{eff}}$ (blue arrow) interacts with $\mathbf{M}$, the precession term $\mathbf{M}\times\mathbf{H}_{\text{eff}}$ is the green arrow that keeps the $\mathbf{M}$ revolving, and the damping term $\mathbf{M}\times\mathbf{M}\times\mathbf{H}_{\text{eff}}$ is the red arrow that tries to align the $\mathbf{M}$ with $\mathbf{H}_{\text{eff}}$. For most practical magnetic materials damping effect always exists.

We first give an introduction to the four most common interactions that broadly exist in many kinds of magnetic systems. We then give an overview of other kinds of interactions that may exist in certain types of magnetic systems.

2.2.1 Magnetostatic interaction

To understand the Magnetostatic interactions [4], we first introduce the concept of a magnetic monopole. It is also known as magnetic charge, a similar concept as a counterpart of electric charge. Just like the electric charge, we can define the volumetric magnetic charge and surface magnetic charge

$$\begin{aligned} q(\mathbf{r}) &= \nabla\cdot\mathbf{M}(\mathbf{r}) \\ \sigma(\mathbf{r}) &= -\hat{\mathbf{n}}(\mathbf{r})\cdot\mathbf{M}(\mathbf{r}) \end{aligned} \tag{2.3}$$

Here, $\hat{\mathbf{n}}$ is the normal direction to the surface of the magnetic domain. These charges, then, result in the so-called magnetic scalar potential that is similar to the electric scalar potential

$$u(\mathbf{r}) = \iiint_{V'} \frac{q(\mathbf{r}')}{4\pi|\mathbf{r}-\mathbf{r}'|} dV' + \oiint_{S'} \frac{\sigma(\mathbf{r}')}{4\pi|\mathbf{r}'-\mathbf{r}'|} dS' \tag{2.4}$$

With the magnetic potential, the magnetostatic field can be calculated as

$$\mathbf{H}_{\text{ms}}(\mathbf{r}) = -\nabla u(\mathbf{r}) \tag{2.5}$$

Combining Eq. (2.3), Eq. (2.4) and Eq. (2.5), we have the complete form of $\mathbf{H}_{\text{ms}}$ with respect to $\mathbf{M}$ as

$$\mathbf{H}_{\text{ms}}(\mathbf{r}) = -\nabla(\iiint_{V'} \frac{\nabla\cdot\mathbf{M}(\mathbf{r}')}{4\pi|\mathbf{r}-\mathbf{r}'|} dV' - \oiint_{S'} \frac{\hat{\mathbf{n}}(\mathbf{r}')\cdot\mathbf{M}(\mathbf{r}')}{4\pi|\mathbf{r}'-\mathbf{r}'|} dS') \tag{2.6}$$

A special care needs to take in Eq. (2.3), as indicated in Maxwell equation that there is no standalone magnetic monopoles, neutrality condition has to be enforced in Eq. (2.3), namely

$$\iiint_{V'} q(\mathbf{r}) dV' + \oiint_{S'} \sigma(\mathbf{r}) dS' = 0 \tag{2.7}$$

The neutrality condition must (2.7)hold for any magnetic systems as an intrinsic constraint.

For many practical micromagnetic problems, evaluating $\mathbf{H}_{\text{ms}}$ is computationally expensive, as the time complexity of superposition sum in Eq. (2.4) scales quadratically with growing problem size. Consequently, it has a major impact on the overall micromagnetic simulation performance. There exist a set of approximate methods to handle the superposition sum efficiently with a desired error level, e.g., fast multipole method (FMM), non-uniform fast Fourier transform (NUFFT) and so on. We adopt the NUFFT method as our core for evaluating $\mathbf{H}_{\text{ms}}$ [5].

2.2.2 Exchange interaction

The exchange interactions appear due to a quantum effect that originates from the Pauli repulsion [6]. In the Heisenberg model [7], the energy $E_{i,j}$ arises from the exchange interaction between spin $\mathbf{S}_i$ and $\mathbf{S}_j$ can be denoted as

$$E_{i,j} = -J_{\mathrm{ex}} \times \mathbf{S}_i \cdot \mathbf{S}_j \tag{2.8}$$

The J_{ex} is the exchange integral that determines the coupling strength through its module and spin configuration through the sign. A positive J_{ex} means the parallel alignment is preferred between spin $\mathbf{S}_i$ and $\mathbf{S}_j$, and a negative J_{ex} signals preference over anti-alignment. In magnetism, the parallel alignment indicates the ferromagnetism as shown in Fig. 2.2(a), and the anti-parallel configuration can either represent the anti-ferromagnetism or ferrimagnetism, shown in Fig. 2.2(b) and Fig. 2.2(c), respectively, depending on whether the net magnetization is 0 or not.

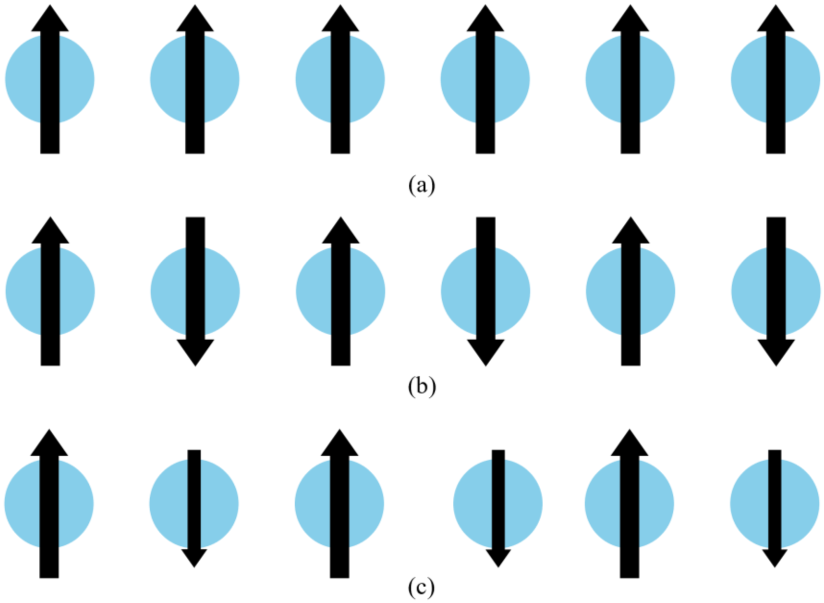


Figure 2.2: Illustration of magnetization with different J_{ex}. (a) Ferromagnetism with $J_{\mathrm{ex}} > 0$, (b) anti-ferromagnetism with $J_{\mathrm{ex}} < 0$, the net magnetization is equal to 0 and (c) ferrimagnetism with $J_{\mathrm{ex}} < 0$, the net magnetization is larger than 0.

In micromagnetics, the exchange interactions are calculated via the divergence of the gradient of the magnetization, i.e. Laplacian operator, denoted as [8]

$$\mathbf{H}_{\mathrm{ex}}(\mathbf{r}) = \frac{2A_{\mathrm{ex}}}{M_s^2(\mathbf{r})} \nabla^2 \mathbf{M}(\mathbf{r}) \tag{2.9}$$

Here A_{ex} is the exchange constant related to the previous J_{ex} in Eq. (2.8). It is an important parameter that reflects the coupling strength within the magnetic material. Based on that we can defined an exchange length as [9]

$$l_{\mathrm{ex}} = \frac{\sqrt{2A_{\mathrm{ex}}}}{M_s} \tag{2.10}$$

The exchange length characterizes the geometry scale at which we can assume the magnetization in Eq. (2.9) to vary slowly. In micromagnetic simulations, the mesh size of the model must be at least of this scale, namely this value determines the lowest bound for mesh size from the perspective of exchange interaction.

The above exchange interaction is the bulk exchange interaction that comes from Pauli repulsion. There is another kind of exchange interaction that comes from the Ruderman-Kittel-Kasuya-Yosida (RKKY) interaction, often referred to as surface exchange interaction [10].

2.2.3 Magneto-crystalline anisotropy interaction

Magneto-crystalline anisotropy interaction refers to the dependence of a material's magnetic properties on its crystallographic structure. This interaction is critical for understanding and designing materials where directional magnetic properties are essential.

The origin of magneto-crystalline anisotropy interaction lies in the spin-orbit coupling (SOC) in a crystal lattice, which causes the energy of the system to depend on the direction of the magnetization relative to the crystal axes. This interaction significantly impacts the magnetization behavior, influencing its coercivity, remanence, and overall magnetic stability. Materials with high anisotropy are often used in applications requiring stable magnetic states, such as permanent magnets and magnetic storage devices.

One of the most common magneto-crystalline anisotropy is the uniaxial anisotropy, which defines a uniaxial preferred direction for magnetization. Such a direction is also called as soft axis. Such anisotropy can be roughly described as [11]

$$\begin{aligned} E_{\text{an}}^{\text{uni}} &= K^U V \sin^2\theta \\ \mathbf{H}_{\text{an}}^{\text{uni}} &= \frac{2K^U}{M_s}\cos\theta\hat{\mathbf{k}} \end{aligned} \tag{2.11}$$

Here, K^U is the uniaxial anisotropy energy density and V is the volume, θ is the angle between $\mathbf{M}$, and soft axis defined by $\hat{\mathbf{k}}$. We can observe $\mathbf{H}_{\text{an}}^{\text{uni}}$ has its maximum when $\mathbf{M}$ is parallel to $\hat{\mathbf{k}}$ and its minimum when $\mathbf{M}$ is perpendicular to $\hat{\mathbf{k}}$.

Another common anisotropy is cubic anisotropy with three easy axes $\hat{\mathbf{x}},\hat{\mathbf{y}},\hat{\mathbf{z}}$. We use $\theta_x,\theta_y,\theta_z$ to denote the angles between $\mathbf{M}$ and each easy axis, then we can express the cubic anisotropy as [11]

$$\begin{aligned} E_{\text{an}}^{\text{cub}} &= K^C V(\cos^2\theta_x\cos^2\theta_y+\cos^2\theta_y\cos^2\theta_z+\cos^2\theta_z\cos^2\theta_x) \\ \mathbf{H}_{\text{an}}^{\text{cub}} &= -\frac{2}{M_s}\begin{bmatrix} K^C\cos\theta_x(\cos^2\theta_y+\cos^2\theta_z) \\ K^C\cos\theta_y(\cos^2\theta_z+\cos^2\theta_x) \\ K^C\cos\theta_z(\cos^2\theta_x+\cos^2\theta_y) \end{bmatrix} \end{aligned} \tag{2.12}$$

Similar to Eq. (2.11), the K^C is the cubic anisotropy energy density.

Another kind of anisotropy arises from the object shape, which is described by a 3×3 shape tensor $\mathrm{N_d}$. This shape anisotropy is significant for certain type of shapes, e.g. thin film, long cylinders etc.

2.2.4 Zeeman interaction

The Zeeman interactions represent the influence of an externally applied magnetic field. This fundamental micromagnetic interaction is critical in manipulating and controlling the magnetic properties of a system.

Zeeman interactions involve the energy coupling between the magnetization and an external applied field. The energy associated with the Zeeman interaction can be expressed as:

$$E_{\mathrm{zee}} = -\mathbf{M} \cdot \mathbf{H}_{\mathrm{ap}} \tag{2.13}$$

This interaction tends to align the magnetic moments with the external field, minimizing the system energy. It is easy to see that the E_{zee} arrives at its minimum when $\mathbf{M}$ is parallel to $\mathbf{H}_{\mathrm{ap}}$.

The alignment of magnetic moments due to the Zeeman effect is critical in applications such as magnetic recording, where external fields are used to set the orientation of magnetic domains. In simpler terms, the Zeeman interaction can dictate the direction of the magnetization, making it an important tool in data storage technologies.

2.2.5 Other interactions

Apart from the four common interactions discussed above, there are other kinds of interactions that may be important in certain kinds of magnetic systems. Examples are the Dzyaloshinskii-Moriya Interaction (DMI), spin-transfer torque, thermal effects, magnetoelastic coupling and so on. This section briefly introduces these additional interactions as a supplement to the above four interactions.

DMI is a result of SOC in systems with a broken inversion symmetry. This interaction is critical in the formation of chiral magnetic structures, such as skyrmions. DMI leads to a preference for non-collinear alignment of neighboring magnetic moments, which can stabilize complex magnetic textures with unique rotational senses. This interaction is particularly significant in thin films and multilayer structures where interface-induced symmetry breaking is prevalent.

Spin-transfer torque (STT) is a phenomenon where the angular momentum of conduction electrons is transferred to local magnetic moments, thereby exerting a torque on them. This effect is fundamental in spintronic devices and has applications in magnetic random-access memory (MRAM) and spin-torque oscillators. STT allows for the manipulation of magnetic states without the need for external magnetic fields, instead using spin-polarized electric currents, which can lead to more energy-efficient magnetic storage technologies. STT can be modeled within LLG equation by adding an extra Slonczewski term, which often is referred as LLGS equation [12]. An extended method is Zhang-Li model [13] that accounts for both adiabatic and non-adiabatic contributions to the torque, capturing more complex dynamics of magnetization, especially in structures with non-uniform geometries and current distributions.

Thermal fluctuations play a crucial role in micromagnetic dynamics, particularly in nanoscale systems [14]. Temperature affects the stability of magnetic configurations and can induce transitions between different magnetic states. The random thermal motion of spins, quantified by the stochastic LLG equation, is essential for understanding phenomena, such as thermal magnetization reversal and the behavior of magnetic materials at finite temperatures.

Magnetoelastic interactions describe the coupling between magnetic and elastic properties of a material [15]. Strain in magnetic materials can influence their magnetic anisotropy, and conversely, the magnetic state can affect the mechanical strain state. This interaction is crucial in magnetostrictive materials, where magnetic fields can induce strains, and in applications like sensors and actuators, where the mechanical response to magnetic fields is utilized.

Other less common but still noteworthy interactions include magneto-optical effects, where the magnetic state influences the optical properties of materials, and vice versa. Such interactions are essential in devices like isolators and circulators in optical communication systems.

In conclusion, while the primary micromagnetic interactions form the backbone of understanding magnetic behavior, these additional interactions enrich the complexity and functional possibilities of magnetic materials in advanced technological applications.

2.3 Spin waves in magnetic materials

Spin waves are collective excitations of the spins in a magnetic material [16], representing a wave-like magnetization perturbation propagating through the medium. Spin waves arise due to the exchange interaction and magnetic dipole, i.e., magnetostatic, interaction between neighboring spins, leading to oscillations around their equilibrium states. Spin waves play a critical role in various applications, such as magnetic storage, spintronics, and magnonic devices, as they can carry energy and information without electrical currents, thus reducing heat dissipation.

Spin waves can be described by the LLG equation (Eq. (2.1)), which governs the dynamics of the magnetization. For small deviations from equilibrium, the linearized LLG equation leads to the description of spin wave phenomena, including their propagation characteristics, dispersion relations, and interaction with external fields [16].

2.3.1 Categories of spin waves

Spin waves can be categorized based on their propagation modes, excitation mechanisms, and their interaction with the material’s structure and geometry. Each category offers unique characteristics, making them suitable for specific applications in spintronics, magnonics, and microwave devices [17].

2.3.1.1 Bulk Spin Waves

Bulk spin waves propagate through the entire volume of the magnetic material and are primarily characterized by their excitation source and the dominant interactions [16].

Exchange spin waves arise due to the short-range exchange interaction between neighboring spins. These waves exhibit:

Short Wavelengths: The wavelength may be on the order of atomic spacings, resulting in high spatial resolution for potential applications.

High Frequencies: Exchange interaction generates spin waves with frequencies in the gigahertz to terahertz range.

Strong Localization: The short wavelength and high-frequency nature make them highly localized within the material, ideal for high-density magnonic devices.

Exchange spin waves are leveraged in high-speed magnonic processors and nanoscale communication devices.

Dipolar spin waves are governed by long-range magnetic dipole-dipole interactions. They exhibit:

Long Wavelengths: Their wavelengths can range from micrometers to millimeters, making them suitable for macroscopic wave manipulation.

Lower Frequencies: Frequencies are typically in the megahertz to low gigahertz range, enabling efficient coupling with microwave systems.

Weaker Localization: The longer wavelength ensures their propagation over larger distances within the material.

Dipolar spin waves are used in non-volatile memory and magnon-based interconnects for long-range signal transfer.

2.3.1.2 Surace Spin Waves

Surface spin waves (SSW) are confined to the surface or interface of a magnetic material. They decay exponentially as they penetrate the bulk, making them highly sensitive to surface conditions [16]. Key characteristics include:

Exponential Decay: The amplitude of these waves decreases with depth from the surface, providing unique confinement properties.

Enhanced Sensitivity: Surface spin waves are particularly sensitive to surface defects, external fields, and changes in boundary conditions.

Directional Propagation: Their propagation direction can be influenced by the magnetic anisotropy and external field.

Surface spin waves are employed in sensors, magnonic crystals, and edge-mode magnonics for efficient waveguiding.

2.3.1.3 Backward volume spin waves

Backward volume spin waves (BVSW) are a category where the group velocity (the energy propagation direction) and phase velocity (the wavefront propagation direction) are opposite [16]. Key features include:

Non-Reciprocal Behavior: This characteristic is particularly useful in isolators and circulators where unidirectional wave propagation is required.

Geometrical Dependence: The properties of backward volume spin waves strongly depend on the material geometry and the orientation of the applied magnetic field.

Backward volume spin waves find applications in designing magnonic waveguides and isolators, enabling energy-efficient wave transmission.

2.3.1.4 Other spin waves

There are other kinds of spin waves, e.g., localized spin waves, which can be seen in the spin-Hall oscillator (SHO) or spin-torque oscillator (STO) devices [18]. In real-world scenarios, multiple types of spin waves may coexist or interact. For example, exchange and dipolar spin waves can hybridize, leading to mixed modes with unique properties. The ability to understand and control these interactions is critical for designing advanced spintronic and magnonic devices. This detailed categorization lays the foundation for exploring the rich physics of spin wave phenomena and their applications in emerging technologies.

2.3.2 Dispersion relationship

In this section we introduce the dispersion relationship of magnetostatic spin wave (MSW), more specifically, we focus on three fundamental types of MSW: Magnetostatic Surface Spin Waves (MSSW), Magnetostatic Backward Volume Waves (MSBVW), and Magnetostatic Forward Volume Waves (MSFVW), arise depending on the relative orientation of the magnetization vector ($\mathbf{M}$), the wavevector ($\mathbf{k}$), and the geometry of the system [17].

2.3.2.1 MSSW

MSSWs occur when $\mathbf{M}$ is perpendicular to $\mathbf{k}$, and both lie in the plane of a thin magnetic film. These waves propagate along the surface of the material, with their amplitude decaying exponentially into the bulk. The dispersion relationship can be shown as [17]

$$\omega^2(k)=\gamma^2\mu_0^2(\mathbf{H}_{\text{app}}+\lambda_{\text{ex}}M_s k^2)\times[(\mathbf{H}_{\text{app}}+M_s)+\lambda_{\text{ex}}M_s k^2]+\frac{\gamma^2\mu_0^2 M_s^2}{4}(1-e^{-2kd}), \qquad (2.14)$$

here the γ is the gyromagnetic ratio, μ_0 is the vacuum permeability, M_s is the saturation magnetization, and $\lambda_{\text{ex}}=\dfrac{2A_{\text{ex}}}{\mu_0 M_s^2}$ is the exchange length and the A_{ex} is the exchange constant. The $\mathbf{H}_{\text{app}}$ is the external applied magnetic field along the direction of $\mathbf{M}$.

2.3.2.2 MSBVW

MSBVWs occur when the $\mathbf{M}$ is parallel to $\mathbf{k}$, and both lie in the plane of a thin magnetic film. This configuration allows these waves to propagate through the bulk of the material. The dispersion relationship can be expressed as

$$\omega^2(k)=\gamma^2\mu_0^2(\mathbf{H}_{\text{app}}+\lambda_{\text{ex}}M_s k^2)\times(\mathbf{H}_{\text{app}}+\lambda_{\text{ex}}M_s k^2+M_s\frac{1-e^{-kd}}{kd}). \qquad (2.15)$$

2.3.2.3 MSFVW

MSFVWs occur when $\mathbf{k}$ is perpendicular to $\mathbf{M}$ [17]. The difference is that the $\mathbf{M}$ is perpendicular to the film and $\mathbf{k}$ lies within the film. This configuration allows these waves to propagate through the bulk, similar to MSBVWs. The dispersion relationship of MSFVW is [16]

$$\omega^2(k) = \gamma^2 \mu_0^2 [(\mathbf{H}_{\text{app}} - M_s) + \lambda_{\text{ex}} M_s k^2] \times [(\mathbf{H}_{\text{app}} - M_s) + \lambda_{\text{ex}} M_s k^2 + M_s (1 - \frac{1 - e^{-kd}}{kd})]. \quad (2.16)$$

Magnetostatic spin waves offer diverse propagation characteristics determined by the relative orientation of the magnetization vector and wavevector. MSSWs can be used in surface-constrained, non-reciprocal applications, while MSBVWs and MSFVWs enable bulk propagation with backward and forward dispersion, respectively. The unique properties of these modes, including their dispersion relationships and field-dependent behaviors, provide a foundation for designing magnonic devices. By leveraging the propagation dynamics of each mode, we can develop technologies for signal processing, logic circuits, and next-generation spintronic systems.

Coherent X-ray Diffraction Imaging (CXDI) is an approach that uses a set of X-ray images to reconstruct 3D objects at a nanoscale resolution [19]. CXDI can resolve volume information of an object thus offering a non-invasive way to identify the object's interior [20]. CXDI finds many applications in physics, chemistry, and biology [21–28]. With refined experiment techniques [29,30] and algorithms [31,32], CXDI is expected to play an increasingly important role across different disciplines.

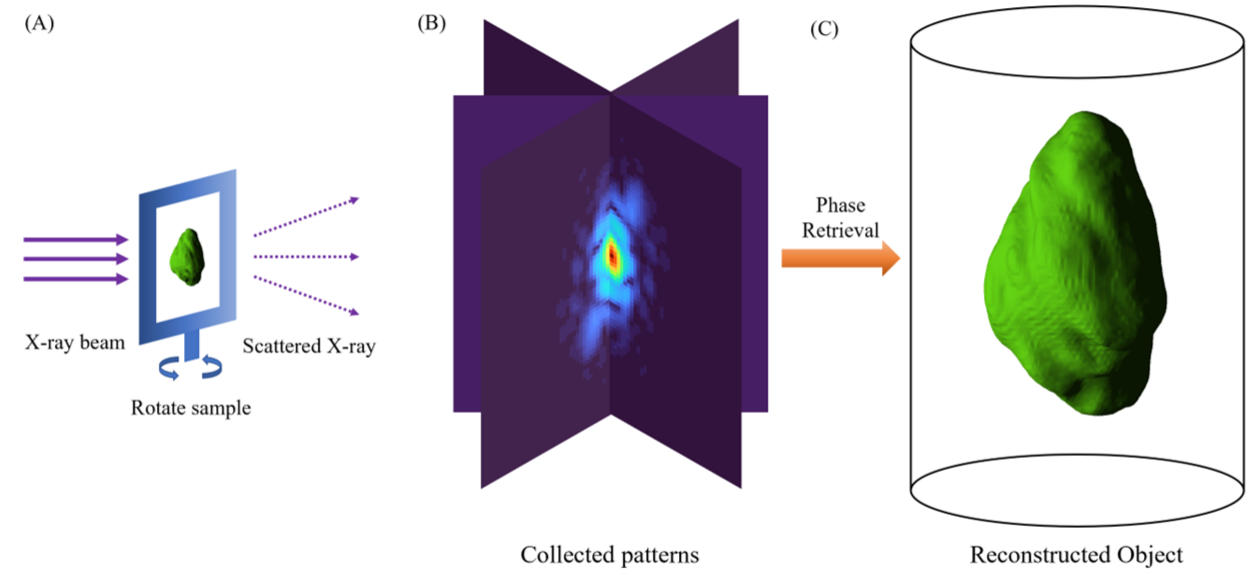


Figure 3.1: (A) A sample holder containing a cell rotated to different angles with coherent X-ray beam impinges on this cell, diffraction patterns are measured. (B) A three-dimensional diffraction pattern was assembled from two-dimensional diffraction patterns. The three-dimensional structure of the cell (C) was iteratively reconstructed.

During an experiment, a coherent X-ray beam is diffracted by the object and a charge-coupled device (CCD) sensor collects the diffraction patterns as 2D images. For 3D object imaging, the object is rotated by a set of angles and the corresponding set of 2D images is collected. This procedure can be treated as a Fourier transform (FT) of the object as Fraunhofer's diffraction (Fig. 3.1). In principle, the object can be reconstructed by an inverse FT. However, the CCD sensor only collects the intensity whereas the phase information is lost. CXDI uses an iterative procedure based on forward and inverse FTs, which allows extracting the phase information from the intensity measurements. The ability to obtain both the intensity and phase leads to a high-resolution reconstruction [33].

Typically, 3D CXDI reconstruction is based on interpolating the 3D image (reciprocal) domain to create a 3D image in a regular set of pixels and then performing a 3D iterative phase retrieval algorithm [33,34]. Often, many iterations are required. Each iteration involves 3D forward and inverse FTs, which are computationally intensive. For large images, CXDI computational time can be large, e.g., it can take several days. Additionally, for large images, CXDI can require much memory, which complicates implementations on massively parallel systems, such as Graphics Processing Units (GPUs). These factors make it nearly impossible to reconstruct objects in real time during the experiment. While with the improving experimental techniques and apparatus, the available collected data becomes larger, the CXDI reconstruction procedure becomes a bottleneck. Highly efficient CXDI algorithms and their numerical implementations are essential to allow for real-time CXDI.

In this chapter, we introduce a "carousel" phase retrieval algorithm (CPRA). Instead of the direct 3D reconstruction, CPRA first retrieves the phases of the collected 2D diffraction patterns and then uses this information to reconstruct the 3D object. CPRA resolves shortcomings of inconsistencies of the 2D diffraction pattern reconstructions of related to approaches, and thus allows for a rapid convergence. CPRA addresses the diffraction pattern inconsistency by a two-level iterative algorithm. In the first iteration level, the 2D patterns are pre-reconstructed starting from the first 2D pattern, such that each adjacent 2D pattern is reconstructed based on the previous patterns until the last pattern returns to the first one, hence the name "carousel". In the second level, the iterations are over mixed 2D patterns obtained in the first iteration level. The phase reconstruction of the 2D diffraction patterns is much more efficient than the direct 3D reconstruction and it can be accomplished in parallel. The final 3D reconstruction is highly efficient and accurate because of the availability of the phase information. The benefits of CPRA include a major reduction of the number of iterations needed for convergence, use of 2D rather than 3D FTs, a major reduction of the computational time and memory consumption, amenability to massive parallelization, increased accuracy, and improved robustness. CPRA is well-suited for implementations on GPUs with major a performance enhancement. CPRA allows for a high-resolution reconstruction in real time during experiments.

3.1 Formulation of algorithm

For 3D CXDI reconstruction, to acquire sufficient information for the 3D image domain, an object is rotated during the experiment at a sequence of angles θ_n corresponding to incident wave vectors $\mathbf{q}_{i,n}$, where $n = 1,...,N_p$ and N_p is the total number of the angles chosen to be large enough for sufficient angular resolution (Fig. 3.1). For each incident wave vector $\mathbf{q}_{i,n}$, a scattered intensity map $I(\mathbf{q}_{t,n}, \mathbf{q}_{i,n})$ of $N \times N$ pixels are collected. The map corresponds to a set of transverse wave vector components $\mathbf{q}_{t,n} = \mathbf{q} - \mathbf{q}_{i,n}$. Each $I(\mathbf{q}_{t,n}, \mathbf{q}_{i,n})$ is related to complex-valued spectral content $E(\mathbf{q}_{t,n}, \mathbf{q}_{i,n})$ via $I(\mathbf{q}_{t,n}, \mathbf{q}_{i,n}) = | E(\mathbf{q}_{t,n}, \mathbf{q}_{i,n}) |^2$. The result of the measurement is N_p intensities, each for $N \times N$ pixels. A high reconstruction resolution requires N_p and N to be large. The goal is to reconstruct the object $O(\mathbf{r})$ in the real space, where $\mathbf{r}$ is the real space vector and $O(\mathbf{r})$ is the electron density of the sample or spin densities of magnetic domain. The object can be reconstructed since it is related to the inverse FT of the measured intensities. However, the measured intensities do not contain the phase information and, therefore, it needs to be recovered via a phase retrieval algorithm.

We now present an alternative greatly efficient CPRA with the aid of Fourier slice theorem. We represent the measured 2D intensity for a given incident wavevector $\mathbf{q}_{i,n}$ as

$$I(\mathbf{q}_{t,n},\mathbf{q}_{i,n}) = \left|E(\mathbf{q}_{t,n},\mathbf{q}_{i,n})\right|^2 = \left|\hat{\mathbf{q}}_{i,n}\cdot\int\left[\int O(\mathbf{r})e^{i\mathbf{q}\cdot\mathbf{r}}d\mathbf{r}\right]d\mathbf{q}\right|^2, \tag{3.1}$$

where we have two integrals: an integral inside the square bracket is a 3D FT transforming the object $O(\mathbf{r})$ into its spectral content $E(\mathbf{q})$ and an outside integral over $d\mathbf{q} = \hat{\mathbf{x}}dq_x + \hat{\mathbf{y}}dq_y + \hat{\mathbf{z}}dq_z$, which is projected along the unit vector $\hat{\mathbf{q}}_{i,n}$ of $\mathbf{q}_{i,n}$.

Based on Eq. (3.1), we can use the Fourier slice theorem, which, in this case, states that a 3D FT followed by a 3D to 2D dimension projection in the reciprocal space is equivalent to 3D to 2D dimension projection in the real space followed by a 2D FT. Hence, we can rewrite Eq. (3.1) as

$$I(\mathbf{q}_{t,n},\mathbf{q}_{i,n}) = \left|E(\mathbf{q}_{t,n},\mathbf{q}_{i,n})\right|^2 = \left|\int\left[\hat{\mathbf{q}}_{i,n}\cdot\int O(\mathbf{r})d\mathbf{r}\right]e^{i\mathbf{q}_{t,n}\cdot\mathbf{r}_{t,n}}ds_{t,n}\right|^2, \tag{3.2}$$

where $\mathbf{r}_{t,n} = \mathbf{r} - (\hat{\mathbf{q}}_{i,n}\cdot\mathbf{r})\hat{\mathbf{q}}_{i,n}$ is the coordinate and $ds_{t,n}$ is the surface differential in the plane transverse to the $\hat{\mathbf{q}}_{i,n}$ direction. Equation (3.2) now has a 3D to 2D real-space projection inside the brackets, which is followed by a 2D FT. Let us regard the result in the square bracket as a new 2D projected object $O'(\mathbf{r}_{t,n},\mathbf{q}_{i,n}) = \hat{\mathbf{q}}_{i,n}\cdot\int O(\mathbf{r})d\mathbf{r}$ and formulate Eq. (3.2) in a 2D form as

$$I(\mathbf{q}_{t,n},\mathbf{q}_{i,n}) = \left|E(\mathbf{q}_{t,n},\mathbf{q}_{i,n})\right|^2 = \left|\int O'(\mathbf{r}_{t,n},\mathbf{q}_{i,n})e^{i\mathbf{q}_{t,n}\cdot\mathbf{r}_{t,n}}ds_{t,n}\right|^2. \tag{3.3}$$

It follows that we can reconstruct a set of 2D projected objects $O'(\mathbf{r}_{t,n}, \mathbf{q}_{i,n})$ representing the projection of the object at the n^{th} angle along $\hat{\mathbf{q}}_{i,n}$, which correspond to the discrete set of incident directions of $\mathbf{q}_{i,n}$ for the measured angles $\theta_{i,n}$. The reconstruction of each projected object can be accomplished via the conventional CXDI procedure for the 2D case.

Once $O'(\mathbf{r}_{t,n}, \mathbf{q}_{i,n})$ are obtained, the corresponding spectral contents $E(\mathbf{q}_{t,n}, \mathbf{q}_{i,n})$ can be used for reconstructing the 3D object via tomography. The tomography procedure is similar to the conventional approach. An important difference is that unlike in CXDI, where the intensity $I(\mathbf{q})$ is used as the constraint, here, $E(\mathbf{q})$, including its recovered phase is used. The wavevectors required for the 3D reconstruction are obtained via interpolation from the set of reconstructed spectral contents $E(\mathbf{q}_{t,n}, \mathbf{q}_{i,n})$. The extra phase information in $E(\mathbf{q})$ provides a much better constraint, which leads to a much faster convergence and a small number of required initial random guesses.

A major difficulty of using CPRA, is that the 2D reconstructed projected objects for different angles can be inconsistent. The projected objects are obtained from the same object, so they are naturally required to point to the same object, while this is not guaranteed. For example, the projected objects may have slightly different positions. While each reconstructed projected object may be accurate, these displacements between the projected objects may introduce errors when the 2D projected objects are combined for 3D reconstruction. Even if we had a feasible alignment method, the reconstruction error or noise may, often unavoidably, have a detrimental effect on the reconstruction quality. Another even more severe factor is within the iterative phase retrieval algorithm itself. The algorithm requires many different random initial guesses, which are averaged for the final result. Different projected objects may converge to different local optima.

Such different projected objects cannot be combined for the final 3D reconstruction. Maintaining the image consistency within a set of 2D reconstructed projected objects is a key factor for enabling the introduced CPRA. A related approach was presented earlier, which suffered from these inefficiencies [35].

To resolve the 2D projected objects consistency challenge, we use the fact that the 2D projected objects correspond to a set of angles with small consecutive changes and the nearby 2D projected objects are similar. Therefore, we can maintain the reconstruction projected object consistency and quality of an n^{th} projected object by using the n-1$^{\text{th}}$, n+1$^{\text{th}}$, n-2$^{\text{th}}$, n+2$^{\text{th}}$, etc., projected objects.

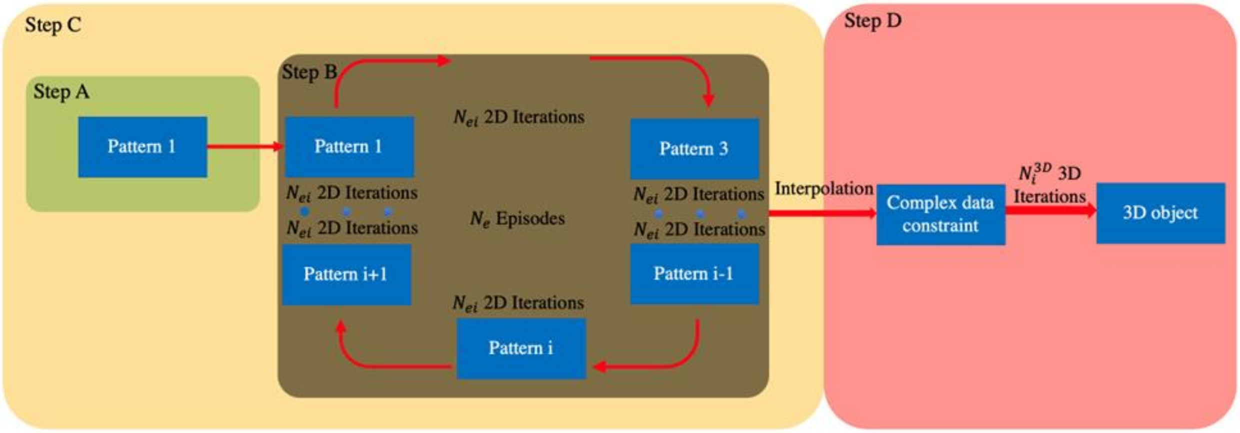


Figure 3.2: Illustration of our routine. The green region represents the step A, pre-reconstruct one projected object. The dark region representing step B, reconstruct multiple episodes sequentially. The yellow region is step C, we repeated step A and B   times and merge some best of them. The pink region is step D, where we interpolate spectral contents to reconstruct the object.

This idea leads to CPRA. Let us define the reconstructed projected objects as $O'_n = O'(\mathbf{r}_{t,n}, \mathbf{q}_{i,n})$ corresponding to the intensity $I_n = I(\mathbf{q}_{t,n}, \mathbf{q}_{i,n})$ and spectral content $E_n = E(\mathbf{q}_{t,n}, q_{i,n})$. Let us call a set of projected objects for all $n = 1, ..., N_p$ as an episode. The n-th intensities and spectral contents of an episode are periodic, such that $n = \ldots, N_p - 1, N_p, 1, 2, \ldots$ correspond to consecutive incident angles. CPRA proceeds with an iterative process, in which an episode W_{m+1} is obtained based on the previous episode W_m via the following four steps, illustrated in Fig. 3.2:

A. **Pre-reconstruction.** Initialize an episode W_0. Pre-reconstruct a single projected object, e.g., for $n=1$, starting with a random initial guess. The pre-reconstruction is accomplished via the conventional algorithm. This pre-reconstruction requires a relatively large number of iterations (N_i^{pre}), similar to the conventional 2D reconstruction. The rest of the projected objects are initialized randomly. This initialization results in a set of projected objects $O_n'^{(0)}$ for the first episode W_0. The computational time of this step scales as $O(N_i^{pre} N^2 \log N)$ and the memory comsumption scales as $O(N^2)$.

B. **Iterative reconstructions of episodes.** Reconstruct a set of episodes W_m for $m=1,...,N_e$, where N_e is the number of episodes. For each episode W_m, the projected objects are reconstructed via the conventional algorithm for the 2D case consecutively, from $n=1$ to $n=N_p$, with the initial guess for each projected object obtained based on the combination of the objects from episode W_{m-1}:

$$O_n'^{(m)} = \sum_{k=-N_p/2-n+1}^{N_p/2-n} \alpha_k O_{k+n+N_p/2}'^{(m-1)} , \tag{3.4}$$

where α_k represents weight coefficients that are periodic with respect to the range of N_p, satisfy $\sum_{k=-N_p/2+1}^{N_p/2} \alpha_k = 1$, and are chosen such that they decrease with an increase of $|k|$. An example choice can be $\alpha_0 = 0.6$, $\alpha_{-1} = \alpha_1 = 0.2$ and $\alpha_k = 0$, otherwise. The fact that the coefficients are periodic implies that each reconstructed object at a certain episode iteration is obtained based on adjacent objects from the previous episode iteration in a carousel (circular) manner. The reconstruction of the projected objects in all the episodes requires a small number of iterations N_{ei} since this reconstruction is based on an approximation from the previous iteration, which is much better than a random guess. The dominant time-consuming part here is the 2D phase retrieval iterations. The computation time scales as $O(N_p N_i^{2D} N^2 \log N)$ and the memory consumption scales as $O(N^2)$, where $N_i^{2D} = N_e \times N_{ei}$ is the total number of 2D iterations for one episode.

C. **Repeating and merging.** Repeat step A and B multiple times, each for different random guesses in step A for the total number of random guesses N_r. We select a certain number $N_l < N_r$ of the best reconstructed episodes, which have the lowest reconstruction error. The projected objects with the same number may be inconsistent between episodes with different random guesses, caused by two factors. The first factor is because the projections of an object rotated along two parallel axes and displaced in the projected plane are the same. The displacement in the projected plane is proportional to the distance between the two axes. To address this issue, we move the mass center of each projected object by $\Delta\mathbf{r}_n = -\int O_n'(\mathbf{r})\mathbf{r}dv$. The second factor comes from the property of FT that the intensity of an object's FT is the same as the centrally inversed object, i.e., $|\int O'_n(\mathbf{r})e^{i\mathbf{q}\cdot\mathbf{r}}d\mathbf{r}| = |\int O'_n(-\mathbf{r})e^{i\mathbf{q}\cdot\mathbf{r}}d\mathbf{r}|$. To address this issue, we use the episode with the lowest reconstruction error among the N_l episodes. Then, we calculate the Euclidean distance between one of the projected objects in this best episode and the same number projected object in other episodes. We also calculate the Euclidean distance for the centrally inverted version of the projected objects in the episodes. We select the version with the lowest Euclidean distance, thus accomplishing the task of selecting the proper central symmetry. After obtaining the properly corrected episodes, we average between them to create the final episode, which is used as a set of the complex spectral contents E_n for the final 3D reconstruction. The most time-consuming part is repeating steps A and B N_r times. Selecting and averaging episodes is only related to elementwise operations. Therefore, the computational time complexity of this step scales as $O(N_r(N_i^{pre} + N_p N_i^{2D})N^2 \log N)$. All parts within this step are 2D operations, thus the memory consumption scales as $O(N^2)$.

D. **Final 3D object reconstruction.** Interpolation and the final 3D reconstruction. We interpolate the final reconstructed spectral content E_n to create a 3D spectral content $E(\mathbf{q})$ required for 3D reconstruction. We, proceed with the conventional CXDI algorithm but using the complex data constraints of $E(\mathbf{q})$ obtained via the steps A-C of CPRA; this procedure can also be regarded as tomography. The phase information in $E(\mathbf{q})$ leads to a rapid convergence and small number of required initial random guesses. Typically, only a single random guess and N_i^{3D} (around 50) iterations are sufficient. The dominant part of this step is the 3D phase retrieval. The time complexity is of $O(N_i^{3D} N^3 \log N)$ and the memory comsumption scales as of $O(N^3)$, if 3D FTs are used directly.

The overall computational time of CPRA is given by step C, and it scales as $O(N_r(N_i^{pre} + N_p N_i^{2D})N^2 \log N)$. The computational time of the conventional approach is due to 3D FTs at each iteration and it scales as $O(N_r N_i^{conv} N^3 \log N)$, where N_i^{conv} is the number of 3D iterations, which may be on the order of 10^3 or even greater. Therefore, CPRA can be much more efficient by having $N_i^{pre} + N_p N_i^{2D} \ll N_i^{conv} N$ especially when the object size N is large. In steps A-C, the memory consumption scales only as $O(N^2)$. Compared with the conventional approach, for which the memory consumption scales as $O(N^3)$, CPRA is more suitable for GPU computing architectures, in which less memory is available. While step D requires $O(N^3)$ memory, it requires a small computational time as compared to the rest of CPRA and it scales as $O(N_i^{3D} N^3 \log N)$. The memory consumption of the final 3D reconstruction may be reduced by performing 3D FTs as a set of 2D FTs, and the corresponding possible increase of the computational time associated with the need for the memory transfers is still insignificant compared to the rest of the CPRA. It shows that we can greatly accelerate CPRA by utilizatiung GPUs even if N is large.

3.2 Validation and performance

For numerical validation, we start by considering an object that is uniformly sampled by $N \times N \times N$ pixels in the object domain. This object contains a lithium-rich layered oxide particle that occupies less than $N/2$ pixels in each dimension to meet the oversampling requirements. The intensities corresponding to what is typically obtained from a synchrotron experiment are simulated by projecting and FT transforming these pixels along different rotating angles. Each 2D intensity map is of size $N \times N$. The number of rotation angles N_p is chosen to scale with N, such that $N/N_p = 16/9$. The 3D intensity map is of size $N \times N \times N$. We use these intensities to reconstruct the object via CPRA and conventional method.

We first compare the CPRA with the conventional Shrinkwrap algorithm (CSWA) when reconstructing the 3D object for a relatively small N, such that CSWA can be used. We, then, present the performance of the CPRA for large N, which is impossible to accomplish with CSWA. For the result verification, we choose $N = 160$ and $N_p = 90$. We use the 2D CSWA method in steps A and B of CPRA, which makes the iterative algorithm identical for CPRA and CSWA. We remove pixels whose values are below a 0.1 relative defogging threshold. For assessing the reconstruction quality, we employ the Phase Retrieval Transfer Function (PRTF) [35], which evaluates the stability for different initial random guesses, and the Fourier Shell Correlation (FSC) [36]between the original and reconstructed objects, which can be done because we have the ground truth original object. From both PRTF and FSC, we can estimate the resolution, denoted as R, of the reconstruction object by taking the cutoff spatial frequency at a threshold value. Similar to other works, we set this threshold as 0.5 [36].

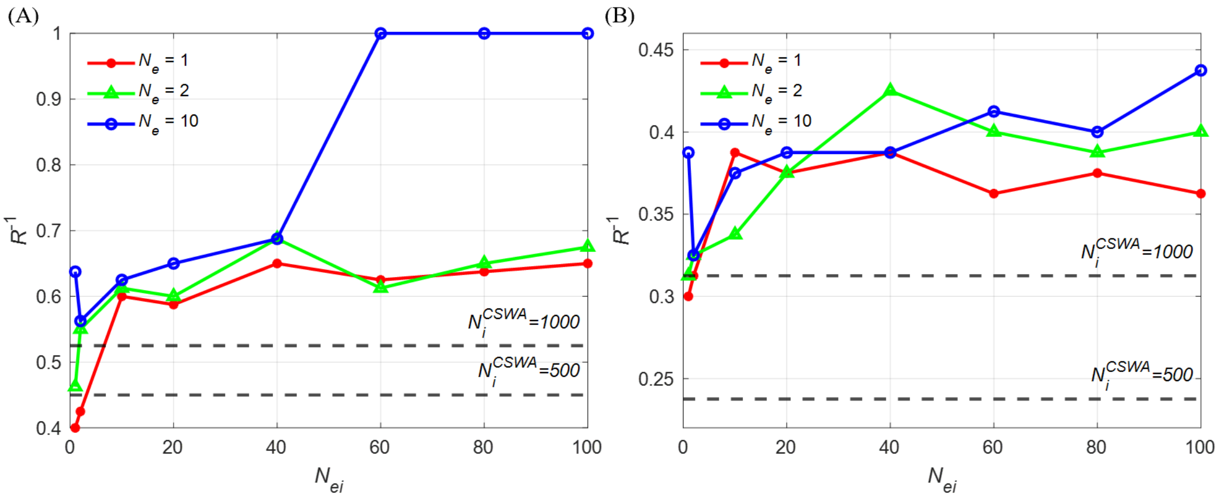


Figure 3.3: Resolution determined by PRTF(A) and FSC(B) of CSWA and CPRA. The dashed lines here are the determined resolutions of CSWA with $N_i^{CSWA} = 500$ and 1000 respectively, the solid lines are determined resolutions of CPRA with different N_e and N_{ei}. CPRA's resolution from FSC are all equal or above CSWA except $N_e = 1$, $N_{ei} = 1$ configuration.

Figures 3.3A and 3.3B show PRTF and FSC as a function of the number of iterations in CPRA. These CPRA results are compared to those from CSWA for 500 and 1000 iterations. Similar to N_i^{conv}, this number can be defined as N_i^{CSWA}. We find that for CSWA with $N_i^{CSWA} = 1000$, we can have more reasonable PRTF and FSC, which is consistent with other publication. CPRA, however, requires a much smaller number of iterations (N_i^{2D}) and has a much better performance. For instance, CPRA achieves the same resolution even when $N_e = 1$, $N_{ei} = 2$, and it achieves a much better resolution with $N_e = 1,\ N_{ei} = 10$. The resolution generally increases with increasing N_{ei}, whereas N_i has a less significant effect.

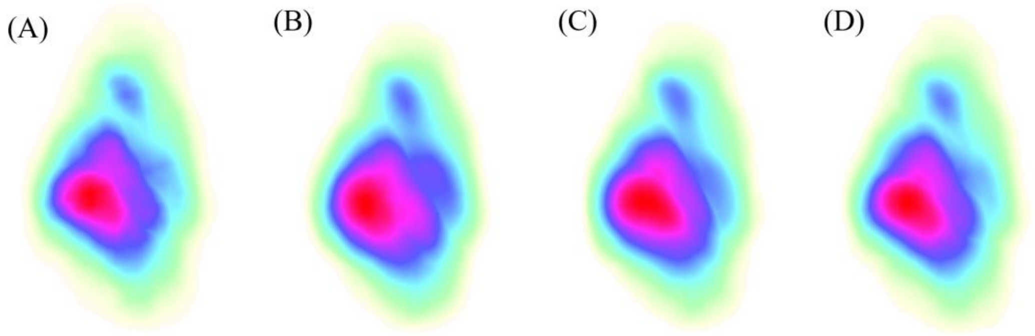


Figure 3.4: Volume density of original object (A), reconstructed object from CSWA (B), CPRA method with $N_e = 1,\ N_{ei} = 2$ (C) and $N_e = 1,\ N_{ei} = 10$ (D). (D) exhibits detached density cloud in the top right corner while (B) and (C) are not. Here (C) is slightly better than (B) since the connection of two density cloud is weaker.

To further compare the reconstruction qualities, we show 3D results (Fig. 3.4) and central slices through *x*-*y*, *y*-*z*, and *z*-*x* planes (Fig. 3.5) for the original and reconstructed objects obtained via the CSWA and CPRA for $N_e = 1,\ N_{ei} = 2$ and $N_e = 1,\ N_{ei} = 10$. It is evident that CSWA is significantly less accurate, even visually, and that CPRA with $N_e = 1,\ N_{ei} = 10$ provides a better quality.

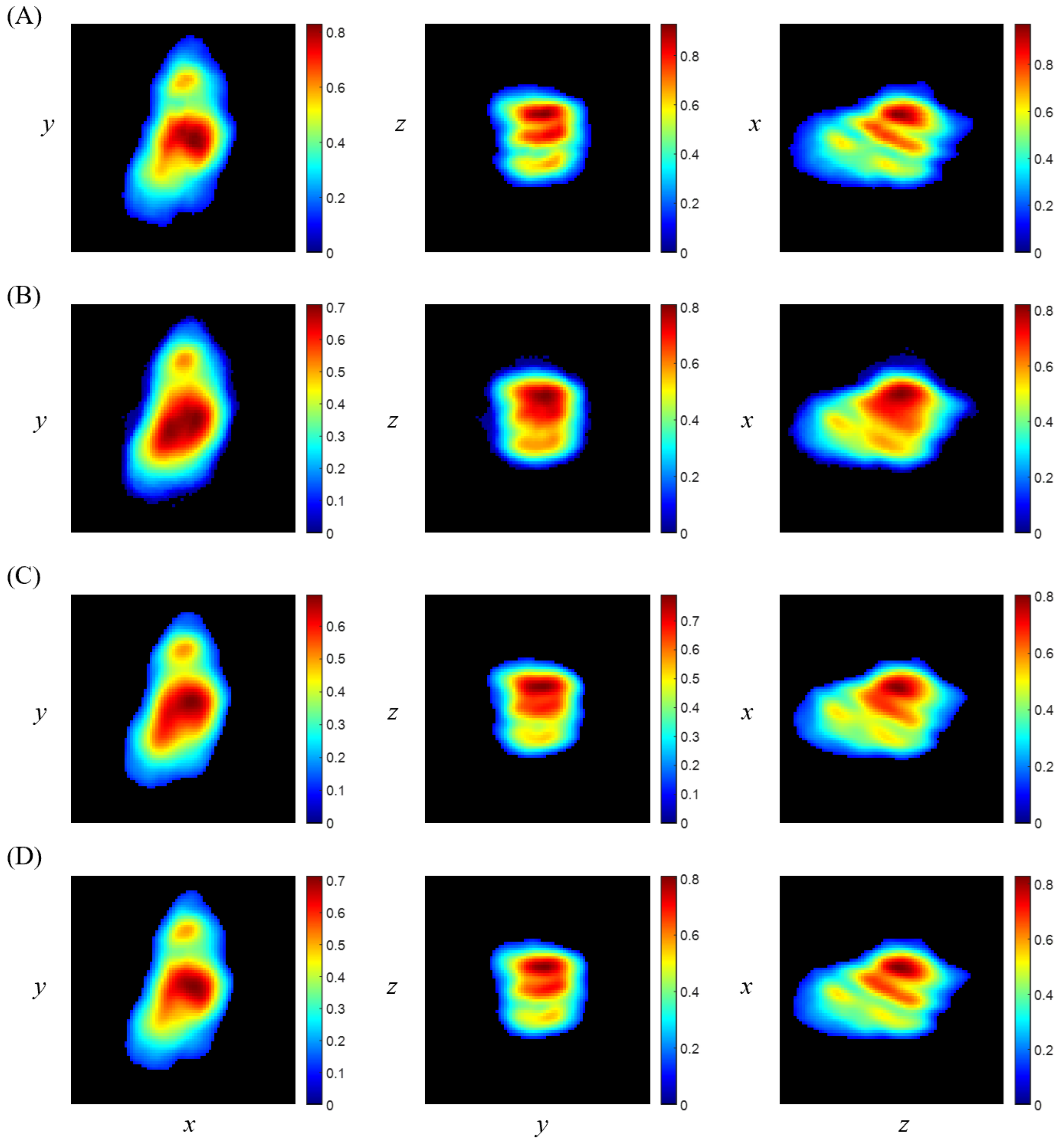


Figure 3.5: Central slices through x-y), y-z , z-x plane, (A) original object, (B) reconstructed object from CSWA, (C) reconstructed object from CPRA with $N_e = 1,\ N_{ei} = 2$, (D) reconstructed object from CPRA with $N_e = 1,\ N_{ei} = 10$. (D) looks most like (A) with many similar features, and (C) is still slightly better than (B) for there're clear boundaries of density.

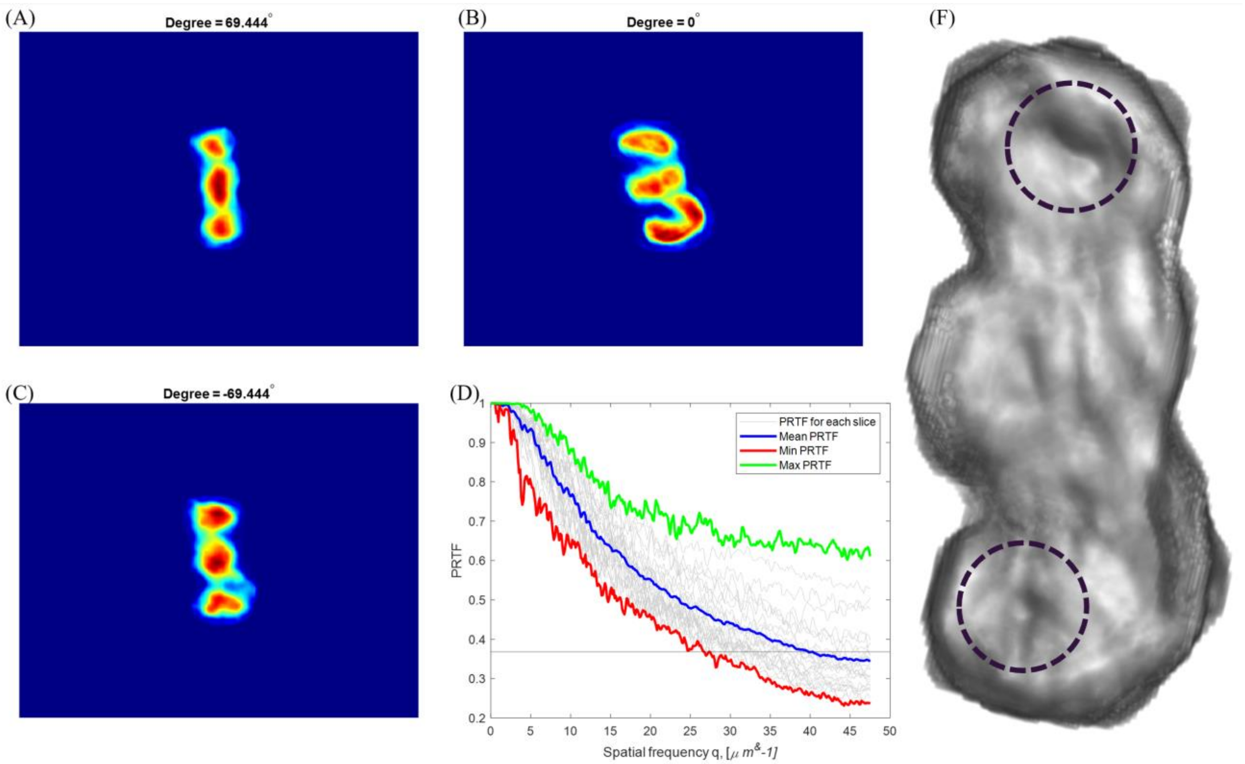


Figure 3.6: Reconstruction of S. aureus cell. (A-C) Reconstructed 2D object. (D) PRTF of all 2D reconstructions (gray lines), mean values (blue line), minimum values (red line) and maximum values (green line). The horizontal line is the 1/e criterion. (D) Reconstructed surface morphology of S. aureus. Black circles indicate the two representative depressions.

We, then, use CPRA with experimental data imaging Staphylococcus aureus (SA) cell [27]. Each 2D intensity map is cropped to size $N \times N$, where $N = 700$, with a beamstop in the center, and there are $N_p = 27$ maps. The space constraints are pre-calculated by the HIO algorithm with the oversampling ratio of 4. To be consistent with the original reconstruction method, we adopt the RAAR algorithm with hyperparameter $\beta = 0.9$. For the pre-reconstruction step (step A), we perform $N_i^{RAAR} = 500$ RAAR iterations on the first intensity map, half of the originally suggested number. We then set $N_e = 2$, $N_{ei} = 10$ for step B to iteratively reconstruct the rest of the 2D intensity maps. At the end of each episode, we perform an extra 2D CSWA iteration to obtain a tighter space constraint for each 2D image. We start from $N_r = 100$ sets of random initial guesses and pick 10 best sets to merge. We project the 2D space constrains back to 3D space to form a relatively tight 3D space constraint and perform 3D reconstruction via GENFIRE package with 50 iterations in total. We shrink the 3D space constraint every 5 iterations, leading to a tighter 3D space constraint close to the shape of the object in the end. We present the 2D reconstructions in Fig. 3.6 (A-C), and the corresponding PRTF curve in Fig. 3.6 (D) with the $1/e$ criterion. In the Fig. 3.6 (D), the mean value (blue line) of all 2D PRTFs suggests a very high resolution close to the achievable limit. The 3D reconstructed object surface morphology is presented in Fig. 3.6 (E), where we find two representative depressions inside the black circles that agree with.

Finally, we compare the computational performance of CPRA and CSWA on CPU and GPU computing architectures. We first consider the computational performance for reconstructing the object in Figs. 3.4-3.5. The computations were performed on a desktop equipped with a 16-core 4.9 GHz AMD R9-5950X CPU and NVIDIA RTX 3080Ti GPU. The system had 128 GB DDR4 memory, and the GPU had 12 GB GDDR6X memory. The shown CPU results were obtained on a single core. The multi-core CPU parallelization efficiency is above 90% for up to 8 cores on this CPU, so multi-core performance can be assessed based on the presented single-core results. The results are shown for $N = 160$, 256, 512, and 1024 with corresponding $N_p = 90$, 144, 288, and 576. For CSWA the number of iterations N_i^{CSWA} was fixed at 1000, which is required for good reconstruction, and for CPRA, we set $N_e = 1$, $N_{ei} = 2$, which, according to the results in Fig. 3.3, gives a similar reconstruction quality. For both CSWA and CPRA, we set $N_r = 100$. For large

N, CSWA can take too much time, hence we run CSWA for only one random initial guess and then multiply this result by N_r.

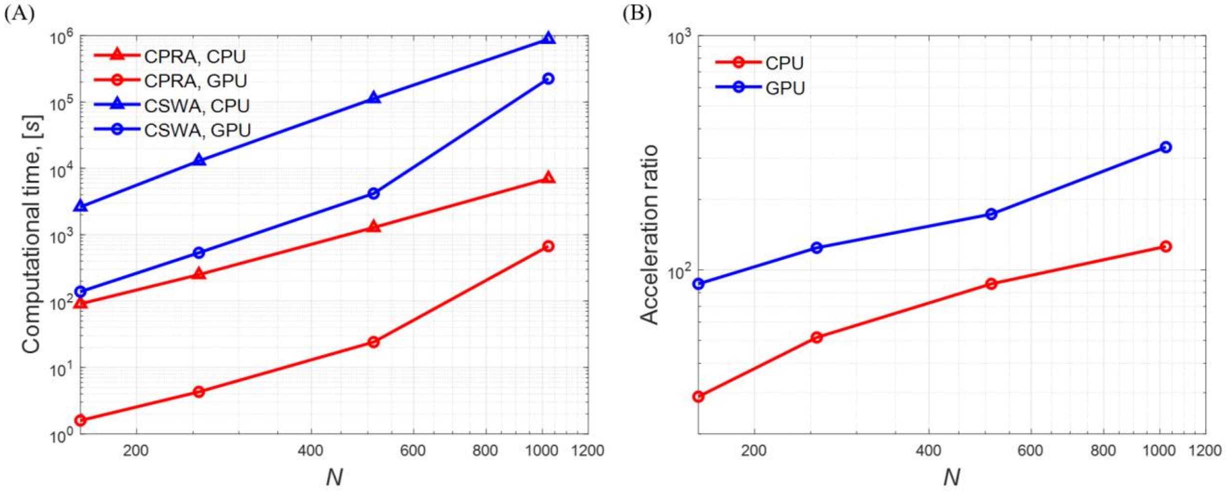


Figure 3.7: Performance of CSWA and CPRA method on different devices. (A) computational time of different methods on CPU and GPU, (B) acceleration ratio across different methods on CPU and GPU, defined by CSWA time consumption over that of CPRA.

Figure 7A presents the computational time for CPU and GPU computations using CPRA and CSWA, and Fig. 7B shows the acceleration ratio of time consumption using CPRA versus CSWA on CPU and GPU. It is evident that CPRA is much faster for any N. The CPRA speed-up is in the range of 100-300 times on GPU and 30-120 times on CPU. The acceleration ratio increases with the problem size, which is related to better parallelization and hardware utilization. As an example, reconstructing a case with $N = 1024$ takes $670s$, which means that such a reconstruction can be performed in real-time during experiment.

We also compared the computational performance of CPRA for reconstructing the SA cell of Fig. 6. This case only required 100 initial guesses and 39 iterations for each 2D intensity map on average. CPRA took less than 20 seconds on Nvidia RTX 3080 Ti, which can be considered as a real time reconstruction. CPRA was around 300 times faster than the conventional method when comparing on the same computing (CPU or GPU) architecture.

3.2 Summary

In this chapter we introduced a highly efficient CPRA for 3D CXDI. CPRA is based on defining episodes of reconstructed 2D projected objects and iterating over the episodes, such that each projected object in an iteration is obtained based on a mixture of the projected objects reconstructed in the previous iteration. Each 2D projected object reconstruction corresponds to a 2D image slice obtained experimentally, and its reconstruction is highly efficient. The final set of the reconstructed 2D objects corresponds to images that contain phase information, so that the final 3D object reconstruction can be accomplished efficiently and accurately via tomographic techniques.

CPRA requires a small number of iterations to achieve a high-quality reconstruction, and even 1-2 iterations may be sufficient. CPRA also uses a significantly smaller amount of memory, which is important to consider large objects and to implement it on GPUs. The numerical comparisons demonstrate that CPRA achieves 100-300 fold speed-ups on GPU, and 30-120 fold speed-ups on CPU with equal reconstructing qualities. Moreover, CPRA allows obtaining a much higher resolution, still being significantly faster. For example, it takes $1.6s$, $4.3s$, $24.1s$ and $670s$ to reconstruct 3D objects with images of linear size of 160, 256, 512, and 1024, respectively. These speeds allow performing 3D CXDI in real time, concurrent with experimental procedures.

CPRA has several benefits that can allow future extensions. CPRA allows concurrent partial reconstruction during the experiment. By noticing that Steps A and B of CPRA are 2D operations that do not require collecting a full episode, reconstructing only a part of the 2D projected objects is possible. Additionally, CPRA can be efficiently implemented on multi-GPU computing systems to provide further speed-ups and to reconstruct larger objects. As an example, the maximal object size can be addressed on a single GPU with 12 GB of memory is 1024. A larger size requires either a greater GPU memory or frequent CPU-GPU data exchange, with an increased computational time. Currently, the largest available single-GPU memory is 80 GB (Nvidia A100) would allow reconstructing images of size around 2000. However, a single-GPU memory is always limited. To handle larger size cases, CPRA can be implemented on multi-GPU systems. A major benefit of CPRA is the ability to reconstruct the 2D objects separately and that only information of a few 2D images is needed for each 2D image. It allows reducing possible cross-GPU communications. For a system contains N_G GPUs, only $2N_G$ communications are required for loading the 2D images. The size of the data transferred between GPUs scales and the corresponding time consumption scale as the projected 2D object size, i.e., $O(N^2)$, which is negligible compared to the reconstruction itself.

Given the benefits of CPRA and its demonstrated computational efficiency, it can become a part of experimental apparatus in synchrotron facilities, where a desktop computer with a GPU can be used to reconstruct objects during the experiment, and thus allow conducting the experiments more efficiently. The real-time CXDI reconstruction can be combined with using numerical simulators to assist performing experiments. For example, micromagnetic modeling tools can be used to predict and assist in choosing experimental procedures for CXDI of magnetic nanostructures.

Acknowledgements

Chapter 3, in full, is a reprint of the material as it appears in F. Ai, O. Shpyrko and V. Lomakin “Real time 3D coherent X-ray diffraction imaging”, which is currently under review in Phys. Rev. Lett. The dissertation author was the primary researcher and author of this paper.

Chapter 4 Spin-wave assisted synchronization in 2D arrays of spin torque oscillators

Spin-transfer nano-oscillators (STNOs) are non-linear oscillators that can convert a DC input current into a microwave voltage output [18,37]. These nanoscale devices have gained significant attention in recent years due to their potential applications in fields such as microwave communications [38,39], signal processing [40,41], and energy-efficient computing [42–44]. One common challenge in these applications is how to synchronize a large number of STNOs to achieve better performance [45–47]. While the theoretical model of mutual synchronization between two STNOs has been well established and confirmed through various experiments, the development of synchronization over more than a pair of STNOs remains a challenging topic in the field. The inherent non-linearity and complicated interactions among a large number of coupled STNOs result in rich and complex dynamical behavior that is difficult to precisely describe [18]. Despite some analysis towards special structures such as a ring structure [48], a comprehensive understanding of the collective behavior of a large 2D array of STNOs is still lacking. There are a quite few experiments based on STNOs or spin-hall nano-oscillators (SHNOs) have demonstrated the emergence of global synchronization over the entire array [47,49], yet they are either of relatively small sizes and lack scalability, or require certain geometry shape to function that do not offer much flexibility in design, resulting in a gap between theory (e.g. a ring of oscillators) and actual tested device (e.g. a square matrix of oscillators). Hence a more comprehensive design that can address these issues is in desperate need.

In this chapter, we present an approach to achieve synchronization in a large 2D array of STNOs using micro-magnetic simulations. Specifically, we designed a honeycomb pattern consisting of a low-damping permalloy hexagonal base plate with a high-damping embedded triangle plate. A STNO is placed at each narrow neck of this pattern, and the spin wave generated by the STNO could propagate smoothly within the hexagonal base plate while being absorbed by the triangle plate. The narrow neck's width was comparable with the spin wave's wavelength, effectively limiting the propagation of spin waves to only the nearest neighbors. Through our simulations, we observed global synchronization across the entire array. Our design created a network of interactions among the oscillators that only involved nearest neighbors, allowing for the emergence of collective behavior. This approach overcomes the challenges posed by the inherent non-linearity and complicated interactions and provides a new pathway towards achieving synchronization in large arrays of STNOs.

4.1 Problem formulation

We consider synchronization in arrays of N STNOs formed in a $d = 3\text{nm}$ thick soft magnetic film with the saturation magnetization $M_s = 6.37\times10^5\,\text{A/m}$ and exchange constant $A_{ex} = 1.4\times10^{-11}\,\text{J/m}$ with no anisotropy (Fig. 4.1). The magnetization is initialized along the +z direction and is relaxed to its equilibrium state under an external perpendicular applied field $\mathbf{H}_{app}$. STNOs are formed by STT set via a point contact of $r_{STNO} = 50\,\text{nm}$ radius with the current of $\mathbf{I}_{stt}$ with -z polarization direction with the polarization efficiency of $p = 0.3$.

The vertical applied field sets the STNO frequency and leads to magnetostatic forward volume wave (MSFVW) spin waves [16] that are isotropic with respect to the in-plane propagation and whose lowest mode wavelength is given by the following dispersion relation[ref]:

$$\omega_{\text{MSFVW}}(k_\perp)^2 = \omega_{ex}[\omega_{ex} + \omega_M(1 - \frac{1-e^{-k_\perp d}}{kd})] \tag{4.1}$$

Where $\omega_M = \gamma\mu_0 M_s$ and $\omega_{ex} = \gamma\mu_0\left(\left|\mathbf{H}_{\text{app}}\right| - M_s\right) + 2\gamma A_{ex}k_\perp^2 / M_s$, $k_\perp$ is the in-plane wave vector of MSFVW and d is the thickness of the film, in our case $d = 3\text{nm}$. One thing to notice is that for the thin film, only the lowest mode without perpendicular component can exist [ref], namely the z -direction wave vector $k_z \equiv 0$. The STNOs are driven by STT currents that ensures their operation with a large-angle precession, leading to equal-strength interactions among all pairs of STNOs.

The synchronization is manifested via frequency and phase locking on the STNO auto-oscillations, i.e., a stable phase difference between any two oscillators. To quantify the synchronization, let $\mathbf{m}_i(\theta_i, \phi_i, t)$ be the spatial averaged magnetization of the i^{th} STNO, θ_i be the corresponding azimuthal angle, and ϕ_i be the elevation angle. The phase difference can then be expressed as $\Delta_{i,j}(t) = \cos^{-1}[\mathbf{m}_i(\theta_i, \phi_i, t) \cdot \mathbf{m}_j(\theta_j, \phi_j, t)]$. If $\Delta_{i,j}(t)$ is independent of time, then the i^{th} and j^{th} STNOs are phase locked. If this holds for all pairs of STNOs, then global synchronization is achieved. The case of all $\Delta_{i,j} = 0$ corresponds to a global in-phase synchronization. A measure of synchronization is the output power averaged over a time period T_0 :

$$P = \frac{1}{T_0} \int_0^{T_0} \left(\sum_{i=1}^{N} m_{ix} \right)^2 dt \, . \tag{4.2}$$

Such an output power corresponds to the total current generated by the array if an in-plane layer is added to each STNO to generate an electric signal via magnetoresistance. For a non-synchronized array in which a single STNO produced power $P_0 = \int_0^{T_0} (m_x)^2 dt / T_0$, the total power is $P_{ns} = NP_0$ because of the arbitrary phase differences, whereas for a synchronized array, $P_s = N^2 P_0$ because of the phase-locking. Based on the total output power scaling we can define the synchronization efficiency as $\eta = P/P_s \le 1$.

4.2 Device structure design

Synchronization between a pair of STNOs by means of spin waves can be readily achieved. However, synchronizing a large 2D array of STNOs by means of spin waves is challenging. As a larger array, we first consider synchronization in an array of $N = 3$ STNOs arranged at vertices of a triangle (Fig. 4.1(a), (d)). The applied field and current are chosen as $\mathbf{H}_{app} = 1.32T\hat{\mathbf{z}}$ and $\mathbf{I}_{stt} = -22.5mA\hat{\mathbf{z}}$, respectively. Via Eq. (4.1), these parameters correspond to the STNO oscillating frequency $\omega / 2\pi \approx 30\text{GHz}$ with spin wave wavelength of $\lambda \approx 60\,\text{nm}$, and the elevation angle of the STNO precession is $\theta_e \approx 30^\circ$. The separation between the STNOs is chosen as $d_{ee} = 420\,\text{nm}=7\lambda$ such that it corresponds to an integer number of spin wave wavelengths, so that in-phase synchronization is obtained. Global synchronization is achieved in this case, similar to the case of two STNOs (Fig. 4.1(d)). We then extend the structure to a hexagonal array of $N = 12$ STNOs (Fig. 4.1 (b), (e)). In this case, a chaotic behavior is obtained with no synchronization (Fig. 4.1(e)). Trying to move the STNO locations or making arranging them in a different, e.g., square, array still leads to a similar chaotic behavior. This lack of synchronization is attributed to the fact that the spin waves generated by each STNO interact with other STNOs at different phases, thus leading to hectic, out of phase interaction patterns. The interactions carry varying phase differences originating from neighbor STNOs at different distances. Furthermore, the strength of these interactions also varies with the distance, contributing to non-uniform effects across the array. These factors result in accumulated discrepancies that significantly disrupt global synchronization. The interplay of inconsistent phase shifts and variable interaction strengths results in a complex dynamic, where the collective behavior deviates from the expected synchronized order, leading to the chaotic patterns observed in larger arrays.

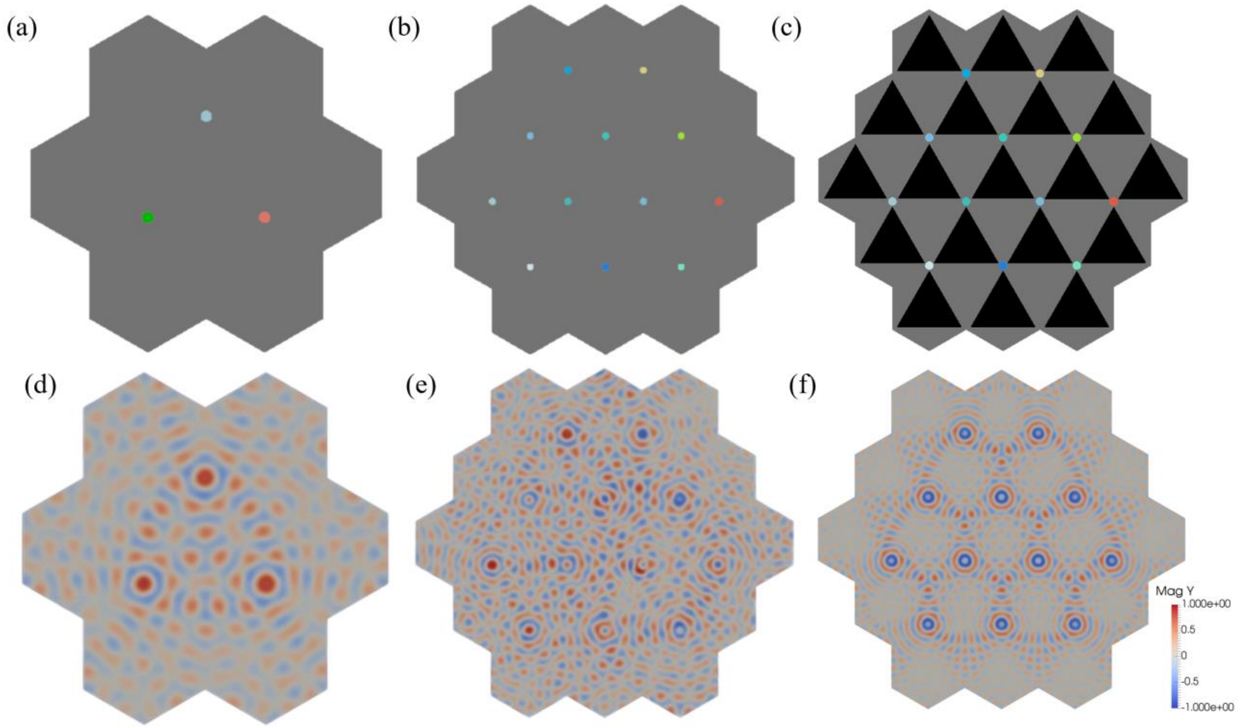


Figure 4.1: Illustration of the model without high damping region for (a) $N = 3$ (b) $N = 12$ and with high damping region (c) $N = 12$. Grey substrate here is of low damping constant, black congruent triangles are of high damping constant, and the small dots spin torque oscillators. The corresponding *y*-magnetization are shown below.

In exploring the synchronization pattern further with various geometry configurations, we find that global synchronization is determined by the presence of next-nearest-neighbor and further interactions and possibly unequal separations between STNOs. An approach to limit the chaotic behavior caused by the long-range interactions is to block/absorb the propagation of spin wave. This understanding leads to introducing the honeycomb structure in Fig. 4.1(c). In this structure, the STNOs are positioned in a hexagonal pattern in a magnetic film. The magnetic properties of the film, except the damping constant, are homogeneous across the film. The damping constant on the other hand has triangular regions of high values (black color in Fig. 4.1(c)) and low values (grey color in Fig. 4.1(c)). The low- and high-damping values can be either determined by ion irradiation or by defining additional nano-contacts with polarized current, which would either increase or decrease the effective damping constant as set by the strength and direction of the STT current. The large and low damping regions have short- and long-range spin wave propagation, respectively. The size of the neck (formed by the high damping regions) d_{neck} is comparable to the spin wave wavelength λ, effectively preventing a strong spin wave transmission through the neck, such that the spin wave excitation in the neck is determined by the corresponding STNO. Additionally, the hexagonal array ensures that the nearest-neighbor distances are the same, which leads to the array symmetry with equal spin phase propagation phases enabling global synchronization. In this particular structure, $d_{neck} = 80\,\text{nm} \approx 1.33\lambda$, the higher damping constant is $\alpha = 0.2$ and lower damping constant is $\alpha = 0.01$. With this high-/low-damping structure, we obtain a global synchronization for the same $N = 12$ STNOs, as is evident via the high synchronization efficiency in Fig. 4.1(f).

We, then, consider larger STNO arrays, up to 48 STNOs, and perform simulations at zero and (300 K) room temperatures with all STNOs driven by the same $\mathbf{H}_{app}$ and $\mathbf{I}_{stt}$ (Fig. 4.2(a)). Global synchronization is again achieved with a high synchronization efficiency η. The linewidth is below 0.2 GHz even for the room temperature cases, which is much smaller the linewidth of a single STNOs, where the Full width at half maximum (FWHM) is $\approx 1.8\text{GHz}$, which is another indication of global synchronization. Fig. 4.2(b) further shows the spin wave and STNOs phase patterns with $N = 48$ under 300K when there is global synchronization. In the figure we observed a phase shift across the array due to thermal fluctuation and the minor mismatch between d_{ee} and λ, resulting in slight drop of η compared with array with smaller sizes and zero temperature simulation. The results demonstrate the same spin wave interference patterns in all low damping regions and strong absorption at the edges of each high damping region. These consistent spin wave patterns provide evidence that the honeycomb design effectively limits long-range spin wave propagation and achieves nearest-neighbor-only interactions among the STNOs.

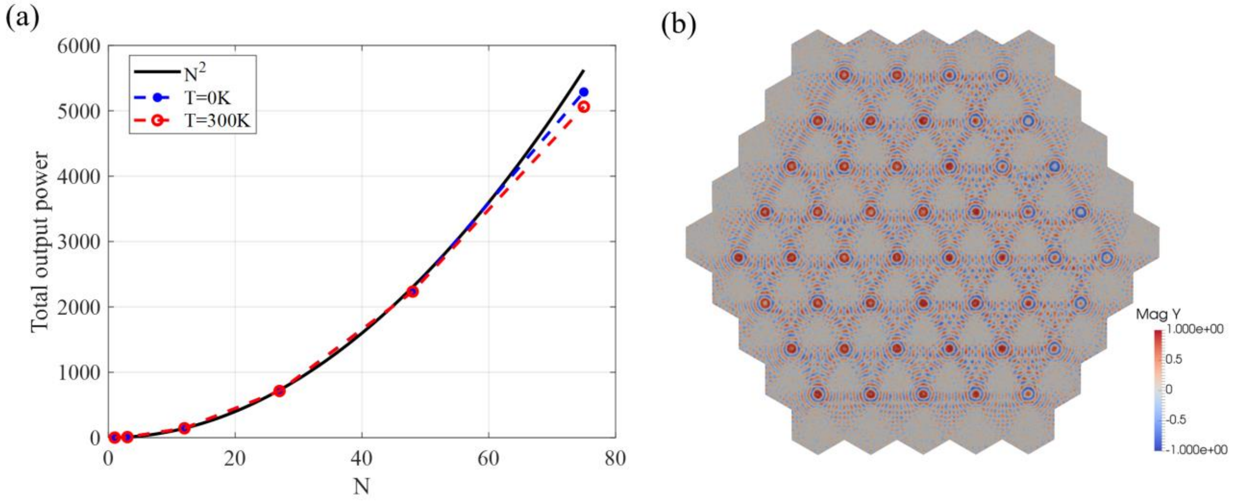


Figure 4.2: (a) The simulated total output power under 0K (blue dashed line) and 300K (red dashed line) vs theoretical maximum (black solid line), at both temperature the result indicates global in-phase synchronization is achieved. (b) Snapshot of y-magnetization under 300K, 48 STNOs in total. Phase shift can be observed via visual inspect.

To find such best combination of $\mathbf{H}_{app}$ and $\mathbf{I}_{stt}$, we first choose a set of $\mathbf{I}_{stt}$ from 22.0 mA to 23.0 mA with 0.1 mA stride. Then for each fixed $\mathbf{I}_{stt}$, we sweep the $\mathbf{H}_{app}$ from $1T\hat{z}$ to $2T\hat{z}$ with 1000 ns duration. We calculate the synchronization efficiency η by integrating all the $m_{ix}, i \in [1, N]$ with a time period $T_0 = 1\text{ns}$, and find under what $\mathbf{H}_{app}$ and $\mathbf{I}_{stt}$ the best synchronization efficiency η can appear. Our sampling frequency is $f_{samp} = 100GHz$, namely the maximum frequency we can detect is $f_{\max} = f_{samp} / 2 = 50GHz$ according to Nyquist theorem [ref], which is sufficient in this case where STNOs are working at around 30 GHz. The number of STNO $N = 3, 12, 27, 48$, and the resulting total output power averaged over the last 10 ns in each simulation is shown in Fig. 4.2(a).

We note synchronization is achieved only in a certain range of the parameters. To characterize such a range, Fig. 4.3 shows the synchronization efficiency different STNO array sizes as a function of $\mathbf{H}_{app}$, keeping the rest of the paramters the same. We observe a sharp transition from a low and irregular efficiency, indicating a lack of synchronization, to a stable high efficiency for a proper range of $\mathbf{H}_{app}$. Smaller N has a wider synchronization range whereas for larger N, only a narrow range of $\mathbf{H}_{app}$ and $\mathbf{I}_{stt}$ leads to synchronization. The lack of synchronization for smaller $\mathbf{H}_{app}$ is a higher spin wave magnitude, so that the spin waves can break through the neck barriers, and they thus reach the next-nearest neighbors, leading to the aforementioned chaotic behavior. With a stronger $\mathbf{H}_{app}$, the spin wave magnitude decreases preventing the next-nearest-neighbor interactions, resulting in synchronization. Too large $\mathbf{H}_{app}$ also may destroy synchronization, which is attributed to the weak spin wave excitation, resulting in too weak coupling between the STNOs. Other parameters, e.g., $\mathbf{I}_{stt}$, affect the synchronization range in a similar way, so that various ranges of parameters can be tuned. The synchronization range for very large arrays is narrower because the phase differences between the STNOs accumulates eventually resulting in destroying the global synchronization.

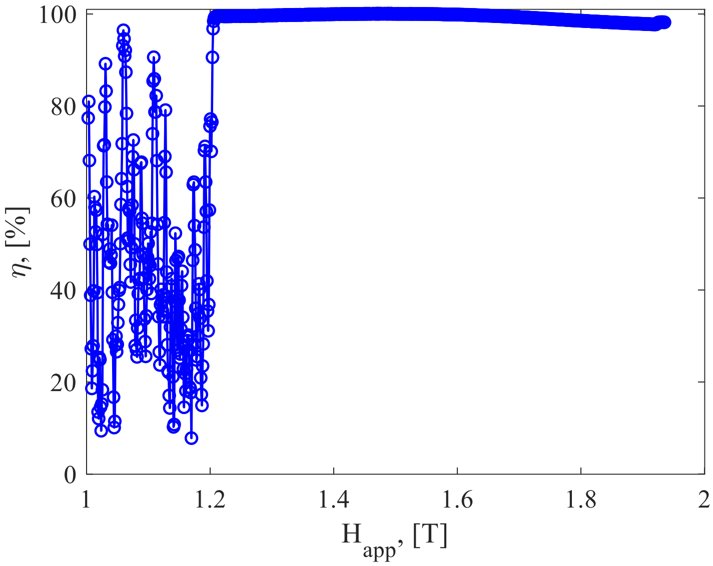


Figure 4.3: Calculated synchronization efficiency η of the whole array under different applied field $\mathbf{H}_{app}$. A sharp transition from low output power to nearly 100% output power indicates the emergence of global in-phase synchronization. And the synchronization efficiency η is maintained well above 90% after this transition.

The synchronization range also is reduced at elevated temperatures due to stochastic (Brownian-like) effects as well as random distributions of the material properties. To characterize the effects of the parameter distributions, Fig. 4.4(a) shows the synchronization efficiency for an STNO array of $N = 48$ with the saturation magnetization of uniform distribution with the mean value of $\mu = 6.37\times10^{5}$ A/m and deviations of $\sigma = 0,1,5,10\,\%$, i.e. $M_s = \mu \pm \sigma / 2$. We find that η slightly drops from around 98% to 96.5%, which is still well above 90%. Fig. 4.4(b) exhibits a snapshot of comparison between the averaged m_y over all 48 STNOs as $\overline{m_y}$ and the m_y of the 1st STNO at the top left corner for the largest σ we test. The $\overline{m_y}$ overlaps with m_y tightly and has no significant difference in between, showing a robust global synchronization is still well maintained with distributed material parameter. The minor decline in synchronization efficiency η is mostly attributed to the resulting distribution in the STNO frequencies, which leads to extra phase shifts between the STNOs. However, the synchronization is still achieved for reasonable ranges of the parameters.

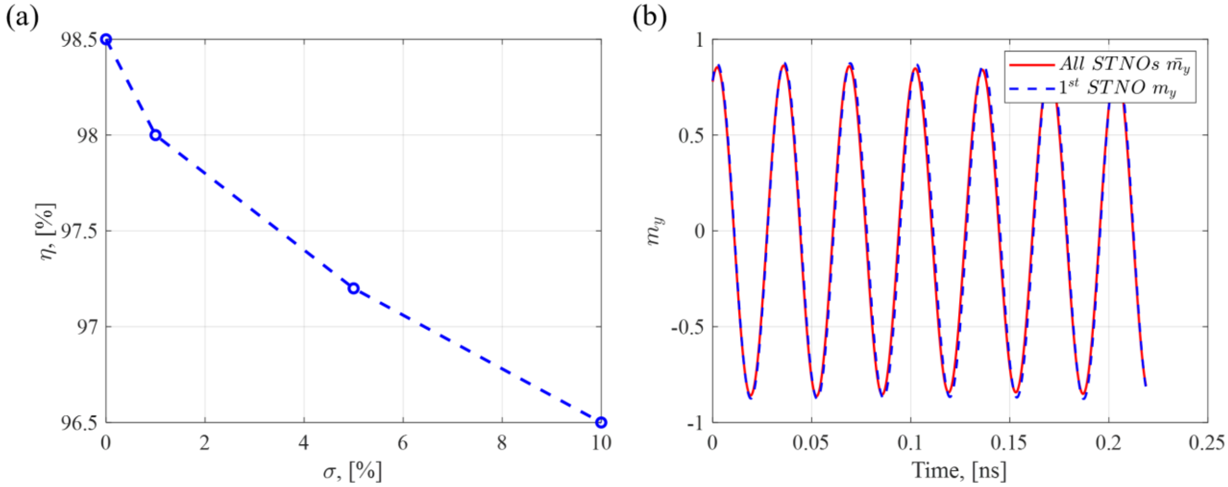


Figure 4.4: Synchronization with mean value $\mu = 6.37 \times 10^5$ A/m and deviation $\sigma = 0,1,5,10\%$ for saturation magnetization $M_s = \mu \pm \sigma / 2$ and $N = 48$. (a) The synchronization efficiency η slightly drops from 98.5% to 96.5. (b) A snapshot of average of m_y over all 48 STNOs vs m_y from two well-separated corners with $\sigma = 10\%$.

The ideas leading to global synchronization also allow creating synchronization in subarrays of the entire large array, providing flexibility for various application requirements. For example, by turning on and off some STNOs at certain position by locally reducing or extending the spin wave propagation length, we can create a synchronized STNO ring by only activating the outermost shell of the array (Fig. 4.5(a)), or concentric loops by activating more inner shells (Fig. 4.5(b)).

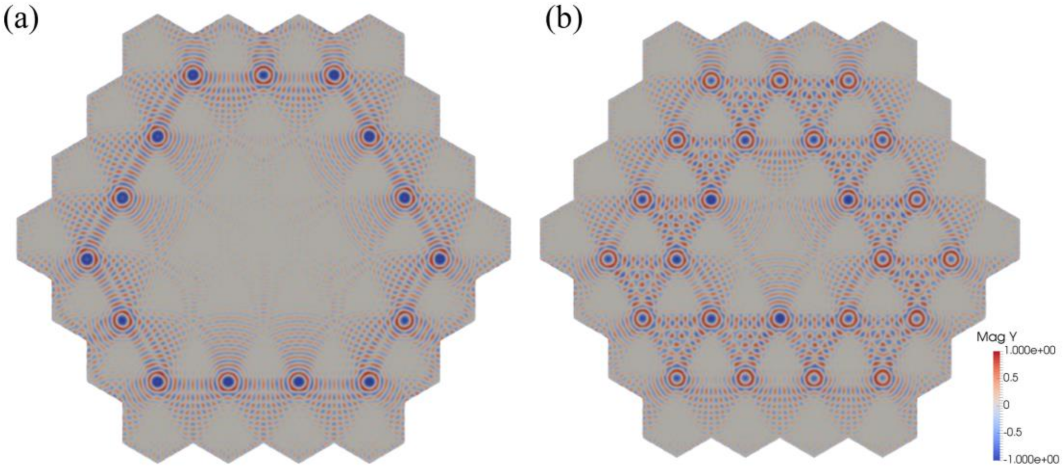


Figure 4.5: Magnetization along y-direction of synchronized (a) ring and (b) concentric loops. These two structures are built by turning off some STNOs on the model of size $N = 27$.

4.3 Summary

In conclusion, synchronization of large arrays of STNOs by spin waves is possible when a mechanism restricting long range interactions between the STNOs is chosen together with properly tuned parameters. The mechanism chosen here was based on controlling the magnetic damping, which controls the spin wave propagation length. Adaptive control of the spin wave propagation length also allows defining synchronization in selected subarrays of the entire array with tuned paths of synchronization. Additional mechanisms may also be possible, e.g., a network of nanowires connecting the STNOs may result in a similar behavior, albeit without an easy potential of adaptive tuning offered by tuning the spin wave propagation length. Potentially, additional tunning mechanisms may also be introduced, e.g., if the current in each STNO is tuned separately. The presented ideas may find applications in creating microwave generators with increased output power as well as in neuromorphic computing and signal processing. In particular, synchronized subarrays of large arrays of STNOs demonstrate diverse synchronization patterns, akin to the dynamics observed in a reservoir computing framework. These patterns can be leveraged to perform complex computational tasks, making them highly relevant for next-generation adaptive systems that require real-time processing capabilities.

Acknowledgements

Chapter 4, in full, is a reprint of the material as it appears in F. Ai and V. Lomakin, “Spin-wave assisted synchronization in 2D arrays of spin torque oscillators”, which is currently under review in Phys. Rev. Appl. The dissertation author was the primary researcher and author of this paper.

Chapter 5 Fast periodic interpolation method for superposition sums in a periodic unit cell

A common category of problem in computational physics is calculating the potential, referred to as periodic scalar potential (PSP), that is generates by a non-uniform distribution of sources arranged in a unit cell that is extended to an infinite periodic array (Fig. 5.1). Examples of such problems are phased-array antennas, crystals, periodic gratings, periodically modulated waveguides, to name a few [50–53]. These problems may present challenges for obtaining accurate and rapid solutions. For a non-periodic unit cell, this task can be accomplished via a standard superposition sum involving free-space Green's function as its kernel. Such sums can be evaluated rapidly in $O(N)$ or $O(N\log N)$ computational cost for N sources and observers using several fast methods, such as fast multipole like method [54,55], H-matrix method [56] , interpolation-based methods [57–59], and FFT-based methods [5,60–62]. However, for the periodic case, the superposition sums involve infinite number of images, which poses additional challenges to the slow converge or even divergence as well as a potentially high computational cost.

Several existing methods can address such periodic problems with certain assumptions and limitations [63–66] . Periodic Green's function (PGF) can be defined in terms of an infinite sum over periodic images. In dynamic problems, e.g., in computational electromagnetics, PGF is assumed to be defined with a phase shift or non-vanishing wavenumber [67–69]. However, for static problems with no phase shift, even in the 1D case, the PGF diverges. In molecular dynamics problems [70], a special form of static PGF assumes uniformly arranged sources [71], such as a cubic lattice. Although various methods have been proposed to address static and dynamic periodic unit cell problems with non-uniform sources, they often focus on specific problem types, such as addressing only 1D or 2D periodicity, addressing only static problems [72] , requiring lattice structure or relying on finite difference methods [73]. There is a need in a general approach capable of effectively handling all types of these problems regardless of the periodicity type, distribution of sources, with or without phase shift, and for static and dynamic cases.

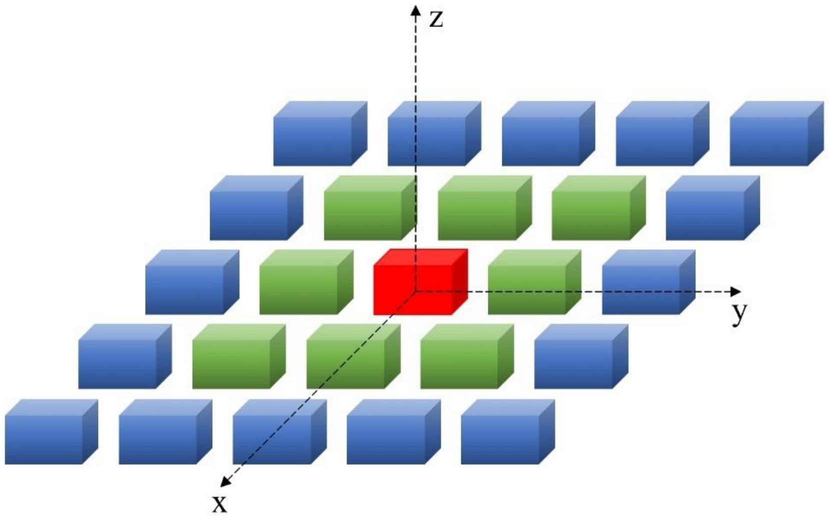


Figure 5.1: Illustration for a 2D periodic problem consisting of an infinite 2D array of cubes. The central red cube is the zeroth unit cell. Surrounding it along the x and y axes are the 1^{st} order image cubes in green and the 2^{nd} order image cubes in blue.

In this chapter, we proposed a general approach, referred to as Fast Fourier Transform Periodic Interpolation Method (FFT-PIM) that is applicable to all these types of problems under a unified framework, including 1D, 2D, and 3D periodicities for static and dynamic problems with and without periodic phase shifts. The FFT-PIM separates the PSP into a near-zone component involving a small number of images near the periodic unit cell of interest, and a spatially slowly varying far-zone component involving the rest of an infinite number of images. The evaluation of the near-zone PSP component is based on the box adaptive integral method [5], which is modified to allow for the rapid computations with multiple near-zone images. The evaluation of the far-zone PSP component is based on the sparse periodic interpolation method [69], which is adapted to allow handling dynamic and static problems with arbitrary phase shifts. The computational cost of the FFT-PIM is of $O(N \log N)$ and the memory consumption is of $O(N)$.

The chapter is organized as follows. Sec. 5.1 presents the problem formulation. Sec. 5.2 presents the algorithmic foundations of the FFT-PIM, including near- and far-zone representations of PGF and PSP as well as efficient ways of their evaluation. Sec. 5.3 shows results for the use of the FFT-PIM, its error analysis, and computational performance. Finally, Sec. 5.4 presents summary and conclusions.

5.1 Problem formulation

Consider an infinite periodic array of unit cells residing in free space (see Fig. 5.1 for a 2D periodicity example). Within each unit cell there are N source points located at $\mathbf{r}_n$. In the zeroth unit cell, the source amplitudes are q_n, where $q=1...N$. In each unit cell, there is the same number of coinciding observer points at which the periodic scalar potentials (PSPs) $u(\mathbf{r}_i)$ are to be found. The periodic array can be 1D, 2D or 3D with the periodicity of L_x, L_y, and L_z in three possible directions x, y and z, respectively. The amplitudes of the sources may be periodically phase shifted with a linear, possibly complex, phase shift determined by the wavenumbers k_{x0}, k_{y0} and k_{z0} along the x, y and z directions. A free space wavenumber k_0 describes the propagation of the waves in the free space for the dynamic case. For $k_0 = 0$, the problem is static, and for $k_{x0} = k_{y0} = k_{z0} = 0$, the sources are not phase shifted.

The PSP in the zeroth unit cell can be written as

$$u(\mathbf{r}_m) = G^p(\mathbf{r}_m - \mathbf{r}_n)q(\mathbf{r}_n) \tag{5.1}$$

Here the $m=1...N$ and the G^p represents the scalar periodic Green's function (PGF), given for the 1D, 2D, and 3D cases as:

$$G_{1D}^p(\mathbf{r}) = \sum_{i_x=-\infty}^{\infty} e^{-jk_{x0}(i_x L_x)} G_0(\mathbf{r} - i_x L_x \hat{\mathbf{x}}) \tag{5.2}$$

$$G_{2D}^{p}(\mathbf{r})=\sum_{i_x=-\infty}^{\infty}\sum_{i_y=-\infty}^{\infty}e^{-j[k_{x0}(i_xL_x)+k_{y0}(i_yL_y)]}G_0(\mathbf{r}-i_xL_x\hat{\mathbf{x}}-i_yL_y\hat{\mathbf{y}}) \quad (5.3)$$

$$G_{3D}^{p}(\mathbf{r})=\sum_{i_x=-\infty}^{\infty}\sum_{i_y=-\infty}^{\infty}\sum_{i_z=-\infty}^{\infty}e^{-j[k_{x0}(i_xL_x)+k_{y0}(i_yL_y)+k_{z0}(i_zL_z)]}G_0(\mathbf{r}-i_xL_x\hat{\mathbf{x}}-i_yL_y\hat{\mathbf{y}}-i_zL_z\hat{\mathbf{z}}) \quad (5.4)$$

respectively. Here, G_0 is the free space scalar Green's function given by $G_0(\mathbf{r})=\exp(-jk_0\,|\,\mathbf{r}\,|)/4\pi\,|\,\mathbf{r}\,|$.

The series in Eq. (5.1) for the PSP is important in multiple computational physics problems. For example, the evaluation of the superposition integrals appearing in electromagnetic integral equation or micromagnetic solvers [74] can be represented as a product of sparse matrices describing the mesh of the geometry and the result of the superposition sum of Eq. (5.1). The sum of Eq. (5.1) can also be directly apply to a discrete set of sources and observers, e.g., when considering a set of point sources.

The series of Eq. (5.2), Eq. (5.3) and Eq. (5.4) for the PGF is slowly convergent or even divergent depending on the wavenumbers, which is because of a slow spatial decay of G_0. An alternative spectral representation of the PGF of Eq. (5.2), Eq. (5.3) and Eq. (5.4) can be based on the Floquet mode expansion [69]:

$$G_{1D}^{p}(\mathbf{r})=\sum_{m=-\infty}^{\infty}\frac{1}{4jL_x}e^{-jk_{xm}x}H_0^{(2)}(k_{\rho m}\sqrt{y^2+z^2}) \quad (5.5)$$

$$G_{2D}^{p}(\mathbf{r})=\sum_{m,n=-\infty}^{\infty}\frac{e^{-jk_{xm}x-jk_{yn}y-jk_{zmn}|z|}}{2jk_{zmn}L_xL_y} \quad (5.6)$$

$$G_{3D}^{p}(\mathbf{r})=\sum_{m,n=-\infty}^{\infty}\frac{e^{-jk_{xm}x-jk_{yn}y}}{2jk_{zmn}L_xL_y}\times\left(e^{-jk_{zmn}|z|}+\frac{e^{-j(k_{zmn}-k_{z0})L_z}e^{-jk_{zmn}z}}{1-e^{-j(k_{zmn}-k_{z0})L_z}}+\frac{e^{-j(k_{zmn}+k_{z0})L_z}e^{jk_{zmn}z}}{1-e^{-j(k_{zmn}+k_{z0})L_z}}\right) \quad (5.7)$$

where $H_0^{(2)}$ is the zeroth order Hankel function of the second kind, $k_{xm} = k_{x0} + 2\pi m / L_x$, $k_{ym} = k_{y0} + 2\pi n / L_z$, $k_{\rho m} = (k_0^2 - k_{xm}^2)^{1/2}$ and $k_{zmn} = (k_0^2 - k_{xm}^2 - k_{yn}^2)^{1/2}$, and the square roots are chosen such that their imaginary parts are non-positive. The spectral series of Eq. (5.5), Eq. (5.6) and Eq. (5.7) converge exponentially fast when $(y^2 + z^2)^{1/2}$ is non-vanishing and $k_0, k_{x0}, k_{y0}, k_{z0}$ are not all vanishing. The series may diverge or not be defined for certain parameter combinations with $k_{zmn} = 0$, $k_{\rho m} = 0$, which are related to so called Rayleigh-Wood anomalies [75–78] . For scattering electromagnetic or acoustic problems, such anomalies correspond to the transition between evanescent and propagating Floquet modes. Additional types of Wood anomalies may be of resonant type [78], e.g., when Green's function is defines in the presence of a layered medium [69]. When such anomalies occur, they may require a special treatment, e.g., Ref. [75] or by defining the phase shift in the complex plane with the integration path deformation [77,79].

A special attention needs to be paid to the no phase static periodic (NPSP) case where $k_0 = k_{x0} = k_{y0} = k_{z0} = 0$. This case is important as it corresponds to several practical problem, such as using periodic extensions to represent infinite 1D, 2D, and 3D domains in various static problems, e.g., to characterize micromagnetic [80] or in molecular dynamics problems. In this NPSP case, the PGF calculated via series of Eq. (5.2), Eq. (5.3) and Eq. (5.4) or Eq. (5.5), Eq. (5.6) and Eq. (5.7) diverges and the PSP for a general source distribution diverges as well. However, a finite PSP can be obtained for the special case of a neutral source, i.e., under the condition Eq. (2.7) . This neutrality condition corresponds to various physical problems, e.g., when computing the magnetic scalar potential generated by a magnetized objects or electrostatic potential in polarized objects [81]. In this case, the PSP converges and we can neglect any non-relevant components (e.g., constant terms) in the PGF, so that it is convergent as well. The result is that the PGF for the neutral NPSP case can be calculated via:

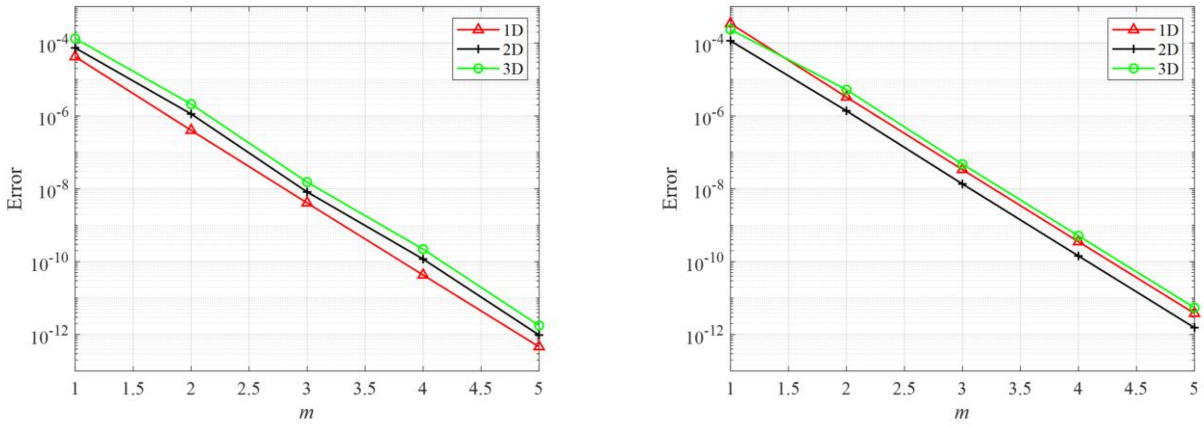
$$G_{1D}^{p}(\mathbf{r}) = -\frac{1}{2\pi L_x}\ln\left(\sqrt{y^2+z^2}\right) + \frac{1}{\pi L_x}\sum_{m=1}^{\infty} K_0\left(\frac{2m\pi}{L_x}\sqrt{y^2+z^2}\right)\cos\left(\frac{2m\pi x}{L_x}\right) \tag{5.8}$$

$$\begin{aligned} G_{2D}^{p}(\mathbf{r}) &= -\frac{|z|}{2L_xL_y} - \frac{1}{4\pi L_x}\ln\left(1-2e^{-\frac{2\pi|z|}{L_y}}\cos\left(\frac{2\pi y}{L_y}\right)+e^{-\frac{4\pi|z|}{L_y}}\right) \\ &+\frac{1}{\pi L_x}\sum_{m=1}^{\infty}\sum_{n=-\infty}^{\infty} K_0\left(\frac{2m\pi}{L_x}\sqrt{(nL_y+y)^2+z^2}\right)\cos\left(\frac{2m\pi x}{L_x}\right) \end{aligned} \tag{5.9}$$

$$\begin{aligned} G_{3D}^{p}(\mathbf{r}) &= \frac{1}{2L_xL_yL_z}(z^2 - |z|L_z + \frac{L_z^2}{6}) \\ &-\frac{1}{4\pi L_x}\sum_{k=-\infty}^{\infty}\ln\left(1-2e^{-\frac{2\pi|kL_z+z|}{L_y}}\cos\left(\frac{2\pi y}{L_y}\right)+e^{-\frac{4\pi|kL_z+z|}{L_y}}\right) \\ &+\frac{1}{\pi L_x}\sum_{m=1}^{\infty}\sum_{n,k=-\infty}^{\infty} K_0\left(\frac{2m\pi}{L_x}\sqrt{(nL_y+y)^2+(kL_z+z)^2}\right)\cos\left(\frac{2m\pi x}{L_x}\right) \end{aligned} \tag{5.10}$$

where K_0 is the zeroth order modified Bessel function of the second kind.

It can be shown that Eq. (5.8) is obtained from Eq. (5.5) under the assumption of Eq. (2.7). In this case, by taking the limit of ($k_0 \to 0, k_{x0} \to 0$) for all $n \neq 0$, the Hankel function becomes the modified Bessel function and the symmetric $\pm n$ terms can be combined into the positive n terms. For the $n = 0$ term, the Hankel function exhibits a logarithmic behavior but it is summed up to 0 due to the source neutrality of Eq. (2.7). The PGF for the 2D and 3D cases can be then obtained by a spatial sum of 1D PGFs. When the sources are arranged on a lattice, the PGF for the NPSP case sometimes is also referred to as the lattice Green's function [71].



(a)

(b)

Figure 5.2: Convergence of PGFs, $L_x, L_y, L_z = 1$, $x = y = z = 0.5$. (a) The relative error of the sum of the first m terms from Eq. (5.5), Eq. (5.6) and Eq. (5.7), with $k_0 = -1 - j$ and $k_{x0} = 1 - j, k_{y0} = 1 + j, k_{z0} = -1 + j$, (b) The relative error of the sum of the first m terms from Eq. (5.8), Eq. (5.9) and Eq. (5.10). Exponential convergence is achieved in both cases.

The PGFs converge exponentially fast provided the $(y^2 + z^2)^{1/2}$ is not too small. For a desired error of ϵ, the number of terms required Eq. (5.5) and Eq. (5.8) scales as $L_x \log(\epsilon^{-1}) / z$ for the 1D case and $L_x L_y \log^2(\epsilon^{-1}) / z^2$ for the 2D and 3D cases. To demonstrate the convergence, Fig. 5.2 shows the PGFs and their convergence when using Eq. (5.5), Eq. (5.6) and Eq. (5.7) or Eq. (5.8), Eq. (5.9) and Eq. (5.10) for $L\_x, L\_y, L\_z = 1$, $k\_0 = -1 - j$ and $k_{x0} = 1 - j,\ k_{y0} = 1 + j,\ k_{z0} = -1 + j$ in Fig. 5.2 (a) and the NPSP case in Fig. 5.2 (b).

5.2 FFT-PIM algorithm

In this section we describe the FFT-PIM algorithm that allows rapidly computation of the PSP in Eq. (5.1).

5.2.1 Representation of PGF and PSP

We start by representing the PGF in terms of its near-zone component G^p_{near} and far-zone component G^p_{far} :

$$G^p = G^p_{near} + G^p_{far} \tag{5.11}$$

Here, the near-zone component G^p_{near} is given by Eq. (5.2), Eq. (5.3) and Eq. (5.4) but with the sums over i_x, i_y, and i_z in the range of $\left[-i_{xd}, i_{xd}\right]$, $\left[-i_{yd}, i_{yd}\right]$, and $\left[-i_{zd}, i_{zd}\right]$, respectively, where i_{xd}, i_{yd}, i_{zd} are small integer numbers. These components represent the contribution from a small number of images around the zeroth unit cell of interest. Assuming that the periodic lengths are close to each other, we can use the same value of i_d for i_{xd}, i_{yd}, i_{zd}. For the sake of clarity, we assume this case in the following discussions. G^p_{near} includes the self-term and a few terms from the surrounding images (see Fig. 5.1 showing the near-zone images for the case of $i_{xd} = i_{yd} = 1$ as green boxes). Therefore, G^p_{near} may be spatially singular and rapidly varying. The evaluation of G^p_{near} involves only a small number of the sum terms.

The far-zone component G^p_{far} is the rest of the images from far-away unit cells and can be evaluated as $G^p_{far} = G^p - G^p_{near}$ (blue boxes in Fig. 5.1). The far-zone PGF G^p_{far} is smoothly varying in space because it corresponds to large source-observer spatial separations. The rate of the variations is reduced by using a greater i_d.

Following the near- and far-zone decomposition of the PGF, the potential is also expressed in terms of its near- and far-zone components:

$$u(\mathbf{r}_m) = u_{near}(\mathbf{r}_m) + u_{far}(\mathbf{r}_m) \tag{5.12}$$

$$u_{near}(\mathbf{r}_m) = \sum_{n=1, n\neq m}^{N} G^p_{near}(\mathbf{r}_m - \mathbf{r}_n) q_n \tag{5.13}$$

$$u_{far}(\mathbf{r}_m) = \sum_{n=1, n\neq m}^{N} G^p_{far}(\mathbf{r}_m - \mathbf{r}_n) q_n \tag{5.14}$$

where, again, the far-zone potential u_{far} is a spatially slowly varying function. These potential and PGF decompositions lead to a fast approach for the numerical evaluation of the PSP.

5.2.2 Evaluation of the far-zone PSP

The far-zone PGF and PSP components vary slowly in space, which enables us to calculate the far-zone PGF and PSP on a uniform grid within the observation domain and subsequently interpolate to the required observer points. For a target object of size D_x, D_y, D_z, we construct a sparse uniform grid of observation points $\{\mathbf{r}^o_{n(l,p,q)} = x^o_l, y^o_p, z^o_q\}$ (red grid in Fig. 5.3) and a spatially shifted uniform grid of source points $\{\mathbf{r}^s_{n(l,p,q)} = x^s_l, y^s_p, z^s_q\}$ (black grid in Fig. 5.3):

$$x^s_{n(l,p,q)} = \frac{(l-1)(D_x - \Delta_x)}{N_{gx}-1}, y^s_{n(l,p,q)} = \frac{(p-1)(D_y - \Delta_y)}{N_{gy}-1}, z^s_{n(l,p,q)} = \frac{(q-1)(D_z - \Delta_z)}{N_{gz}-1} \quad (5.15)$$

$$x^o_{n(l,p,q)} = x^s_{n(l,p,q)} + \frac{\Delta_x}{2}, \quad y^o_{n(l,p,q)} = y^s_{n(l,p,q)} + \frac{\Delta_y}{2}, \quad z^o_{n(l,p,q)} = z^s_{n(l,p,q)} + \frac{\Delta_z}{2} \quad (5.16)$$

$$\Delta_x = \frac{D_x}{N_{gx}}, \Delta_y = \frac{D_y}{N_{gy}}, \Delta_z = \frac{D_z}{N_{gz}}; n = lN_{gz}N_{gy} + pN_{gz} + q \quad (5.17)$$

Here, N_{gx}, N_{gy}, N_{gz} are the number of the grid points in the x, y, z dimensions, $N_g = N_{gx}N_{gy}N_{gz}$ is the total number of the grid points, which is of $O(1)$, with $n \in [1, N_g]$, and $\Delta_x, \Delta_y, \Delta_z$ are grid spacing on in the x, y, z dimensions. The source grid is shifted from the observer grid by $\Delta_x / 2, \Delta_y / 2, \Delta_z / 2$ to make the grids non-overlapping, which results in a fast convergence of the sum in Eq. (5.5), Eq. (5.6) and Eq. (5.7) or Eq. (5.8), Eq. (5.9) and Eq. (5.10) when calculating PGF.

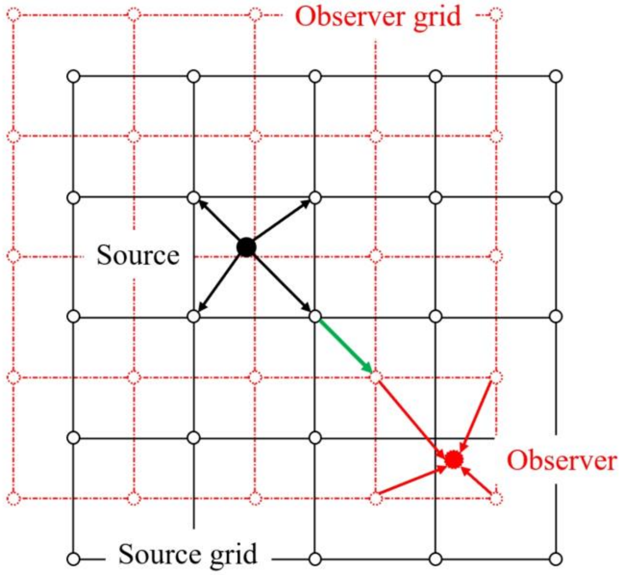


Figure 5.3: Illustration of the source (black) and observer (red) grids for the far-zone PSP component. Black and red circles are source/observer grid points on the Cartesian lattices. With on-grid black/red circles and random black/red dots. Black/red/green arrows are projections, interpolations and interactions among grid sources.

zeroth unit cell can be calculated by interpolation:

$$G^{p}_{far}(\mathbf{r}_m-\mathbf{r}_n)=\sum_{m'=1}^{N_{go}(\mathbf{r}_m)}\sum_{n'=1}^{N_{gs}(\mathbf{r}_n)}\omega^{o}(\mathbf{r}_m,\mathbf{r}^{o}_{m'})G^{p}_{far}(\mathbf{r}^{o}_{m'}-\mathbf{r}^{s}_{n'})\omega^{s}(\mathbf{r}^{s}_{n'},\mathbf{r}_n) \tag{5.18}$$

Here, $\omega^{s}(\mathbf{r}^{s}_{n'},\mathbf{r}_n)$ represents the interpolation coefficients from the source grid points $\mathbf{r}^{s}_{n'}$ to original non-uniform source points $\mathbf{r}_n$ and $\omega^{o}(\mathbf{r}_m,\mathbf{r}^{o}_{m'})$ represents the interpolation coefficients from the observer grid points $\mathbf{r}^{o}_{m'}$ to the original non-uniform observer points $\mathbf{r}_m$. The number of the source interpolation points $N_{gs}(\mathbf{r}_n)$ and observer interpolation points $N_{go}(\mathbf{r}_n)$ represents the number of the grid points that need to be used for the interpolation. For example, we can choose the q^{th} order Lagrange interpolation such that all grid points participate in the interpolation for any of the non-uniform points. For this choice, $N_q=(q+1)^3$, $N_{go}=N_{gs}=N_q$, and the coefficients ω^{o} and ω^{s} are the same for the same points. For a more general case of a grid with N_q greater than N_{go} and N_{gs}, the interpolation for each non-uniform point involve only a subset of the grid points, and the coefficients ω^o and ω^s can be represented as sparse matrices. The corresponding sparse matrices are generally transpose versions of each other.

We can substitute $G^p_{far}(\mathbf{r}_m - \mathbf{r}_n)$ from Eq. (5.18) into Eq. (5.1) and re-arrange the sums to obtain an alternative representation:

$$u(\mathbf{r}_m) = \sum_{m'=1}^{N_{go}(\mathbf{r}_m)} \omega^o(\mathbf{r}_m, \mathbf{r}^o_{m'}) \sum_{n'=1}^{N_g} G^p_{far}(\mathbf{r}^o_{m'} - \mathbf{r}^s_{n'}) \sum_{n=1}^{N_s(\mathbf{r}^s_{n'})} \omega^s(\mathbf{r}^s_{n'}, \mathbf{r}_n) q(\mathbf{r}_n) \tag{5.19}$$

Here, N_s is the number of the non-uniform source points that participate in the process of interpolations with a source grid point $\mathbf{r}^s_{n'}$. These grid points can be found from the interpolation procedure of Eq. (5.18). For example, when choosing the q^{th} order Lagrange interpolation with $N_q = (q+1)^3$, $N_s = N$. For a more general choice of a greater N_q, N_s is found from the sparse matrix representation of the interpolation coefficients ω^o and ω^s.

Based on this grid construction in Eq. (5.15), Eq. (5.16), Eq. (5.17) and PSP representation in Eq. (5.19), we can first pre-compute the coefficients ω^o and ω^s, e.g., as a sparse matrix, and pre-compute $G^p_{far}(\mathbf{r}^o_{m'} - \mathbf{r}^s_{n'})$ table at the grid points. The computation of ω^o and ω^s involves $O(N)$ operations since the interpolations are spatially local. The computation of $G^p_{far}(\mathbf{r}^o_{m'} - \mathbf{r}^s_{n'})$ involves $O(N_g)$ operations since the grid is uniform and G^p_{far} is shift invariant in term of the differences $\mathbf{r}^o_{m'} - \mathbf{r}^s_{n'}$. The computation of $G^p_{far}(\mathbf{r}^o_{m'} - \mathbf{r}^s_{n'})$ can be done via the rapidly convergent sums of Eq. (5.5), Eq. (5.6) and Eq. (5.7) or Eq. (5.8), Eq. (5.9) and Eq. (5.10) due to the choice of the source and observer grid in Eq. (5.15), Eq. (5.16) and Eq. (5.17) with a spatial separation.

The far-zone PSP u_{far} can, then, be evaluated via the following three steps, which are illustrated in Fig. 5.3.

Step 1, Projection from the source points to the source grid: This step calculates effective charges at the uniform source grid. Substituting the representation of Eq. (5.19) for G_{far}^{p} into Eq. (5.1) for the far-zone PGF and potential components allows obtaining an expression for the effective charge at the source grid points as projections from the original non-uniform sources (black arrows in Fig. 5.3 represent a contribution of a single non-uniform source to its relevant source grid points):

$$q_g(\mathbf{r}_{n'}^{s}) = \sum_{n=1}^{N_s(\mathbf{r}_{n'}^{s})} \omega^{s}(\mathbf{r}_{n'}^{s}, \mathbf{r}_n) q(\mathbf{r}_n), \tag{5.20}$$

where $n' \in [1, N_s]$.

Step 2, Evaluation of the PSP at the observer grid: This step calculates the far-zone PSP at the sparse observer grid points from the sparse source effective charges via the following convolution (green arrows in Fig. 5.3):

$$u_{far}^{g}(\mathbf{r}_{m'}^{o}) = \sum_{n'=1}^{N_g} G_{far}^{p}(\mathbf{r}_{m'}^{o} - \mathbf{r}_{n'}^{s}) q_g(\mathbf{r}_{n'}^{s}), \tag{5.21}$$

which can be evaluated rapidly directly since $N_g = O(1)$.

Step 3, Interpolation from the observer grid to the observer points: The far-zone component of the PSP at the original non-uniform points can be finally calculated by interpolation, which can be viewed as an inverse procedure of step 1 (red arrows in Fig. 5.3):

$$u_{far}(\mathbf{r}_m) = \sum_{m'=1}^{N_{go}} \omega^{o}(\mathbf{r}_m, \mathbf{r}_{m'}^{o}) u_{far}^{g}(\mathbf{r}_{m'}^{o}), \tag{5.22}$$

where $m \in [1, N]$.

5.2.3 Evaluation of the near-zone PSP

To evaluate the near-zone PSP component, u_{near}, we note that for $i_d = 0$, we only consider the zeroth order unit cell, which is equivalent to a non-periodic problem. The evaluation of u_{near} for a free-space non-periodic problem can be efficiently handled using various fast computational approaches, such as the fast multipole like methods [54,55] or interpolation-based approaches [57–59]. For $i_d > 0$, neighbor images in the near-zone evaluation are included and they require modifying the fast evaluation algorithms. Here, we use the box-adaptive integral method of [5,82], which is also related to the pre-corrected FFT method [62] and adaptive integral method [60,61]. We modify this approach to efficiently account for the inclusion of the periodic images. The idea is similar to the far-zone procedure in Sec. 5.2.2 in its first three steps, but it also adds another modified correction step.

Similar to the grid construction in Sec. 5.2.2, we construct uniform grids. However, there are two notable differences. The first difference is that the source and observer uniform grids coincide, i.e., there is a single uniform grid $\{\mathbf{r}_m = x_m, y_m, z_m\} = \{\mathbf{r}_m^o = x_m^o, y_m^o, z_m^o\} = \{\mathbf{r}_n^s = x_n^s, y_n^s, z_n^s\}$ by removing $\Delta_x, \Delta_y, \Delta_z$ in Eq. (5.15) and Eq. (5.16). The reason for using a single grid is that the near-zone PGF only involves a small number of terms in its evaluation and does not need to have a spatial separation between source and observer points. The second difference is that the number of the grid points N_g is significantly larger than what is used in the far-zone evaluation. Typically, N_g is comparable to the number of the non-uniform points N for the optimal performance. This large N_g is required to have N_s (as defined in Eq. (5.19)) to be of $O(1)$ to result in a reduced number of computations in the correction step 4. Based on this grid definition, the near-zone evaluation proceeds in the following four steps.

Step 1, Projection from the source points to the source grid: This step is similar to step 1 in Sec. 5.2.2, where G_{far}^{p} is replaced with G_{near}^{p}. Here, N_g is typically much larger than N_s, and the matrices of the interpolation coefficients are sparse. The resulting computational cost is of $O(N_g) = O(N)$.

Step 2, Evaluation of the PSP at the observer grid: This step is similar to step 2 in Sec. 5.2.2, where G_{far}^{p} is replaced with G_{near}^{p}. However, because N_g is large, the convolution sum of Eq. (5.21) is computed via FFT, which is allowed because the grids are uniform. Since G_{near}^{p} is not periodic, using FFT requires making the equivalent matrix circular, which involves doubling its size in each periodicity dimension. For performing FFTs in the computation stage, G_{near}^{p} is tabulated in the pre-processing stage. The tabulation computational cost is of $O(N_g) = O(N)$ and the FFT computational cost is of $O(N \log N)$.

Step 3, Interpolation from the observer grid to the observer points: The interpolation step for u_{near} is identical to step 3 in Sec. 5.2.2 for u_{far} and it can be viewed as an inverse of step 1 using the same sparse matrix for the interpolation coefficients. The resulting computational cost is of $O\left(N_g\right) = O\left(N\right)$.

Step 4: Error correction: The ability to perform the projection/interpolation in steps 1 and 3 rely on the assumption of slow variations of the Green's function. For the far-zone evaluation, G_{far}^{p} indeed varies slowly in the domain of the zeroth unit cell because it excludes the direct and near-zone image interactions between the sources and observers. The near-zone component with $G_{near}^{p} G_{near}^{p}$, however, involves direct and near-image interactions, which result in rapid spatial variations, e.g., the rate of such spatial variations is unbounded when the source and observer coincide. Therefore, the interpolations for the nearby source-observer pair contributions are highly inaccurate, and they need to be corrected. To eliminate these errors, we define an error correction range D^{ER} for each box (defined by $\Delta_x, \Delta_y, \Delta_z$ in Eq. (5.17)), determined by the uniform grid. This range is of the same order as $\max\{\Delta_x, \Delta_y, \Delta_z\}$. For all observer points in a box, the corrections are performed in the error-correction region, $\Omega^{ER}(\mathbf{r}_m)$, for all the non-uniform sources in the same box and a certain number of surrounding boxes that are withing the D^{ER} distance from the box. The correction involves subtracting the contributions due to steps 1-3 and adding the exact point-to-point superposition sums via appropriate contributions in Eq. (5.1). The error correction procedure can be expressed as

$$u_{near}(\mathbf{r}_m) = u_{near}(\mathbf{r}_m) + \sum_{n}^{\mathbf{r}_n \in \Omega^{ER}(\mathbf{r}_m)} [G^p_{near}(\mathbf{r}_m - \mathbf{r}_n) - G_{grid}(\mathbf{r}_m - \mathbf{r}_n)] q(\mathbf{r}_n) \tag{5.23}$$

$$G_{grid}(\mathbf{r}_m, \mathbf{r}_n) = \sum_{m'}^{\omega^o(\mathbf{r}_m, \mathbf{r}_{m'}) \neq 0} \sum_{n'}^{\omega^s(\mathbf{r}_{n'}, \mathbf{r}_n) \neq 0} \omega^o_{nufft}(\mathbf{r}_m, \mathbf{r}_{m'}) G^p_{near}(\mathbf{r}_{m'} - \mathbf{r}_{n'}) \omega^s_{nufft}(\mathbf{r}_{n'}, \mathbf{r}_n) \tag{5.24}$$

where the error is corrected in the region $\Omega^{ER}(\mathbf{r}_m)$, i.e., for any point $\mathbf{r}$ within this region, $|\mathbf{r} - \mathbf{r}_m| \leq D^{ER}$. Similar to Eq. (5.20) and Eq. (5.22), the ω^s_{nufft} and ω^o_{nufft} correspond to the interpolation matrices containing the weights for the above step 1 and 2. The function G_{grid} represents the near-zone Green's function between the source and observer points obtained via the grid interactions with interpolation, which needs to be subtracted in the error correction process.

While the procedure in Eq. (5.23) and Eq. (5.24) can be efficient, it still can significantly increase the computational cost due to the need to account for all the near-zone images via G^p_{near}. For example, in the 3D periodicity case, when including 2 surrounding boxes per each dimension

($i_d = 2$), there are 125 ($(2i_d + 1)^3$) images in total, which leads to a high additional computational cost.

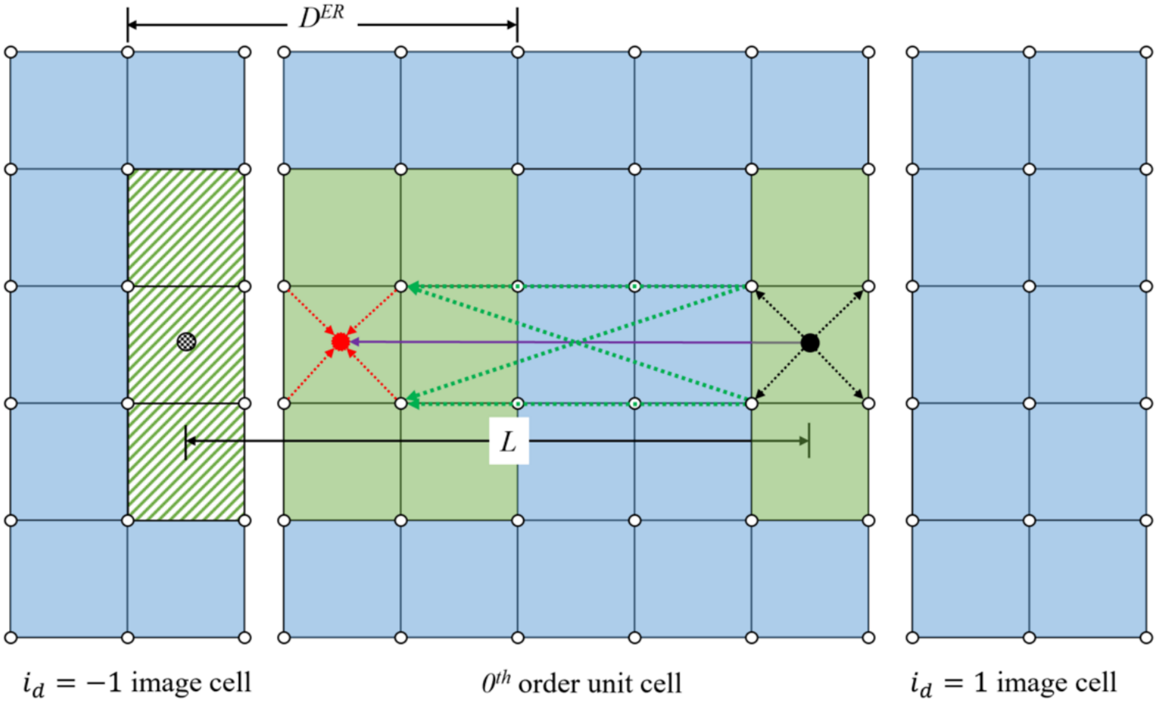
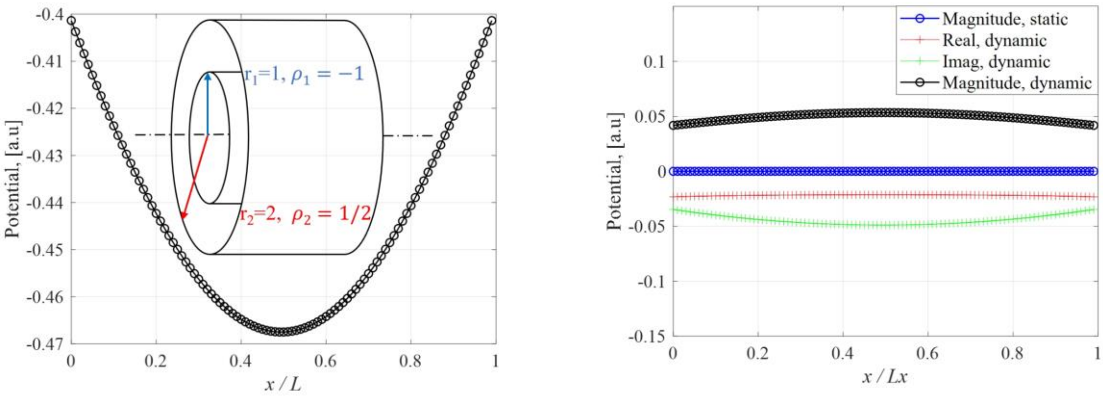


Figure 5.4: The error correction step 4 for the near-zone PSP component. The black dot is a source point and red dot is an observer point. The green region is the error-correction region Ω^{ER}. The black circles are grid points. Direct calculation within this region is done by subtracting the grid interaction inside the green region and adding the direct interactions.

This computational cost can be much reduced by including only a single extra periodic image per periodicity dimension. This is possible due to the fact that $\{\Delta_x, \Delta_y, \Delta_z\}$ are typically much smaller than $\{D_x, D_y, D_z\}$, so that most of the images correspond to a large spatial separation. For example, in Fig. 5.4 showing an example for a 1D x-dimension periodicity case, for the sources in the internal boxes of the zeroth (middle) unit cell, only the $G_0(\mathbf{r}_m - \mathbf{r}_n)$ component of the $G^p_{near}(\mathbf{r}_m - \mathbf{r}_n)$ can be used for the near-zone correction. On the other hand, for the left box, which is next to the left edge of the zeroth unit cell (the left green box with red dot and arrows), in addition to the $G_0(\mathbf{r}_m - \mathbf{r}_n)$ component, the component of $G_0(\mathbf{r}_m - \mathbf{r}_n - L_x\hat{\mathbf{x}})$ of $G_{near}^p(\mathbf{r}_m - \mathbf{r}_n)$ needs to be accounted for. The latter component corresponds to the sources from the right box (right green box with a black dot and arrows) contributing to the left box observers via the $i_x = 1$ image. These sources are equivalent to the sources in the right box of the left image cell (the dashed green box with a grey dot) contributing to the left box observers via the image Green's function $G_0(\mathbf{r}_m - \mathbf{r}_n)$, i.e., they are equivalent to the geometrically close boxes. This procedure is generalized to 2D and 3D periodicities by including corresponding required images. The computational cost of this procedure is only slightly higher than that for a non-periodic case, in which only the $G_0(\mathbf{r}_m - \mathbf{r}_n)$ contribution is required. The cost increase is associated with the need to include additional images at the sources associated with the boxes located at the corners, edges, and sides of the computational domain. There is a small number of such boxes as compared to the total number of near-zone images, so that the computational cost increase is insignificant. The overall computational cost of this step is of $O(N)$.

5.3 Numerical results

We implemented FFT-PIM both in a Central Processing Unit (CPU) and Graphics Processing Unit (GPU) based code. This section presents numerical test results demonstrating the accuracy and performance of the FFT-PIM. Sec. 5.3.1 presents error analysis, comparing the results from the FFT-PIM with results obtained via the direct sum of Eq. (5.1) with Eq. (5.8) for different problem sizes, uniform grid sizes, and interpolation orders. Sec. 5.3.2 examines the computational speed of the FFT-PIM for different parameters, such as problem size, uniform grid size, and the periodicity dimension.



(a) (b)

Figure 5.5: Potential on the x axis of a coaxial structure. (a) Coaxial structure unit cell ($L=1$) and potential of the non-periodic unit cell. Inner radius $r_1=1$ with negative line charge density $\rho_1=-1$, outer radius $r_2=2$ with positive line charge density $\rho_2=1/2$. (b) PSP with a 1D periodicity along the x axis with $L_x=1$ for $k_0=k_{x0}=0$ and a $k_0=1$, $k_{x0}=1-j$.

We start by showing results for the calculated potential due to a source distribution (Fig. 5.5). The structure is 1D periodic in the x-direction with the periodicity of $L_x=1$ and the unit cell comprises a section of coaxial cable along the x-axis that extends through the entire period. The inner and outer radius is $r_1=1$ and $r_2=2$, respectively. Negative and positive sources (e.g., charges) are distributed uniformly on the inner surface with charge density $\rho_1=-1$ and $\rho_2=1/2$, respectively, resulting in a neutral unit cell. We first calculate the potential on the central x axis for a non-periodic case, as shown in Fig. 5.5 (a) and we observe significant edge effects in terms of the potential spatial dependence. We then use the FFT-PIM with a 1D periodicity, making it into an infinitely long coaxial source distribution. For the NPSP case, we find that the potential becomes near zero everywhere (blue curve in Fig. 5.5 (b)), the edge effect is eliminated, which agrees with the analytical result. We also calculate the potential for the dynamic case with the wavenumber $k_0=1$ and a complex phase shift $k_{x0}=1-j$, in which case the potential becomes complex with non-zero magnitude and corresponding real/imaginary parts (black, red, and green curves in Fig. 5.5 (b)).

We, then, proceed with showing results for the error analysis and computational performance.

5.3.1 Error analysis

To test the error of the FFT-PIM, we use an infinite long 1D bar along the x axis, and the target object is a cube of the size $D_x, D_y, D_z = 50$ with 1D x-direction periodicity with $L_x = D_x = 50$. We assume a NPSP case, i.e., $k = k_{x0} = 0$. We mesh the cube with a tetrahedral mesh and the total number of the mesh points (vertices) is $N = 7189$. In order to calculate the error, we randomly set 7000 mesh points with finite values as source points while the rest 189 points are set to 0, and make sure that $\sum_{l=1}^{N} q_l = 0$. The 189 mesh points set to 0 are considered as observer points, and the locations of these observer points are chosen such that they are spatially shifted as compared to the source points such that the PGF in Eq. (5.8) converges for all source-observer pairs when the direct superposition sum is used. We compare the results for relative errors between the FFT-PIM and the direct computation via Eq. (5.8), which is equivalent to Eq. (5.5) as shown in Sec. 5.2. We tested the relative errors at the observer points with different interpolation orders, uniform grid size, and number of subtracted near-zone images. The results are shown in Fig. 5.6. We find that with cubic interpolation order and a uniform grid of size $N_{gx}, N_{gy}, N_{gz} = 10$, by subtracting the zeroth order and only the nearest neighbors ($i_d = 1$), we can achieve an error at the level of 10^{-3}, which is sufficient for many practical cases. The relative error is further reduced by increasing the number of grid points, order, and the near-zone images.

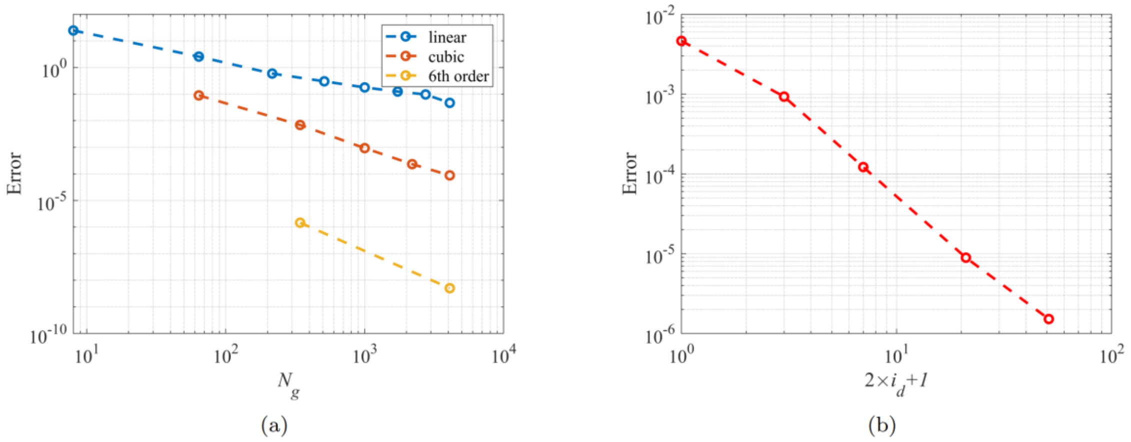


Figure 5.6: Relative error of the far-zone PSP for a NPSP case with a 1D periodicity along the x-axis. (a) Relative error for 1st, 3rd, and 6th order interpolation with varying uniform grid sizes with $i_d = 1$. (b) Relative error of cubic interpolation with 1000 sparse grid points and different numbers of subtracted near-zone unit cells.

5.3.2 Computational Performance analysis

To benchmark the performance of the FFT-PIM, we conducted a set of tests on Purdue Anvil cluster at Rosen Center for Advanced Computing (RCAC) in Purdue University. CPU tests were run on eight cores of AMD Epyc "Milan" processors and GPU tests were run on Nvidia A100. The results are shown for a 3D NPSP problem. The target object is a cube with dimensions $D_x, D_y, D_z = 100$ and $L_{x,y,z} = D_{x,y,z} + 1$. This problem has $L_{x,y,z} \approx D_{x,y,z}$ and it represents the maximal computational complexity as compared to 1D and 2D periodicity cases. The configuration of the far-zone calculation is set as cubic interpolation with uniform grid size of $N_{gx}, N_{gy}, N_{gz} = 10$, and $i_d = 1$. The near-zone is handled using the procedures from the FastMag micromagnetic simulator [74,82] for non-periodic and periodic cases. When periodicity presents, it is modified to allow for an efficient handling of periodicities, as outlined in Sec. 5.2.3.

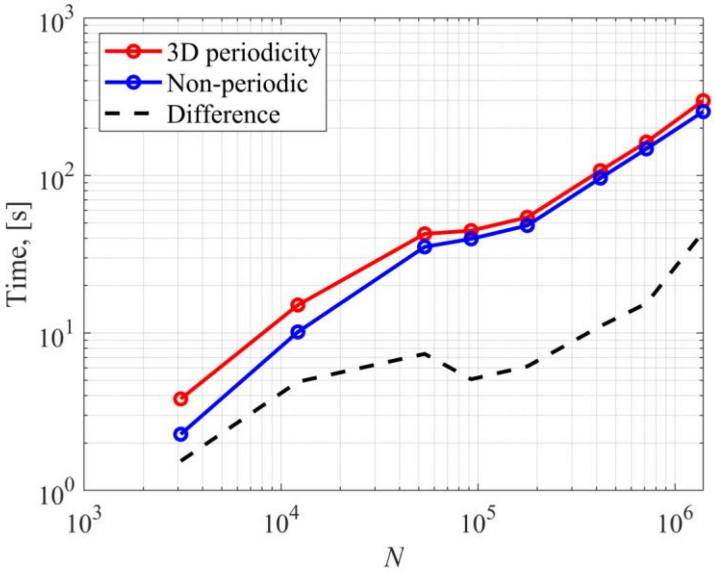


Figure 5.7: Prepossessing time for the non-periodic case, NPSP case with 3D periodicity, and for their difference.

Fig. 5.7 shows the computational time on CPU for pre-processing needed for computing the interpolation coefficients and PGF tables. We compare the CPU time with the case of the same source - observer distributions but without any periodicity, which demonstrates the overhead due to the periodicity. The predominant additional time complexity arises from substituting the original free-space Green's function, G_0, with a more complicated G_{near}. Specifically in this 3D case with $i_d = 1$, G_{near} is $(2\times1+1)^3$ times slower than G_0. However, the overall discrepancy is marginal, resulting in only a 15% overhead. This small impact is due to the fact that tabulating the Green's function constitutes only a small part of the pre-processing time, and the time complexities of other segments do not increase as the periodicity is introduced. We also evaluated 1D and 2D periodic cases and obtained similar performance with a reduced overhead.

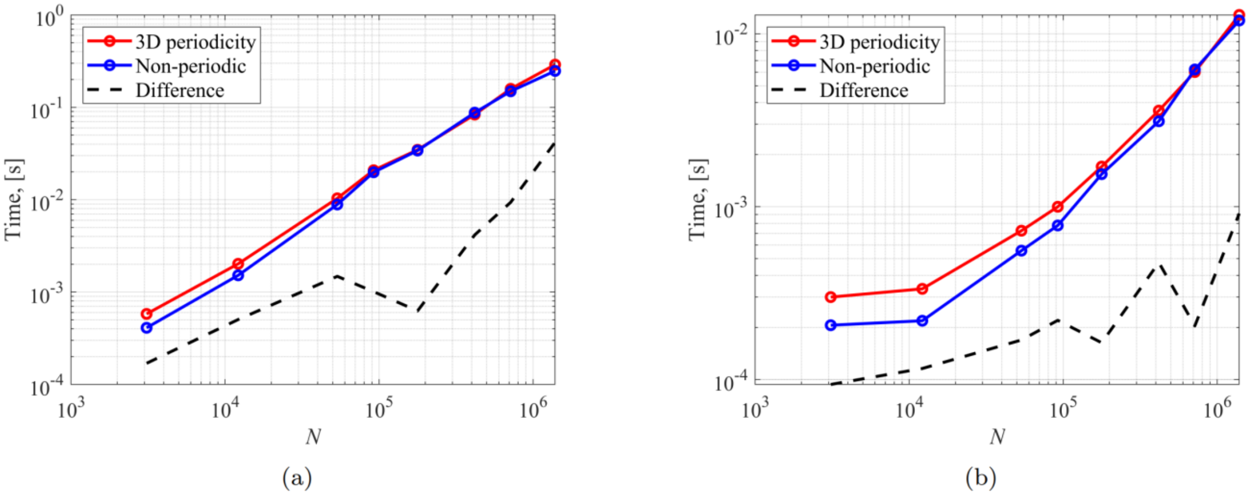


Figure 5.8: Execution time versus N of evaluating the non-periodic potential and NPSP PSP with 3D periodicity on (a) eight-core CPU and (b) on GPU.

Fig. 5.8 shows the execution time for a single evaluation of PSP on CPU and GPU. When compared to a non-periodic problem, the major execution time increase is due to the evaluation of the far-zone PSP component, which can be performed separately from the near-zone evaluation. The presence or absence of periodicity has minimal impact on the near-zone evaluation since all Green's function values are tabulated during the pre-processing step. Consequently, in the eight-core CPU benchmark, only a minor difference (< 15%) is observed between configurations with and without periodicity for moderately large case where $N > 10^4$. Furthermore, in the GPU benchmark, NVIDIA CUDA multi-stream concurrency enables simultaneous implementation of both near-zone and far-zone evaluations. This approach effectively eliminates the overhead induced by the far-zone evaluation, which is already marginal in eight-core CPU execution, resulting in a small difference (< 5%) between cases with and without periodicity, including the 3D periodicity cases. The execution performance on GPU is generally 20-30X faster than that on eight-core CPU for the tested sizes, which is equivalent to around ~200X speedup between GPU and single CPU core. The 1D and 2D periodicity cases exhibit a similar performance, indicating a highly efficient implementation for all 1D, 2D, 3D periodicity types. To further present differences between the near-zone and far-zone component, we show the grid size of the near-zone PSP component in Table 5.1. For most practical problem sizes, the near-zone grid in Table 5.1 is much denser than the far-zone grid.

Table 5.1: Grid size for near-zone PSP component evaluation. The near-zone grid is much larger than the far-zone grid when the problem size N is large. Second-order projection/interpolation for near-zone evaluation is used.

N	Near-zone grid size
3096	17^3
12175	25^3
53601	41^3
92233	49^3
177973	59^3
418308	77^3
1391742	111^3

From Fig. 5.8 we observe that the near-zone component PSP dominates the computational load for the 1D, 2D, and 3D periodicity cases. Similarly, Fig. 5.6 reveals that the error associated with the far-zone PSP component can be reduced with minimal adjustments to the parameters and limited overhead. This observation suggests that the near-zone PSP component is a critical factor in determining the balance between overall performance and error rates. To further explore this relationship, we conducted a series of benchmarks with a cube meshed in its volume, containing $N = 53601$points for a coarser mesh and $N = 510643$ for a finer mesh. The results are shown in

Fig. 5.9. The performance shown here demonstrate a good performance compared to available solutions [64].

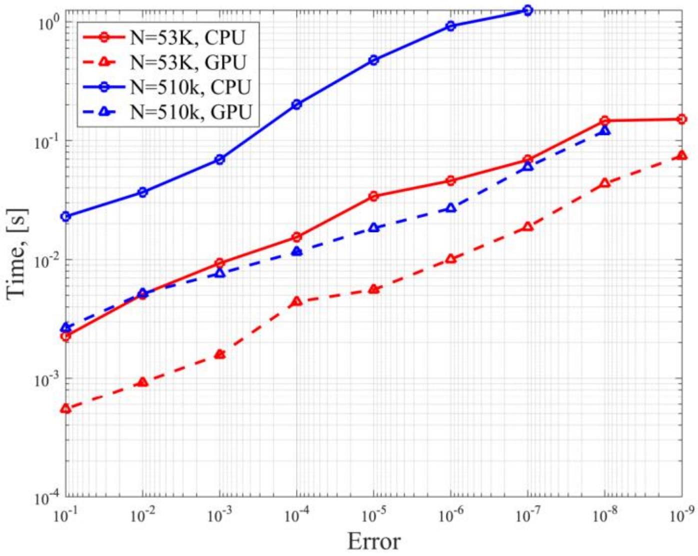
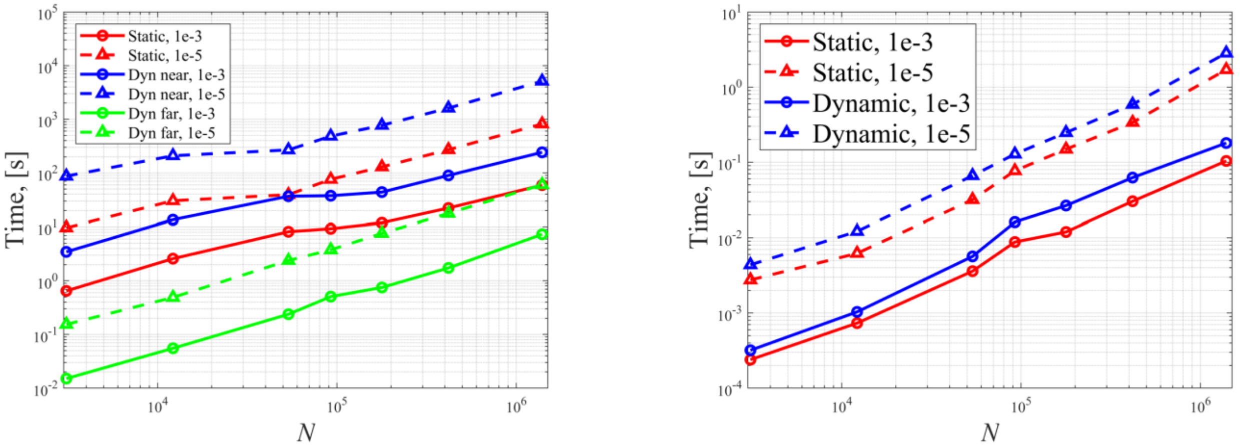


Figure 5.9: Execution time versus the relative error for the near-zone PSP component on CPU and GPU for a 3D NPSP case with $N = 53K$ and $N = 510K$.

Fig. 5.10 compares the preprocessing and computational time of the NPSP and dynamic cases for a 3D periodicity of $L_x = 101$, $L_y = 101$, $L_z = 101$ for relative errors of 10^{-3} and 10^{-5}. The NPSP case parameters are as those in Fig. 5.8 and Table 5.1 for error level 10^{-3}, and we double the near-zone grid size for error level 10^{-5}. The dynamic cases have phase shifts of $k_{x0} = 0.01 - 0.01j$, $k_{y0} = 0.01 - 0.01j$, $k_{z0} = 0.01 - 0.01j$. For the dynamic case, the wavenumber k_0 was chosen such that the average distance between the sources/observers was $\lambda / 20$. This choice is typical when considering high-frequency dynamic problems and it allows resolving the wave process spatial variations. For the largest considered N, this choice results in the domain size of $L_x = L_y = L_z = 5\lambda$. We find that the computational time for the low- and high-frequency cases are close to each other, and they scale similarly as that for the non-periodic static case. The computational time of the dynamic cases is around 2 times greater, which is related to performing complex-valued versus real-valued operations.



(a) (b)

Figure 5.10: (a)Preprocessing time and (b) execution time versus N of evaluating the PSP for a 3D NPSP and high-frequency dynamic cases for the relative errors of $1e-3$ and $1e-5$.

5.4 Summary

We introduced an efficient and flexible FFT-PIM for computing PSP in this chapter. FFT-PIM can be used for a wide variety of problem types with the same numerical implementation. It allows computing PSP for a non-uniform source distribution, works for arbitrary 1D, 2D, and 3D periodicities, can operate with or without a phase shift between the periodic boundaries, and is applicable to dynamic and static problems. A requirement is imposed for using FFT-PIM for NPSP problems, which is that the source must be neutral within the unit cell. Such a requirement is natural to many practical problems, such as problems dealing with electric dipoles, magnetization, and molecular structures.

FFT-PIM is based on a superposition sum between the source distribution and PGF. PGF, which originally is defined as an infinite sum that does not have a rapid convergence, can be represented as an exponentially convergent sum in terms of spectral series expansions, which are given for the dynamic case and static case with and without a periodic phase shift. The expansions converge under the assumption of a sufficient separation between the source and observer points. To lead to a fast superposition sum computation, PGF and the corresponding PSP are represented in terms of the near- and far-zone components. The near-zone component includes a finite, typically small, number of periodic images, whereas the far-zone component includes all the rest of an infinite number of images. The far-zone component is recognized as a slowly varying function of the spatial coordinates, which allows computing the far-zone PSP component by first computing it on a sparse uniform grid, i.e., at a small number of points, and then interpolating it to all the required non-uniform observation points by local interpolation. This process involves defining shifted source and observer grids to allow for a rapidly convergence of the spectral sums for PGF. This process requires a small number of operations and low computational cost, while allowing for a rapid converge. The near-zone PSP component can be evaluated by any fast method, but it required accounting for the additional near-zone images, which may substantially increase the computational cost and memory consumption. An approach based on the box-adaptive adaptive integral method is presented that allows evaluating the superposition sum for the far-zone component rapidly based on including essential images only at the required locations near the periodic boundaries. The result is an approach that has an overall computational cost of $O(N \log N)$ and memory consumption of $O(N)$. The presented results demonstrate the high convergence, accuracy, and computational performance of the FFT-PIM.

We note that FFT-PIM can be regarded as an extension of the fast periodic interpolation method [68] and the non-periodic box-adaptive integral method [82] or pre-corrected FFT method [62]. With respect to the fast periodic interpolation method [68], the extensions are in enabling both dynamic and static cases with any phase shifts or no phase shift as well as in enabling the complete fields computation, including fast near-zone computations via an FFT-based method without a need of extending the computational domain. With respect to the non-periodic box-adaptive integral method [82] or pre-corrected FFT method [62], FFT-PIM allows considering problems with 1D, 2D, and 3D periodicities with almost the same computational cost as the non-periodic approaches. We also note that the approach of separating the computations into the near- and far-zone components is related to other fast evaluation techniques [62–66,68,82] for periodic and non-periodic domains. Contributions of this work are in the efficient evaluation of the far-zone component via the properly chosen uniform grids and Green's function evaluation as well as in the adaptation of the near-zone component computation via an FFT-based method.

FFT-PIM can be used in several problems in electromagnetics, acoustics, and quantum mechanics. The evaluation of the periodic sums with dynamic (Helmholtz potential) PGF can be used in the context of electromagnetic and acoustic integral equations for characterizing wave propagation, radiation, scattering, and dispersion diagrams for periodic arrays. The evaluation of the periodic sums with NPSP PGF can be used for modeling infinite periodic arrays or mimic infinite domains in electro-/magneto-statics, micromagnetics, and density functional theory. The evaluation of the periodic sums with static PGF with a phase shift can be used to calculate dispersion diagrams in micromagnetics, e.g., spin waves or linearized density functional theory.

While FFT-PIM is a powerful method, it also has some limitations. In particular, for high-frequency problems with electrically large periodicities, the computation of PGF may become slow. This time may be reduced by using alternative methods for computing PGF, e.g., see Refs. [69] and [83]. Additionally, using FFTs mean that the computations are done on the entire FFT grid, which may be inefficient for problems of curved linear or surface domains or in cases of highly non-uniform domains with dense constellations of sources and observers in certain locations. For such problems using methods such as Fast Multiple Method [54], [84] or Non-Uniform Grid

Interpolation method [58] can be efficient. These methods can substitute the FFT-based approach for the near-zone component and the ideas presented here can be adapted to extend these methods to account for the additional images required due to periodicity.

We open-sourced our far-zone component code package Periodic Unit Far Field (PUFF) under Apache-2.0 license on GitHub (https://github.com/UCSD-CEM/PeriodicUnitFarField).

Acknowledgements

Chapter 5, in full, is a reprint of the material as it appears in F. Ai and V. Lomakin, “Fast Fourier Transform periodic interpolation method for superposition sums in a periodic unit cell”, Comp. Phys. Comm., Volume 304, 109291 (2024). The dissertation author was the primary researcher and author of this paper.

Chapter 6 Periodic micromagnetic finite element method

Periodic structures are common in many fields of study, such as crystals [85,86], molecular dynamics [87,88], electromagnetics among others. Periodic structures are also of a high importance in micromagnetics [89,90]. For magnetic structures, periodicity can be used to mimic infinite domains, e.g., infinite wires, films, or bulk. Periodicity can be used to account for spin wave propagation. It also can be used to approximate structures that are large in one of their dimensions.

There is a range of methods used in various fields of study to account for periodicity, or specific algorithms customized for micromagnetics [63,68,76,79,91]. In Micromagnetics, approaches accounting for periodicity are based on finite difference method (FDM) based domain discretization schemes with uniform grids [73]. For a uniform grid with N grid points, the magnetostatic field accounting for periodicity can accomplished using the Fast Fourier Transform (FFT) method in $O(N \log N)$ computational cost. Even for the FDM, one needs to be careful in the approaches of computing the periodic superposition kernel, referred to as periodic Green's function (PGF) as such computational can become slow or even lead to improper results [92]. The situation is more complicated when using micromagnetic codes based on the finite element method (FEM) [93]. For FEM, the structure to be modelled is meshed into a generally non-uniform mesh, which is often tetrahedral or hexahedral. Such meshing allows for a great flexibility in modeling complex materials and devices. However, the mesh non-uniformity makes it more complicated to account for general periodicities. Currently, there are no reported numerical methods or codes for micromagnetics that can handle general periodic structures.

Here, we introduce a formulation, referred to as periodic micromagnetic finite element method (PM-FEM) to handle periodicities in complex micromagnetic simulations using FEM.

PM-FEM offers a unified framework incorporating 1D, 2D, and 3D periodicities. For the computation of the exchange field, the PM-FEM modifies the exchange matrix to make use of the corresponding (1D, 2D, 3D) PBC. For the computation of the magnetostatic field, the PM-FEM adapts the Green's function to PGF with a modified pre-corrected Fast Fourier Transform or adaptive integral method to calculate the periodic scalar potential (PSP) followed by using a periodized gradient operator. Additionally, PM-FEM allows accounting for periodic unit cells of different configurations, including cases of touching and non-touching boundaries as well as cases of contained and protruding unit cells. In this chapter we first present the formulation of the problem, followed by the implementation details for the localized and non-localized interactions. We then show results obtained using the PM-FEM method, including its performance and error analysis. Finally, we give the summary and conclusion.

6.1 Problem formulation

6.1.1 Continuous formulation

We consider a magnetic structure characterized by the magnetization $\mathbf{M}(\mathbf{r})$, where $\mathbf{r}$ is the location vector defined in a periodic unit cell with 1D, 2D, or 3D PBC (Fig. 6.1). For the 3D PBC the periodic unit cell has periodic dimensions (periods) of L_x, L_y and L_z. For the 2D PBC, the periodic dimensions are L_x, L_y whereas the other z dimension is non periodic. For the 1D PBC, the periodic dimension is L_x, whereas the other (y and z) dimensions are non-periodic. The magnetization satisfies the following periodic condition:

$$\begin{aligned}
&\mathbf{M}(\mathbf{r}_n + L_x\hat{x}) = \mathbf{M}(\mathbf{r}_n),\\
&\mathbf{M}(\mathbf{r}_n + L_x\hat{x} + L_y\hat{y}) = \mathbf{M}(\mathbf{r}_n),\\
&\mathbf{M}(\mathbf{r}_n + L_x\hat{x} + L_y\hat{y} + L_z\hat{z}) = \mathbf{M}(\mathbf{r}_n).
\end{aligned} \tag{6.1}$$

There are two orthogonal PBC categories that require a special attention in numerical treatment, touching-(T-)/non-touching-(NT-)PBC and protruding (P-)/non-protruding (NP-)PBC.

For T-PBC, such cases occur when for a point at location $\mathbf{r}_i, i \in [1, N]$, one of its periodic images $\mathbf{r}_i + \gamma L_x \hat{x} + \zeta L_y \hat{y} + \xi L_z \hat{z}$ can still be found within the 0^{th} unit cell, namely we can find another point located at $\mathbf{r}_j = (x_i + \gamma L_x, y_i + \zeta L_y, z_i + \xi L_z), j \in [1, N], j \neq i$, Here, γ, ζ and ξ are zero if no periodicity is present at the $\hat{x}$, $\hat{y}$ or $\hat{z}$ direction, and they can be 1 if a periodicity is present. We also have $\gamma + \zeta + \xi > 0$ to avoid $j = i$, if for any $\mathbf{r}_i$, we can't find the corresponding $\mathbf{r}_j$, then it is NT-PBC. Regarding of the P-PBC, it is defined as the unit cell is tightly confined by the periodic length, namely the unit cell size is strictly less or equal than periodic length on all directions, and NP-PBC is defined if the condition does not hold. These two categories are orthogonal to each other, and all periodic problems can be classified as arbitrary combination of these two categories, e.g. T-NP-PBC, NT-P-PBC etc. Here we show some 1D PBC examples for illustration purpose.

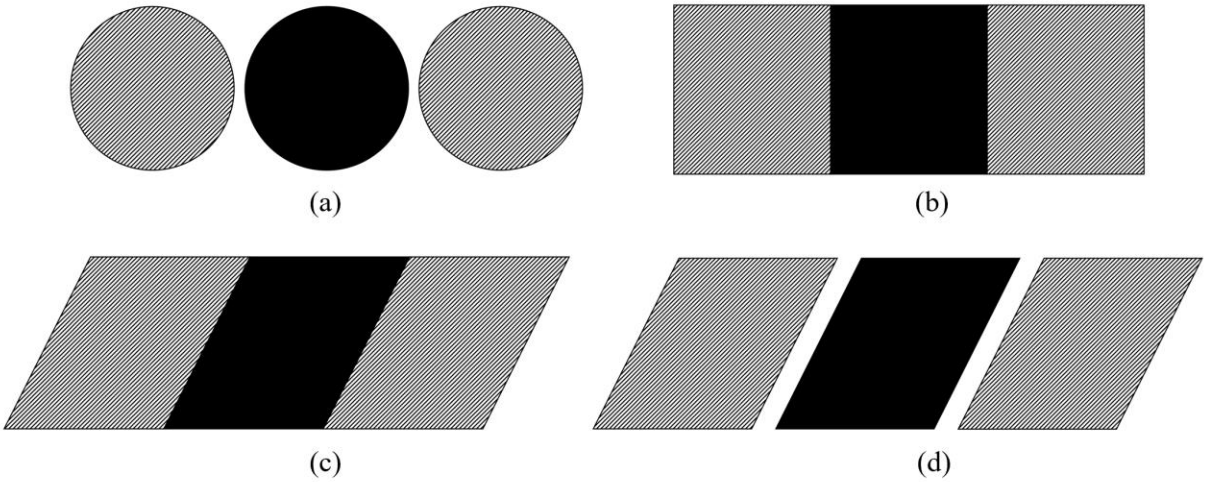


Figure 6.1: Categories of 1D PBC unit cell (solid black) and its nearest images (shadowed), (a) non-touching, non-protruding case, (b) touching, non-protruding case, (c) non-touching, protruding case and (d) touching, protruding case.

Fig. 6.1(a) shows a NT-NP-PBC unit cell, it can be represented as an infinite array of spheres, Fig. 6.1(b) shows a T-NP-PBC, it can be used to model an infinite wire, and Fig. 6.1(c) and Fig. 6.1(d) shows the NT-P-PBC and T-P-PBC respectively. All these cases can be met in micromagnetic simulations and need to be addressed. Specifically, T-PBC (either protruding or non-protruding) requires a special treatment for handling relevant differential operators whereas the protruding PBC (either touching or non-touching) requires special handling of the long-range superposition integrals / sums.

The magnetization dynamics is governed by the Landau-Lifshits-Gilbert equation as shown in Eq. (2.1) and the total effective field can be decomposed into several sub-components as indicated in Eq. (2.2). The effective field and all its sub-components satisfy the periodicity conditions, as in Eq. (6.1).

6.1.2 Discrete formulation

For numerical calculations, we utilize the finite element method (FEM), in which the unit cell is meshed into a set of elements. We choose tetrahedrons as the discretization elements for a broad availability of meshers but also hexahedrons can be a good choice. The magnetization is represented as

$$\mathbf{M} = \sum_{n=1}^{N} \mathbf{M}_n \varphi_n(\mathbf{r}), \tag{6.2}$$

where $\mathbf{M}_n = \mathbf{M}(\mathbf{r}_n)$ and $\varphi_n(\mathbf{r})$ are basis functions chosen as linear interpolatory polynomials. The solutions for the magnetization $\mathbf{M}_n$ are obtained at points $\mathbf{r}_n$ and the magnetization at any other point can be found via Eq. (6.2).

The discretization in Eq. (2.2) leads to a a unit cell that contains N coinciding source (magnetizations) / observer (effective fields) points. Each point is located at $\mathbf{r}_n = (x_n, y_n, z_n), n \in [1, N]$ and has magnetization $\mathbf{M}(\mathbf{r}_n) = (M_n^x, M_n^y, M_n^z)$ with an infinite number of images in the corresponding periodic dimensions. The magnetization in each image satisfies the following periodic conditions:

$$\begin{aligned}&\mathbf{M}(\mathbf{r}_n + i_x L_x \hat{x}) = \mathbf{M}(\mathbf{r}_n),\\ &\mathbf{M}(\mathbf{r}_n + i_x L_x \hat{x} + i_y L_y \hat{y}) = \mathbf{M}(\mathbf{r}_n),\\ &\mathbf{M}(\mathbf{r}_n + i_x L_x \hat{x} + i_y L_y \hat{y} + i_z L_z \hat{z}) = \mathbf{M}(\mathbf{r}_n).\end{aligned} \tag{6.3}$$

The integer indices i_x, i_y and $i_z \in [-\infty, \infty]$ if the periodicity is present at that specific direction, representing the image's order with respect to the original 0^{th} unit cell ($i_x = i_y = i_z = 0$).

Cases of T-PBC (Fig. 6.1(b) and Fig. 6.1(d)) need to be given a special care of as we mentioned before. In these T-PBC cases, the $\mathbf{M}_j$, as a periodic image of $\mathbf{M}_i$ on the PBC, is defined by Eq. (6.3) that solely depends on the value of its periodic counterpart $\mathbf{M}_i$, thus reducing the numerical problem size to less than N . The (i, j) index pairs that satisfy these requirements are referred to as PBC pairs, and we also refer the i^{th} point as the parent of the j^{th} point, and j^{th} point as the child of the i^{th} point. Furthermore, we merge these pairs to their lowest common ancestors (LCA) [94] by the union-find algorithm. Namely, for two PBC pairs (i, j) and (j, k), they are merged to (i, j) and (i, k). These can be proven by adding up their coordinates. A single LCA can have multiple children, e.g., a corner point of a cubic unit cell with 3D PBC can have up to 7 children. Hence, the number of all unique LCAs N_{LCA} is equal or less than the number of PBC pairs N_{PBC}, i.e. $N_{LCA} \leq N_{PBC}$. For a problem of size N with N_{PBC} PBC pairs, only $N - N_{PBC}$ points are unknown, reducing the actual problem size down to $N' = N \text{-} N_{PBC}$, so that $N' \leq N$. For simplicity, we assume that the first N' points are within the 0^{th} unit cell, and the rest $N - N'$ points are images due to the PBC.

Note that T-PBC can only appear at the direction with periodicity, yet this is not always true when periodicity is present. For example, consider the 3D periodic case with only $x-$ or

$y-$ direction having T-PBC while the $z-$ direction having a NT-PBC, e.g., an infinite set of infinite $x-y$ plane films of thickness d stacking over the $z-$ direction, with a gap of $L_z - d$.

The discretized LLG equation is written similar to its continuous counterpart:

$$\frac{\partial \mathbf{M}_n}{\partial t} = -\frac{\gamma}{1+\alpha^2}\left(\mathbf{M}_n \times \mathbf{H}_{\text{eff,n}} + \frac{\alpha}{M_{s,\text{n}}}\mathbf{M} \times \mathbf{M}_n \times \mathbf{H}_{\text{eff,n}}\right), \tag{6.4}$$

where the solutions are obtained for $\mathbf{M}_n$ and $\mathbf{H}_{\text{eff},n}$ are obtained at the locations $\mathbf{r}_n$ based on the FEM representation of $\mathbf{M}$ via the basis functions in Eq. (6.2). The time integration can be accomplished via various techniques. We implemented implicit time step adaptive predictor-corrector schemes based on explicit second order Adams-Moulton (midpoint) approach as well as time step and order adaptive backward differentiation formulas, both approaches having absolute stability up to the second-order time integration.

6.2 Implementation

In this section, we introduce methods to handle periodicity for computing the effective magnetic field components. The anisotropy field $\mathbf{H}_{\text{an}}(\mathbf{r}_n)$ and applied field $\mathbf{H}_{\text{ap}}(\mathbf{r}_n)$ are local in that they are solely determined by their location, so that their computation is straightforward. Approaches for computing the exchange and magnetostatic fields are described next.

For exchange interaction, modification is only required when PBC present, i.e. $N' < N$. For long-range magnetostatic interaction, the periodic scalar potential (PSP) is via a modified pFFT which employs periodic Green's function as the kernel. This can later be further accelerated by reducing the FFT size due to the periodicity of the kernel.

6.2.1 Exchange interaction

The exchange interactions are due to quantum mechanics effects that originates from Pauli repulsion. In the continuous micromagnetic representation, the exchange field is given by Eq. (2.9). The numerical representation of the exchange field is given by

$$\mathbf{H}_{ex}(\mathbf{r}_n) = \sum_{\mathbf{r}_m \in \Omega_n} \omega_{mn} \mathbf{M}(\mathbf{r}_m), \tag{6.5}$$

where $n \in [1, N'], m \in [1, N]$, ω_{mn} are elements of a sparse matrix that has band of non-zero values defined via Ω_n, such that each point is connected to the surrounding points through their connecting mesh edges [74]. For the NP-PBC case, the calculation of the exchange matrix is identical to that of the conventional case of non-periodic problems.

For the case of T-PBC, the sparse matrix construction needs to be modified for several cases.

Case 1: This case is for the interior points, such that the point at $\mathbf{r}_n$ is not the LCA of any other points and all the neighboring points within Ω_n are not LCA of any other points, i.e., $LCA(i) \neq n, i \in [1, N], i \neq n$ and $\forall m \in \Omega_n$ we have $LCA(j) \neq m, j \in [1, N], j \neq m$. Namely, both n and $m \in \Omega_n \in [1, N']$. This case represents the majority of the sparse matrix construction since most points within the unit cell volume that are not close to a PBC boundary fall within this category. The resulting sparse matrix elements of Eq. (6.5) are the same as for the non-periodic case.

Case 2: In this case, the point at $\mathbf{r}_n$ is not the LCA of any other points, and some of the points within Ω_n have different LCAs other than themselves. We need to map these points at $\mathbf{r}_m$ to their LCAs and Eq. (6.5) can be extended as

$$\mathbf{H}_{ex}(\mathbf{r}_n) = \sum_{\mathbf{r}_m \in \Omega_n} \omega_{mn} \mathbf{M}(\mathbf{r}_{LCA(m)}). \tag{6.6}$$

To retrieve the magnetization on the PBC from within 0^{th} unit cell. The weight ω_{mn} is not mapped, since though the magnetization is from the $\mathbf{r}_{LCA(m)}$, the location should still stay within Ω_n . Note that if m is not a child in a PBC pair, then its LCA is itself, i.e., $LCA(m) = m, m \in [1, N']$, which falls back to the original form of Eq. (6.5).

Case 3: In this case, the point at $\mathbf{r}_n$ is the LCA of some points on the boundary, i.e., $\exists n', n' \in [N'+1, N]$ and $LCA(n') = n$. Following the same approach, for each n' we can define its surrounding region as $\Omega_{n'}$. Then, the completed surrounding region of $\mathbf{r}_n$ is $\Omega_n \cup_{n'} \Omega_{n'}$, denoted as $\Omega_{n \cup \{n'\}}$. Here, $\{n'\}$ represent the union set of all n' . We can see that such new region falls back to the original Ω_n if $\{n'\} = \varnothing$. Similar to Eq. (6.5), we define all the points inside the $\Omega_{n \cup \{n'\}}$ with index m. We further note that an edge can occur multiple times in this object of size N, hence direct computations via Eq. (6.6) would introduce duplicated calculations. To avoid such duplication, we first defined a new set of indices m' and corresponding coordinates $\mathbf{r}_{m'}$. For any $m \in \Omega_{n \cup \{n'\}}$, a new m' and $\mathbf{r}_{m'}$ can be generated as

$$\begin{aligned} &LCA(m') = LCA(m), \\ &\mathbf{r}_{m'} - \mathbf{r}_m = \mathbf{r}_{n'} - \mathbf{r}_n. \end{aligned} \tag{6.7}$$

Combining all $\mathbf{r}_{m'}$ and m' together, we obtain a new location set $\{\mathbf{r}_{m'}\}$ and the corresponding indices set $\{m'\}$. Then, the desired locations and indices set represent a unique set of all $\mathbf{r}_{m'}$, written as $\{\mathbf{r}_{m'}\}^{\dagger}$ and their corresponding indices, denoted as $\{m'\}^{\dagger}$. Here, $\{\mathbf{r}_{m'}\}^{\dagger}$ and $\{m'\}^{\dagger}$ satisfies

$$\begin{aligned}&\{\mathbf{r}_{m'}\}^{\dagger} \subseteq \{\mathbf{r}_{m'}\}, \{m'\}^{\dagger} \subseteq \{m'\},\\ &\forall i \in \{m'\}, \exists j \in \{m'\}^{\dagger} \Rightarrow \mathbf{r}_i = \mathbf{r}_j \;\; \mathbf{r}_i, \mathbf{r}_j \in \{\mathbf{r}_{m'}\}^{\dagger},\\ &\forall \alpha, \beta \in \{m'\}^{\dagger}, \alpha \neq \beta \Rightarrow \mathbf{r}_{\alpha} \neq \mathbf{r}_{\beta}.\end{aligned} \tag{6.8}$$

This new location set can, then, form a new region as $\Omega^{\dagger}_{m'}$, which, combined with the original region Ω_n, determines a generalized surrounding region defined as $\tilde{\Omega}_n = \Omega_n \cup \Omega^{\dagger}_{m'}$. Finally with this new region, we can rewrite the Eq. (6.6) as

$$\mathbf{H}_{ex}(\mathbf{r}_n) = \sum_{\mathbf{r}_k \in \tilde{\Omega}_n} \omega_{kn} \mathbf{M}(\mathbf{r}_{LCA(k)}). \tag{6.9}$$

A fast implementation to construct the above generalized surrounding region defined in Eq. (6.8) is to place a filter in Eq. (6.7). Such filter rejects the indices between $N'+1$ and N, i.e., we only map the non-PBC location. In this case, the equality $\{\mathbf{r}_{m'}\}^{\dagger} = \{\mathbf{r}_{m'}\}, \{m'\}^{\dagger} = \{m'\}$ holds directly.

By combining Eq. **Error! Reference source not found.**, Eq. (6.5), Eq. (6.6) and Eq. (6.9), we can handle the exchange field when T-PBC is present.

6.2.2 Magnetostatic interaction

The magnetostatic field is due to long-range interactions, and it can be calculated either by solving the Poisson equation or by evaluating the superposition integrals [95]. Here, we evaluate the magnetostatic field using superposition integrals via the following formulation

$$\rho(\mathbf{r}) = \nabla \cdot \mathbf{M}(\mathbf{r}), \rho_s(\mathbf{r}) = -\mathbf{n} \cdot \mathbf{M}(\mathbf{r}), \tag{6.10}$$

$$u(\mathbf{r}) = \iiint_V G^p(\mathbf{r}-\mathbf{r}')\rho(\mathbf{r})d\mathbf{r}' + \iint_S G^p(\mathbf{r}-\mathbf{r}')\rho_s(\mathbf{r})d\mathbf{r}', \tag{6.11}$$

$$\mathbf{H}_{ms}(\mathbf{r}) = -\nabla u(\mathbf{r}). \tag{6.12}$$

Here, ρ is the volumetric magnetic charge density, ρ_s is the surface magnetic charge density, . . is the magnetic scalar potential, and G_p is the 3D periodic Green's function (PGF), which can be

1D, 2D, or 3D periodic [71,96]. A discrete FEM representation of Eq. (6.10), Eq. (6.11) and Eq. (6.12) can be given as

$$q(\mathbf{r}_n) = \sum_{\mathbf{r}_k \in \tilde{\Omega}_n} \omega_{kn}^{q} \mathbf{M}(\mathbf{r}_{LCA(k)}), \tag{6.13}$$

$$\tilde{u}(\mathbf{r}_n) = \sum_{k \in [1,N']} G^{p}(\mathbf{r}_n - \mathbf{r}_k) q(\mathbf{r}_k), \tag{6.14}$$

$$u(\mathbf{r}_n) = \tilde{u}(\mathbf{r}_n) + \sum_{\mathbf{r}_k \in \Omega_n^{\text{near}}} \omega_{kn}^{\text{near}} \mathbf{M}(\mathbf{r}_k), \tag{6.15}$$

$$\mathbf{H}_{ms}(\mathbf{r}_n) = -\sum_{\mathbf{r}_k \in \tilde{\Omega}_n} \omega_{kn}^{\text{grad}} u(\mathbf{r}_{LCA(k)}). \tag{6.16}$$

Here, $q(\mathbf{r}_n)$ are nodal magnetic charges assigned to each point $\mathbf{r}_n$, i.e., nodes of the FEM mesh. These charges are found via an FEM representation of the divergence operator in the volume and additional surface related charge components on the surfaces of the magnetization discontinuities, e.g., boundaries of the magnetized domain or boundaries between material of different saturation magnetization. The nodal charges are given as a sparse matrix-vector products with the sparse matrix elements ω_{kn}^{q} . Similarly, $\omega_{kn}^{\text{grad}}$ are elements of the sparse matrix representing the gradient operator. For the NT-PBC cases, the charge and gradient sparse matrices are constructed identically to non-periodic cases. For the T-PBC cases, the construction of these sparse matrices is modified similarly to the modifications in the Laplace matrix construction, as outlined in Sec. 6.2.1 to generalize the regions surrounding each point.

In Eq. (6.11), $\tilde{u}(\mathbf{r}_n)$ is the magnetic periodic scalar potential (PSP) at the mesh points and it is given as a standard superposition sum over the charges $q(\mathbf{r}_k)$ in the periodic unit cell, where $G^{p}(\mathbf{r}_n - \mathbf{r}_k)$ is the periodic Green's function (PGF) defined for the 1D, 2D, and 3D periodicities and can be found in Eq. (5.8), Eq. (5.9) and Eq. (5.10). The PSP can be calculated efficiently for the NP-PBC case and these approaches can be adapted to account for the P-PBC cases as discussed next.

For a given charge used for the calculation of the PSP, the PSP values are calculated accurately only at distances substantially distant from the charge, specifically, for distances greater than the surrounding element size. For smaller distances, the result is inaccurate, and it needs to be corrected. In Eq. (6.15), $u(\mathbf{r}_n)$ is the corrected PSP (C-PSP), which includes a correction term, where $\omega_{kn}^{\text{near}}$ are elements of the sparse correction matrix. The correction matrix has non-zero elements for a range of points Ω_n^{near} in the proper vicinity of $\mathbf{r}_n$. The correction sparse matrix elements are found through exact integrals of Eq. (6.14) performed over the relevant elements (tetrahedrons in our case).

Here we present an approach to account for the PBCs in the computations of $\mathbf{H}_{\text{ms}}$. With such a correction, the C-PSP can be computed with a controllable error.

Eq. (6.14) includes a superposition sum and if computed directly it has its computation cost scaling as $O(N^2)$, which is high for large N. We use a modified version of the pFFT method, also referred to as box-adaptive integral method (BAIM) that has a computational complexity of $O(N\log N)$. We define a Cartesian grid of size $N_x \times N_y \times N_z$ over the domain size of $D_x,\ D_y,\ D_z$ as

$$\begin{aligned} &x_{i,j,k} = (i-1)\Delta_x, \quad y_{i,j,k} = (j-1)\Delta_x, \quad z_{i,j,k} = (k-1)\Delta_x \\ &\Delta_x = \frac{D_x}{N_x - 1} \quad \Delta_y = \frac{D_y}{N_y - 1} \quad \Delta_z = \frac{D_z}{N_z - 1} \end{aligned} \tag{6.17}$$

with $i \in [1, N_x],\ j \in [1, N_y], k \in [1, N_z]$ and denote the grid points as $\mathbf{r}_{i,j,k} = (x_{i,j,k}, y_{i,j,k}, z_{i,j,k})$. The BAIM procedure follows the following four steps.

Step 1: Project non-uniformly distributed sources $q(\mathbf{r}_n)$ of Eq. (6.10) from locations $\mathbf{r}_n$ to the charges $q_{i,j,k}$ defined at the uniform Cartesian grid at $\mathbf{r}_{i,j,k}$.

Step 2: Calculate the PSP on all Cartesian grid points $\mathbf{r}_{i,j,k}$ as a convolution between PGF on $G_p(\mathbf{r}_{i,j,k})$ and changes $q_{i,j,k}$ on the grid, obtaining the PSP $u_{i,j,k}$ on the grid.

Step 3: Interpolate the grid PSP $u_{i,j,k}$ to the PSP at the non-uniform points $\mathbf{r}_n$.

The result from Steps 1-3 provide a PSP at the non-uniform points that has errors due to the fact that the projection and interpolation procedures are inaccurate from the PSP generated by sources that are located near the observers. To correct these inaccuracies, we implement a correction Step 4.

Step 4: Correct the errors introduced by projection and interpolation from Steps 1 and 3 to obtain the desired PSP $u(\mathbf{r}_n)$ with a controllable error level.

Compared with the non-periodic problems, two major modifications are made in these four steps.

First, in the original BAIM, Green's function in step 2 is the free space Green's function $G_0 = 1/|\mathbf{r}|$. To adapt the PBCs, the free-space Green's function is replaced with the PGF G_p and it can be computed efficiently.

The second major modification is for the error correction in Step 4. In general, the projection and interpolation only introduce large errors when projected/interpolated functions vary drastically within the region of interest. For the free-space Green's function G_0 it occurs when the separation between source and observer is small. We define a threshold $|\mathbf{r}_{ER}|$ such that for all neighboring sources within this range, contributions from these sources are calculated directly instead of the convolution in step 2. This can be done by subtracting the contributions from step 2 and adding the accurate contributions computed via a direct contribution.

For the free-space case, the threshold $|\mathbf{r}_{ER}|$ is found simply based on the source-observer distance. For the PBC cases, the periodic source images result in additional periodically copied points that also need to be included in the threshold $|\mathbf{r}_{ER}|$ range. Although we replace G_0 with G_p in Step 2, only G_0 varies drastically within nearby region. Therefore regardless of the periodicity configuration, the error correction is always calculating the contributions from sources through G_0, namely we only subtract contributions of G_0 in step 2 and add them back still using G_0.

The $|\mathbf{r}_{ER}|$ here defines a correction region for each point as Ω_n^{ER} that includes all qualified neighboring source points, this is usually defined based on the box size $\Delta_x, \Delta_y, \Delta_z$, e.g. $|\mathbf{r}_{ER}| = max(\Delta_x, \Delta_y, \Delta_z)$. This region needs to be expanded when periodicity presents. Similar to Sec. 6.2.1 we have different scenarios. We first define a symbol ι to represent the periodic direction, e.g. ι can be x for 1D periodicity, x, y for 2D periodicity and x, y, z for 3D periodicity. Then for a specific point at location $\mathbf{r}_n$ we have

Case 1: $\forall \iota \Rightarrow \mathbf{r}_n \hat{\iota} \pm |\mathbf{r}_{ER}| \in (D_\iota - L_\iota, L_\iota)$. This is the major case for all source points that are sufficiently far away from the boundary of periodic direction. In this case Ω_n^{ER} is unchanged.

Case 2: $\exists \iota \Rightarrow \mathbf{r}_n \hat{\iota} + |\mathbf{r}_{ER}| \geq L_\iota$ or $\mathbf{r}_n \hat{\iota} - |\mathbf{r}_{ER}| \leq D_\iota - L_\iota$. This means that the error correction region is including source points from nearest neighboring images. The error correction region then needs to be expanded. The original Ω_n^{ER} is defined within a region that satisfies $\forall \mathbf{r}_k \in \Omega_n^{ER} \Rightarrow \forall \iota, \mathbf{r}_k \hat{\iota} \in [\mathbf{r}_n \hat{\iota} - |\mathbf{r}_{ER}|, \mathbf{r}_n \hat{\iota} + |\mathbf{r}_{ER}|]$. We here defined a expanded error correction as $\tilde{\Omega}_n^{ER}$. For each scenario we have

$$\begin{aligned}&\forall \mathbf{r}_k \in \tilde{\Omega}_n^{ER} \Rightarrow \forall \iota, \mathbf{r}_k \hat{\iota} \in [\mathbf{r}_n \hat{\iota} - |\mathbf{r}_{ER}|, \mathbf{r}_n \hat{\iota} + |\mathbf{r}_{ER}|] \cup [0, \mathbf{r}_n \hat{\iota} + |\mathbf{r}_{ER}| - L_\iota],\\ &\forall \mathbf{r}_k \in \tilde{\Omega}_n^{ER} \Rightarrow \forall \iota, \mathbf{r}_k \hat{\iota} \in [\mathbf{r}_n \hat{\iota} - |\mathbf{r}_{ER}|, \mathbf{r}_n \hat{\iota} + |\mathbf{r}_{ER}|] \cup [L_\iota + \mathbf{r}_n \hat{\iota} - |\mathbf{r}_{ER}|, D_\iota].\end{aligned} \qquad (6.18)$$

In BAIM, the source points within each range can be easily found by searching "boxes" constructed by grid points of size $\Delta_x, \Delta_y, \Delta_z$. And all these source points' corresponding box index are pre-tabulated.

The construction of the sparse matrix $\mathbf{Z_0}(\mathbf{r}_n)$ in Eq. (6.16) also involves very similar modification, except that the error correction range $|\mathbf{r}_{ER}|$ is typically defined based on the tetrahedron size instead of the box size, i.e. the average edge length. Otherwise, we also need to expand the region searching for neighbor following the same style defined in in Eq. (6.18).

Here we first assume that the case is NP-PBC. This assumption naturally holds for most regular shapes, e.g. a cube, a cylinder and etc. However, this property is not guaranteed to be held with an irregular shape. In this case the judging condition in case 1 and 2 are not working anymore.

To fix these issues, here we propose a very simple solution. When calculating the $\mathbf{H}_{\mathrm{ms}}$ over the unit cell that $\exists \iota \Rightarrow D_\iota > L_\iota$. We define a new unit cell based on the original first N' points specifically for $\mathbf{H}_{\mathrm{ms}}$ that

$$\begin{aligned}&\mathbf{r}_{n'} = \underset{\delta}{\operatorname{argmin}} \sum_\iota |\mathbf{r}_n \hat{\iota} - \mathbf{r}_0 \hat{\iota} + \delta L_\iota|, \delta \in \mathbb{Z}\\ &H_{ms}(\mathbf{r}_{n'}) = H_{ms}(\mathbf{r}_n)\end{aligned} \qquad (6.19)$$

This equation defines a new unit cell with new coordinates $\mathbf{r}_{n'}$. The coordinates are shifted by arbitrary periodic length along each periodic direction to minimize the Manhattan distance with respect to a fixed point $\mathbf{r}_0$. For convenience we typically choose the geometry center of the original unit cell, namely $\mathbf{r}_0 = \sum_n \mathbf{r}_n / N'$. The new unit cell defined by Eq. (6.19) satisfies the condition that $\forall \iota \Rightarrow D_\iota \le L_\iota$. Since $\forall \mathbf{r}_{n'}\hat{\iota} \Rightarrow$ $\mathrm{mod}(\mathbf{r}_n\hat{\iota} - \mathbf{r}_0\hat{\iota}, L_\iota) \in [0, L_\iota)$. After solving the $\mathbf{H}_{\mathrm{ms}}$ on each shifted point, we can map all the values from $\mathbf{r}_{n'}$ back to $\mathbf{r}_n$ to restore the original unit cell.

6.3 Results

PM-FEM was implemented as a part of the FastMag micromagnetic simulator to allow modeling of complex magnetic materials and devices. The updated FastMag components include the magnetostatic and exchange field modules as well as the mesh construction module for touching and protruding unit cell types. In addition to these modules, FastMag implements implicite time stepping schemes, including time step and order adaptive backward differentiation formulas and time step adaptive second-order midpoint rule. In these implicit time stepping schemes, at each time step, a non-linear equation is solved via iterative Newton's method assisted with a linear solver at each non-linear iteration. The convergence of the linear solver is assisted with a preconditioner, such as ILU or block inverse preconditioners [97,98].

We first show a transformation of a 1D periodic protruding touching unit cell as in. We then present examples of micromagnetic simulations for 1D/2D/3D periodic problems. Finally, we show the computational performance of PM-FEM. We do not show results for computing the periodized exchange field since the computational time for the periodic and non-periodic cases of computing the exchange field is nearly the same and this time is significantly below that of the magnetostatic field.

6.3.1 Protruding unit cell

The protruding unit cell is defined such that there is at least one periodic direction where the structure size exceeds the periodic length. In Fig. 6.2, the structure is a parallelogram with a 1D $x-$direction T-P-PBC. The cell size along the $x-$direction is one and a half of the periodic length, e.g., $L_x = 2, D_x = 3$. By utilizing Eq. (6.19) with $\mathbf{r}_0$ set to the geometry center of the unit cell (the black dot in Fig. 6.2(b)). , we shift the green and blue part in Fig. 6.2(b) via Eq. (6.19), and obtain a new unit cell with a regular shape, i.e., a rectangular unit cell with T-NP-PBC (Fig. 6.3(c)). This step is only needed once in the preprocessing stage to map the coordinates, and it has no effect on the rest of the computations.

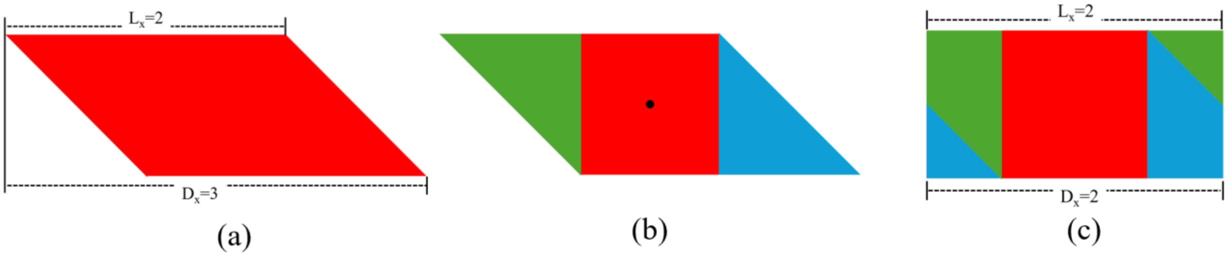


Figure 6.2: Illustration of (a) Protruding unit cell; (b) its geometry center (black dot) and its protruding parts in green and blue; (c) Regular unit cell after shifting the protruding parts.

6.3.2 1D periodicity

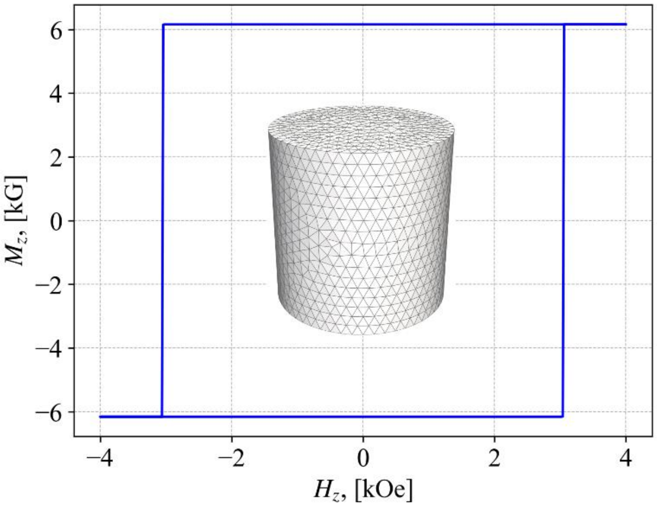


Figure 6.3: Hysteresis loop along $\hat{z}-$ direction of infinite long periodic rod along $\hat{z}-$ direction, the inset is the unit cell. The loop is of square shape and coercive field is around 3050 Oe, close to theoretical value 3085 Oe.

Fig. 6.3 shows a calculated M-H hysteresis loop for an infinitely long cylindrical rod with $A_{ex} = 9.604\times10^{-7}$ erg/cm , $M_s = 490$ emu/cm^3 , $\alpha = 0.5$, and no anisotropy. The corresponding exchange length is $\lambda_{ex} = \sqrt{A_{ex} / 2\pi M_s^2} \approx 8\times10^{-7}$ cm. The prediction is a mostly square hysteresis loop and T-PBC is required especially for the case where the radius of the rod R is close to the normalized length $R = R_0 = \sqrt{2\pi}\lambda_{ex} \approx 2\times10^{-6}\,\mathrm{cm}$ due to the buckling reversal mechanism. We choose the height of the unit cell twice of the radius, i.e., $h = 2R$ (shown as the inset of Fig. 6.3), as indicated in expecting accurate results. We impose a 1D T-NP-PBC along the height ($z-$) direction. The switching is via buckling mode and the coercive field is $H_c == 3085\,\mathrm{Oe}$, which agrees with solutions in literature [99,100].

6.3.3 2D periodicity

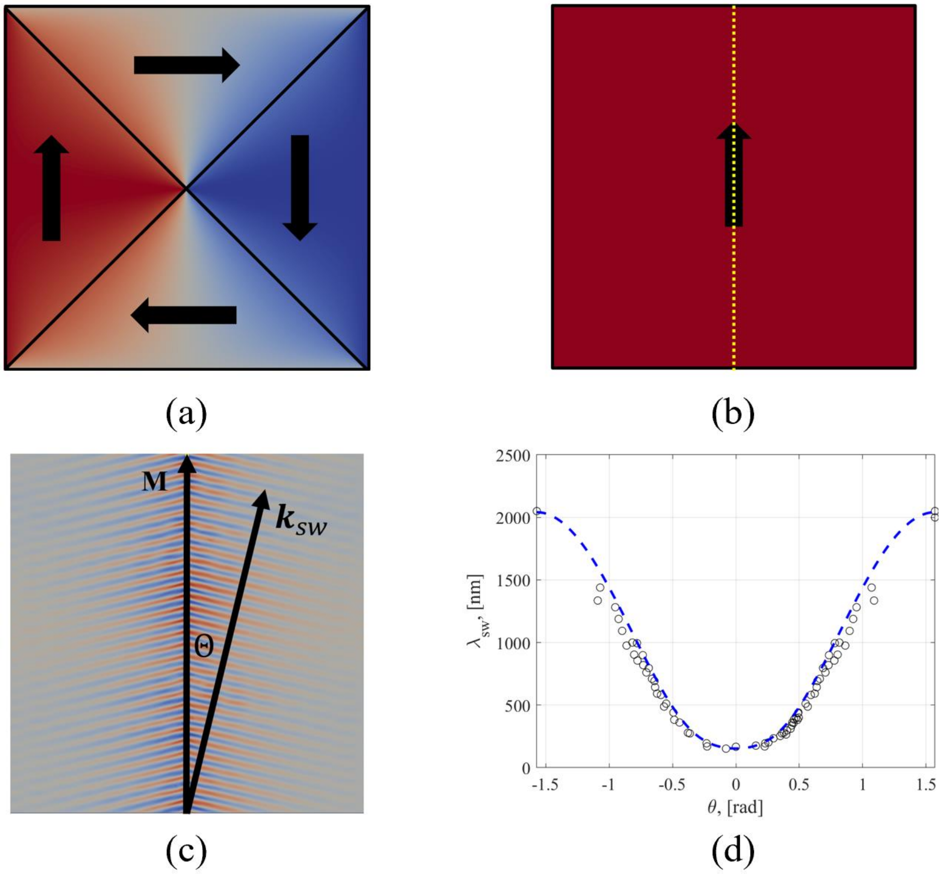
Here, we present results for an infinite Permalloy film, modelled as a 2D T-NP-PBC square unit cell of the side length of $D_x = D_y = 2\times10^{-5}$ cm and thickness $D_z = 2\times10^{-6}$ cm (Fig. 6.4(a)) . The material parameters are $M_s = 637\,\mathrm{emu/cc}, A_{ex} = 1.4\times10^{-6}\,\mathrm{erg/cm}$, zero anisotropy, and $\alpha = 0.02$. We impose a 2D T-PBC in the $\hat{x}$ and $\hat{y}$ directions, i.e., $L_x = L_y = D_x = D_y$. We compare the results for the cases with and without the PBC. Without the PBC, the equilibrium is a vortex state, related to the magnetostatic effects of the edges (Fig. 6.4(a)). With the PBC, the equilibrium state is uniform in the $x-y$ plane (e.g., magnetization aligned along the $y-$direction in Fig. 6.4(b)).

We then use a line source (bright yellow line in Fig. 6.4(b)) in the middle of the film along the $y-$ direction parallel to the equilibrium magnetization. The line source represents a radio frequency (RF) excitation $H(\mathbf{r},t) = H_0 \cos(\omega t - |\mathbf{k}|\, y)$ of width $d = 10^{-6}\,\mathrm{cm} \ll 2\pi/|\mathbf{k}|$. We fix the driving frequency $\omega_0 = 4.85\times 2\pi$ GRad/s and tune wavelength $2\pi/|\mathbf{k}|$. Due to the phase matching, excited propagating spin waves (Fig. 6.4(c)) share the same wavenumber with tilted angle θ between the wave vector $\mathbf{k}_{sw}$ and magnetization $\mathbf{M}$ that satisfy $\mathbf{k}_{sw}\cos\theta = \mathbf{k}$. Therefore, by sweeping the $\mathbf{k}$, we can extract the dispersion relationship of the spin wave for the wavelength

$\lambda_{sw} = 2\pi / |\mathbf{k}_{sw}|$ versus the angle θ. We compare the calculated dispersion relationship with a theoretical result [101]:

$$\omega^2(\mathbf{k}_{sw}) = \gamma_{ex}\omega_M \mathbf{k}_{sw}^2 \times \left[\gamma_{ex}\omega_M \mathbf{k}_{sw}^2 + \omega_M \times \left(1 - \frac{|\mathbf{k}_{sw}| D_z \cos^2\theta}{2} + \omega_M \frac{|\mathbf{k}_{sw}| D_z (2 - |\mathbf{k}_{sw}| D_z) \sin^2\theta}{4\gamma_{ex}\omega_M \mathbf{k}_{sw}^2}\right)\right], \tag{6.20}$$

where $\omega_M = \gamma_{\text{gr}} M_s$, $\gamma_{ex} = \lambda_{ex}^2 = A_{ex} / 2\pi M_s^2$. By fixing $\omega(\mathbf{k}_{sw}) = \omega_0$ in Eq. (6.20) and solving its implicit counterpart with a desired θ, we obtain the dispersion relationship in $\theta \in [-\pi/2, \pi/2]$. The comparison between the numerical (circle marks) and analytical (dashed line) results are shown in Fig. 6.4(d). The results match each other well.



(a)

(b)

(c)

(d)

Figure 6.4: Example of 2D T-NP-PBC case. (a) Equilibrium vortex state without PBC and (b) the uniform state when with 2D T-NP-PBC. The yellow dashed line is the location of the line

excitation. (c) Angles between wave vector of propagating spin wave and magnetization. (d) Simulated wavelength (circle marks) and theoretical prediction (blue dashed line).

6.3.4 3D periodicity

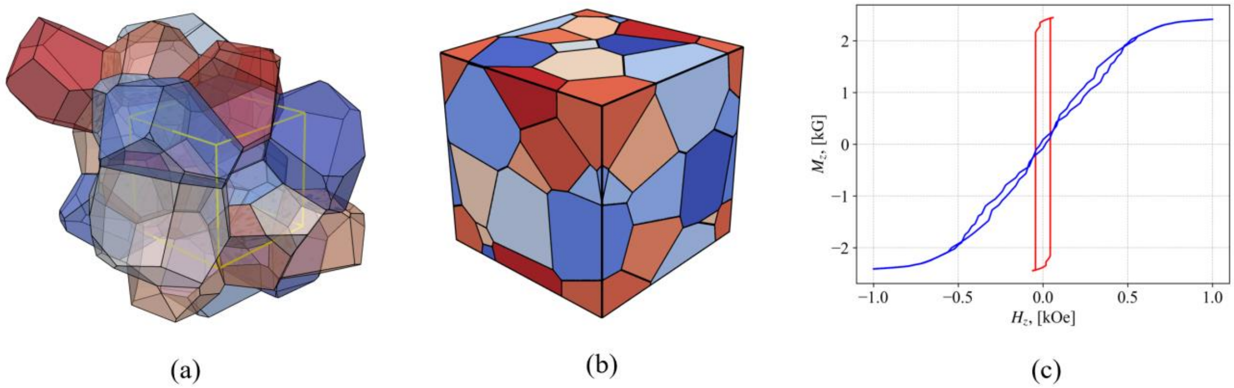


Figure 6.5: Example of 3D T-NP-PBC case. (a) Multi-grain structure (the unit cell is marked as a yellow cube); (b) multi-grain structure folded into the unit cell; (c) hysteresis loop for the periodic multi-grain structure. The unit cell size is $6\mu m$, the average grain size is $2.5\mu m$. The structure was meshed in a tetrahedral mesh with 70 million elements and 12 million nodes.

For a 3D periodicity problem example, we use a complex multi-grain structure representing a granular soft magnetic material. The structure consists of 30 unique grains (Fig. 6.5(b)). The grains are configured to be periodic in terms of the corresponding dimensions. The structure is classified as both touching and protruding (Fig. 6.5(a)). After folding the protruding parts, the resulting unit cell is a cube with 3D PBC. The side length of the unit cell is $Dx = Dy = Dz = 6\,\mu m$. The unit cell has $M_s = 200\,\text{emu/cm}^3$, $A_{ex} = 1\times10^{-6}\,\text{erg/cm}$, and cubic anisotropy of $K_1 = 3\times10^4\,\text{erg/cm}^3$ with a random axis direction per grain. Additionally, the grains are separated by a small (nm-length) randomly variable distance, and they have an interfacial exchange coupling with the surface energy density of $A_{iex} = 0.05\,\text{erg/cm}^2$. The corresponding exchange length is $\lambda_{ex} = \sqrt{A_{ex} / 2\pi M_s^2} \approx 20\,\text{nm}$, and the mesh size is set to 0.5 λ_{ex}. We impose the 3D T-NP-PBC along all directions and set $L_{[x,y,z]}$ to be same as $D_{[x,y,z]}$ to mimic an infinite domain and eliminate undesired magnetostatic field edge effects that are present without PBC. The simulation is to obtain a $M-H$ hysteresis loop. As seen from Fig. 6.5(c) , the non-PBC case has a significantly lower remanence magnetization and higher saturation field due to the magnetic charges at the boundary of the unit cell, which signifies the necessity of applying the PBC condition in soft material analysis.

6.3.5 Performance

To demonstrate the computational performance, Fig. 6.6 shows the computational time of a one-time evaluation of the magnetostatic field for the PM-FEM and its corresponding non-periodic case with the original BAIM-based method. The magnetostatic field evaluation is shown since it is often one of the important computational bottlenecks. To best demonstrate the performance, we choose to compare the 3D T-NP-PBC case that possesses maximal complexity as 1D/2D cases require a lower computational time. The unit cell is a cube of size $D_x = D_y = D_z = 100$ with periodicity settings of $L_x = D_x, L_y = D_y, L_z = D_z$. We mesh the cube with a tetrahedral mesh. By changing the tetrahedron size, we generate models with different numbers of vertices N . We use the PM-FEM method and the original non-PBC method on one core of AMD 5950X CPU and on NVIDIA RTX 3080 Ti GPU with single precision and relative error level of 10^{-3} . The results show that the periodic case may be even faster than the non-periodic case. The performance gain is from the step 2 in which the FFT-based convolution is utilized to calculate the potential on the grid. Thanks to PGF, we avoid the extra zero-padding and mirroring, which leads to a higher performance. The computational time on CPU scales nearly linearly with N . On GPU, for smaller N , the computational time scales less than linearly because the GPUs are underutilized. For larger sizes, with a full GPU utilization, the computational time scaling is also linear with N .

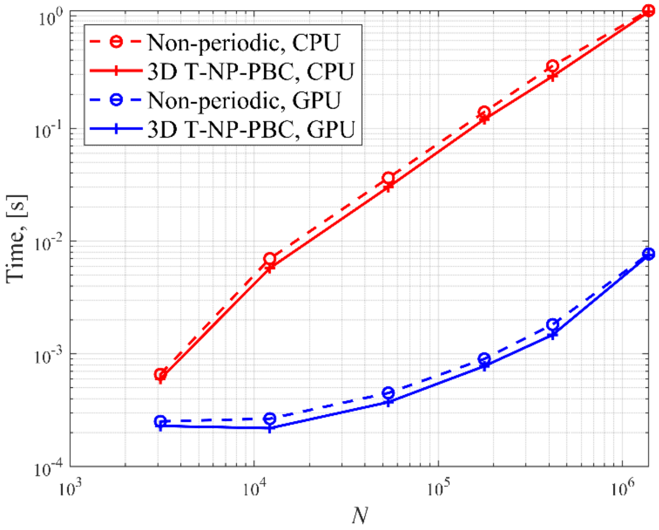


Figure 6.6: Performance comparison of calculation of magnetostatic field between the PM-FEM on 3D T-NP-PBC case and non-periodic case with original method on single-thread CPU and GPU.

Finally, Fig. 6.7 shows the computational time of computing the magnetization dynamics in the 3D PBC case of the granular structure of Fig. 6.5(a), which can be used to calculate the core loss and permeability of soft magnetic materials. The excitation was by a uniformly applied magnetic field of 1.0 Oe at the frequency of 5 MHz. The computational time is given for a 1 ns simulation. The simulations were done on a NVIDIA A100 GPU. For the 1 ns simulation, it took 101 time steps, 112 non-linear iterations, and 888 linear iterations used for the implicit BDF time stepping. The results show a mostly linear computational time scaling with N . For the largest considered case of $N = 11.7\ \mathrm{M}$, the computational time is 216 s, which allows doing a design of experiment for such complex structures in real time. The computational time for a non-periodic case for the same structure was nearly the same. The achievable limit on N is set by the available GPU memory, which is related not only to the periodic code components, such as the exchange and magnetostatic field, but also to other components, such as preconditioners for the linear solver used for implicit time stepping as well as data structure required for outputs.

6.4 Summary

In this chapter we introduced an efficient PM-FEM approach to account for periodicity in micromagnetic simulations. PM-FEM is applicable to all 1D/2D/3D periodic problems within a uniform framework. PM-FEM implementation is based on the non-periodic FastMag micromagnetic framework, which is updated in several aspects to include PBC. The main modifications are in handling the exchange and magnetostatic fields. Handling the exchange field is based on modifying the construction of the sparse matrix representing the Laplace operator to include elements corresponding the periodic extension in the case of touching periodic unit cells. Handling the magnetostatic field includes the local operators, such as the gradient, divergence, and surface magnetic charges as well as the long-range superposition operator. The local operators for the magnetostatic field computation are handled similar to the Laplace operator for the exchange field. The long-range superposition operator involved a rapidly convergent approach for computing the PGF as well as an updated FFT-based BAIM approach to compute the PSP in $O(N \log N)$ computational cost. The modifications allow for the BAIM extension are based on the identification of the local error correction range that is based on the periodic extensions of the sources from the proper sides of the periodic unit cell.

Numerical examples demonstrate the efficiency and generality of PM-FEM. It is shown that FM-FEM can handle any 1D, 2D, and 3D periodicities with the speed that is comparable or even higher than that for similar non-periodic problems. The results also show that the updated FastMag simulator can handle large complex meshes of tens of millions of elements. PM-FEM can be used for many micromagnetic problems, including those requiring large meshes. Examples of using PM-FEM with FastMag are the study of the magnetization dynamics in periodic structures, mimicking infinite domains, such as wires, films, and bulks, and the study of spin wave propagation.

Acknowledgements

Chapter 6, in full, is a reprint of the material as it appears in F. Ai, J. Duan and V. Lomakin, “Periodic micromagnetic finite element method”, which is currently under review in J. Magn. Magn. Mater. The dissertation author was the primary researcher and author of this paper.

Chapter 7 Periodic phase diagrams in micromagnetics with an eigenvalue solver

From Chapter 5, we introduced the way to solve the PSP in a neutral system when no phase shift / free-space wave propagation. Furthermore, in Chapter 6 we propose a systematic method to solve the LLG equation in time domain for periodic system without phase shift. In this chapter, we propose an approach to solve eigenvalue problems using the linearized LLG equation with periodic boundary conditions in the FEM framework. We consider general 3D periodic problems possibly with 1D, 2D, and 3D periodicities (Fig. 7.1). A key aspect of our methodology is the precise computation of the periodic fields, which ensures compatibility with the micromagnetic framework while preserving the periodicity of the system. By employing this approach, we aim to construct periodic phase diagrams that capture the stability and dynamic (spin wave propagation) properties of micromagnetic systems under various conditions.

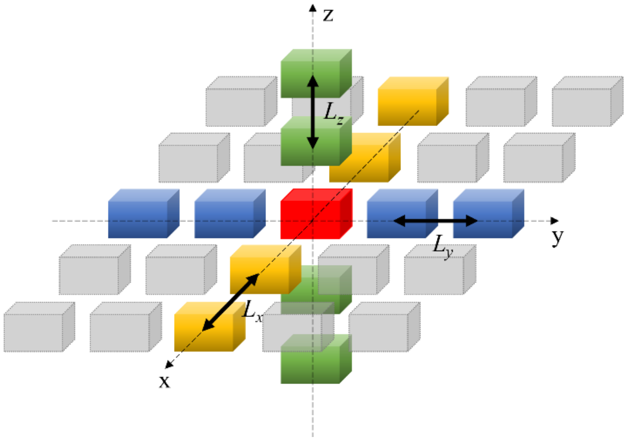


Figure 7.1: Illustration for a periodic problem consisting along $\hat{x}$ (yellow), $\hat{y}$ (blue) and $\hat{z}$ (green) direction with 0^{th} unit cell (red) and its periodic images (grey).

We validate our method through applications to well-studied periodic micromagnetic systems, demonstrating its efficiency and accuracy in calculating the periodic problem with phase shift. We then demonstrate the power of the approach by calculating dispersion diagrams of 1D and 2D periodic problems. This work contributes a new computational framework for analyzing periodic micromagnetic structures, offering potential insights for researchers in dynamic magnetism, spintronics, and materials science.

7.1 Problem formulation

We consider a 3D domain (Fig. 7.1) with the a magnetic structure of an arbitrary shape that is infinitely periodic in 1D, 2D, or 3D directions with periodicities of L_x, L_y and L_z, respectively. We develop a formulation that allows calculating the dispersion diagrams for the periodically modulated magnetization with the same periodicities as the domain.

This section presets the formulation of the periodic linearized LLG equation [12,102] and field with phase-shifted periodic boundary condition (PBC), leading to the ability to set up a periodic eigenvalue problem.

7.1.1 Periodic linearized LLG equation

We have introduced the LLG equation and its total effective field $\mathbf{H}_{\text{eff}}$ in Eq. (2.1) and Eq. (2.2). Under the assumption of a weak excitation, the magnetization can be considered as a weak perturbation $\mathbf{v}$ around the equilibrium state $\mathbf{M}_0$, i.e., $\mathbf{M} = \mathbf{M}_0 + \mathbf{v}$. The effective field also can be written similarly: $\mathbf{H}_{\text{eff}} = \mathbf{H}_{\text{eff},0} + \mathbf{h}$, where $\mathbf{H}_{\text{eff},0} = \mathbf{H}_{\text{eff}}(\mathbf{M}_0)$ is the effective field for the magnetization equilibrium state and $\mathbf{h} = \mathbf{H}_{\text{eff}}(\mathbf{v})$ is the corresponding field perturbation. The equilibrium state satisfies Brown's condition $\mathbf{M}_0 \times \mathbf{H}_0 = 0$ subject to a PBC with no periodic phase shift as indicated in Eq. (6.1) for the 1D, 2D, or 3D periodicity, respectively. The equilibrium effective field $\mathbf{H}_{\text{eff},0}$ satisfies the same PBC with no phase shift as $\mathbf{M}_0$. The equilibrium state $\mathbf{M}_0$ is found by satisfying the Brown condition by energy minimization or solving the dynamic LLG equation (Eq. (2.1)) with high damping until no significant time variations are obtained.

The magnetization perturbation $\mathbf{v}$ satisfies the following phase-shifted PBC:

$$\begin{aligned}&\mathbf{v}(\mathbf{r}+L_x\mathbf{x})=e^{-jk_{x0}L_x}\mathbf{v}(\mathbf{r}),\\&\mathbf{v}(\mathbf{r}+L_x\mathbf{x}+L_y\mathbf{y})=e^{-j\left(k_{x0}L_x+k_{y0}L_y\right)}\mathbf{v}(\mathbf{r}),\\&\mathbf{v}(\mathbf{r}+L_x\mathbf{x}+L_y\mathbf{y}+L_z\hat{\mathbf{z}})=e^{-j\left(k_{x0}L_x+k_{y0}L_y+k_{z0}L_z\right)}\mathbf{v}(\mathbf{r}),\end{aligned}\tag{7.1}$$

for the 1D, 2D, or 3D periodicity, respectively. Here, k_{x0},k_{y0}, and k_{z0} are periodic phase shift wave numbers in the x,y and z direction, respectively. These phase shift wave numbers can be combined into a phase shift wave vector $\mathbf{k}_0=\mathbf{x}k_{x0}+\mathbf{y}k_{y0}+\hat{\mathbf{z}}k_{z0}$, which is defined depending on the dimensionality of the periodicity. The perturbation field $\mathbf{h}$ satisfies the same PBC as $\mathbf{v}$, and it is linear in $\mathbf{v}$, i.e., it can be written as:

$$\mathbf{h}=\mathcal{C}(\mathbf{k}_0)\mathbf{v},\tag{7.2}$$

where $\mathcal{C}(\mathbf{k}_0)$ is a linear operator that includes the linear operators corresponding to the effective field components. This operator depends on the wave vector $\mathbf{k}_0$ 。

Keeping only terms that are linear in the small perturbation $\mathbf{v}$ in the LLG equation Eq. (2.1), we obtain a linearized LLG equation

$$\begin{aligned}\frac{d\mathbf{v}}{dt}&=-\frac{\gamma}{1+\alpha^2}\left(\mathbf{M}_0\times\mathbf{h}-\mathbf{H}_{\mathrm{eff},0}\times\mathbf{v}\right)-\frac{\gamma\alpha}{(1+\alpha^2)M_s}\mathbf{M}_0\times\left(\mathbf{M}_0\times\mathbf{h}-\mathbf{H}_{\mathrm{eff},0}\times\mathbf{v}\right)\\&=\mathcal{A}(\mathbf{k}_0)\mathbf{v},\end{aligned}\tag{7.3}$$

where $\mathcal{A}$ is a linear operator representing the right hand side of the equation and this operator accounts for the fact that the field perturbation $\mathbf{h}$ is linear with respect to $\mathbf{v}$ via Eq. (7.2). The operator $\mathcal{A}$ is a function of the phases shift wave vector $\mathbf{k}_0$.

For finding the dispersion diagrams, we define $\mathbf{v}$ in the form of $\mathbf{v}(\mathbf{r},t)=e^{j\omega t}\varphi(\mathbf{r})$ and assume no external excitation, i.e., $\mathbf{H}_{\mathrm{ap}}=0$, which allows writing the linearized equation Eq. (7.3) as an eigenvalue problem:

$$j\omega\varphi=\mathcal{A}(\mathbf{k}_0)\varphi.\tag{7.4}$$

Here, ω is the eigen frequency corresponding to the eigenstate $\varphi(\mathbf{r})$ that satisfies the PBC as in Eq. (7.1). Since the operator $\mathcal{A}$ is a function $\mathbf{k}_0$, solving this eigenvalue problem result in calculating a dispersion diagram, i.e., the dependence of the eigen frequency of the wave vector $\mathbf{k}_0$. This dispersion diagram can be considered from two points of view. One can obtain a dependence of generally complex ω versus $\mathbf{k}_0$ by solving the explicit eigenvalue problem of Eq. (7.4) for a range of given real $\mathbf{k}_0$. Alternatively, one can obtain generally complex $\mathbf{k}_0$ for a set of given real ω. The latter approach may require solving an implicit eigenvalue problem because $\mathbf{k}_0$ appears in the operator $\mathcal{A}$ implicitly. While being more complex problem to solve, this approach allows directly calculating not only the real but also imaginary parts of the wave numbers, thus providing the spin wave propagation length.

7.1.2 Field under periodic boundary condition with phase shift

Only the magnetostatic and exchange fields need special care in terms of PBCs since they come from non-localized interactions. For the exchange field, changes are needed when touching periodic boundary condition (T-PBC) is present, namely the object size is equal to the periodic length.

The exchange fields corresponding to the dynamic perturbation is given by

$$\mathbf{h}_{\mathrm{ex}}(\mathbf{r}) = \frac{2A_{ex}}{M_s^2(\mathbf{r})}\nabla^2\mathbf{v}(\mathbf{r}), \tag{7.5}$$

where ∇^2 is the Laplacian operator and A_{ex} is the exchange constant. In typical FEM in micromagnetics, the structure is meshed into a mesh, that often is based on tetrahedral tessellation. The solution is obtained as the magnetization states at the vertices of the mesh. Inside the mesh elements, the magnetization is interpolated via polynomials, which often are chose as linear. The Laplacian operator is implemented as a sparse matrix with the matrix band determined by the connectivity of the mesh vertices to the surrounding vertices connected via common elements. When the computational domain is smaller than the periodicity, there is no need to modify the conventional sparse matrix representation. When the computational domain extends through the periodic boundaries, the PBC of Eq. (7.1) needs to be accounted for by properly updating the sparse matrix. The exchange field corresponding to the equilibrium state $\mathbf{H}_{\mathrm{ex},0}$ is given by the right hand side of Eq. (2.9), where $\mathbf{v}$ is replaces with $\mathbf{M}_0$ and the PBC of Eq. (6.1) is used.

The perturbation magnetostatic field $\mathbf{h}_{\mathrm{ms}}$ is due to long-range interactions, and it can be calculated either by solving the Poisson equation or by evaluating the superposition integrals. We evaluate the magnetostatic field efficiently using superposition integrals that is introduced in chapter 6.2.2.

7.2 Results

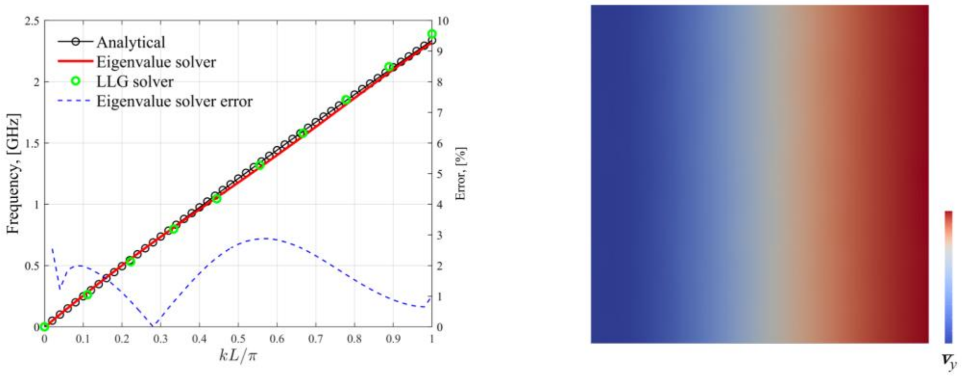
Here we demonstrate results obtained using the periodic eigenvalue LLG solver. The structure is meshed via a tetrahedral mesh and linear nodal elements are used. The dispersion diagrams are calculated by solving the eigenvalue problem for the eigen frequencies ω using the periodic eigenvalue LLG solver for a set of prescribed wave numbers. The results include a validation example and examples of calculating the dispersion diagrams for 3D problems with 1D and 2D periodicities.

We first validate the presented approach by calculating the dispersion relation of magnetostatic backward volume wave (MSBVW) in an infinitely large permalloy film.

7.2.1 MSBVW dispersion relationship

As we introduced in chapter 2.3.2.2, the theoretical dispersion relationship of the MSBVW can be found in Eq. (2.15). In the numerical solution, we set the unit cell of size $200\times200\times20$ nm, with the mesh edge length of 8 nm. The material parameters are $M_s = 637\,\mathrm{emu/cc}$,

$A_{ex} = 1.4\times10^{-6}$ erg/cm , no anisotropy, and $\alpha = 0.02$. We impose a 2D PBC in the x and y directions, i.e., $L_x = L_y = 200$ nm. The equilibrium magnetization is aligned along the $x-$ direction. By sweeping $k = k_{x0}$ and keeping $k_{y0} = 0$, the corresponding eigen-frequency related to the MSBVW of wave vector k_x is calculated. For further comparison, we also calculate the dispersion relationship using the time-domain LLG solver. All these results are present in Fig. 7.2 and the eigen state of the MSBVW with the wave vector $k = \pi / L_x$ is also plotted. We observe all methods agree well with each other validating the presented solver. We note that running the time domain LLG equation is much slower and has various issues, such as a possible non-linearity in the behavior.

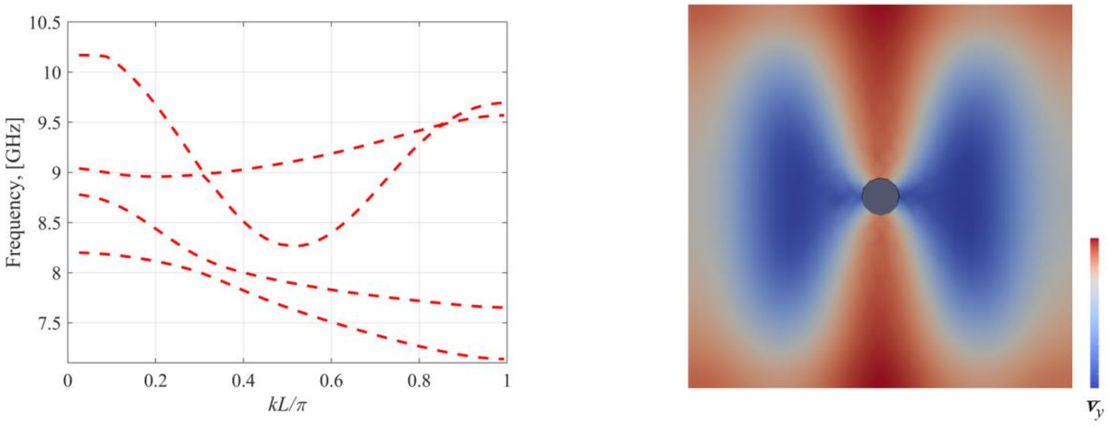


(a) (b)

Figure 7.2: The left figure is the phase diagram (dispersion relationship) calculated from theoretical equation (black circles), LLG time-domain solver (green circles), eigenvalue solver (red line) and its relative error with respect to the theoretical values. The right figure is the y component of the magnetization perturbation $\mathbf{v}$ from the eigenstate of $k_x = \pi / L_x$.

7.2.2 1D periodicity with a hole

We then show a 1D periodic phase diagram for the same film with a periodic array of holes in the x direction with the same periodicity of 200 nm, which is of a width of 200 nm in the $y-$ direction. The equilibrium state, which is calculated via the periodic LLG time-domain solver, is found to be slightly different than that for the case of the uniform film. The eigen state of the lowest branch with $k_{x0} = \pi / L_x$ and the periodic phase diagram of the first 4 branches with the lowest energy are shown in Fig. 7.3. The eigen state exhibits a butterfly feature which is very different from that in Fig. 7.2.



(a) (b)

Figure 7.3: The left figure is the 1D phase diagram of the film with a hole in the middle from the periodic LLG eigen value solver (red dashed line). The right figure is the y-component of the magnetization perturbation $\mathbf{v}$ from the eigen state of $k_{x0} = \pi / L_x$.

7.2.3 2D periodicity with a hole

Finally, we calculate a periodic phase diagram of the above example with both x and y periodicities. The eigen state of the lowest branch with $k_{x0} = \pi / L_x$ and the periodic phase diagram of the first 3 branches with the lowest energy are shown in Fig. 7.4. The eigenstate here is very different from that in Fig. 7.2.

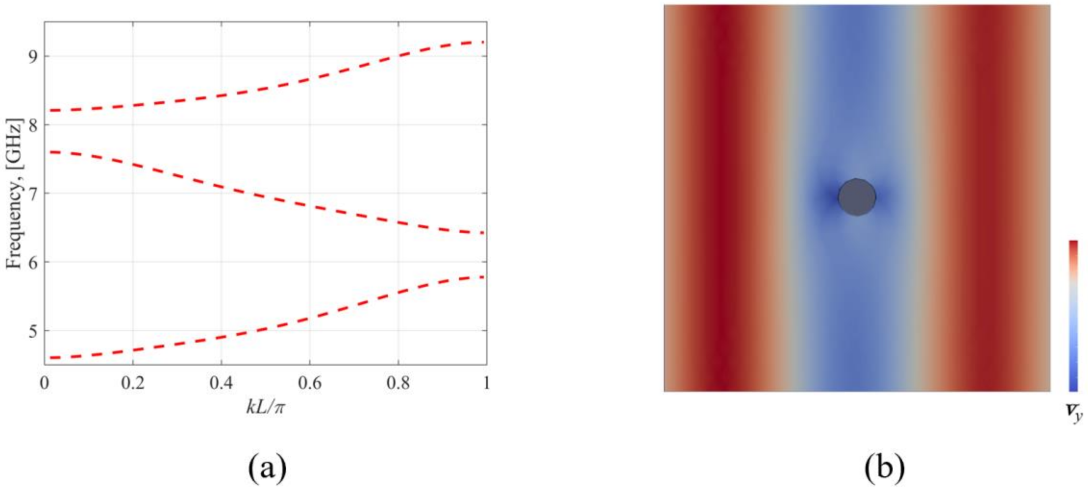


Figure 7.4: The left figure is the 2D phase diagram of the film with a hole in the middle from the periodic LLG eigenvalue solver (red dashed line). The right figure is the y component of the magnetization perturbation $\mathbf{v}$ from the eigen state of $k_{x0} = \pi / L_x$.

7.3 Conclusion

In this chapter We presented a computational framework for calculating periodic phase diagrams in micromagnetic systems by solving a linearized LLG equation with an eigenvalue solver. A critical aspect of the presented approach is the calculation of the periodic field, which ensures the compatibility of PBCs with the micromagnetic formalism. By addressing the challenges associated with periodic systems, the presented approach offers an efficient and accurate means to analyze the stability and dynamic properties of micromagnetic structures.

The results demonstrate the validity and generality of the approach in capturing the key features of periodic micromagnetic systems, including resonance behaviors and phase behavior under varying conditions. Case studies illustrate the utility of the proposed method for investigating nanostructured materials and patterned magnetic systems, providing valuable insights into their dynamic properties.

This work not only advances the computational tools available for micromagnetic analysis but also lays the foundation for further exploration of periodic systems in nano-magnetism and materials science. Future studies could extend this methodology to include nonlinear dynamics, thermal effects, and more complex geometries, broadening its applicability to a wider range of micromagnetic phenomena.

Acknowledgements

Chapter 7, in full, is a reprint of the material as it appears in F. Ai, Z. Lin, J. Duan and V. Lomakin, “Periodic phase diagrams in micromagnetics with an eigenvalue solver”, which is currently under review in IEEE Trans. Magn. The dissertation author was the primary researcher and author of this paper.

Chapter 8 Conclusion

In this dissertation, we have explored a series of topics including high-performance algorithms in CXDI, a novel design of STNO based device that can achieve global synchronization, fast calculation of PSP in the periodic structures, and a specific application in periodic micromagnetic problems. We further present methods to solve the LLG equation in the time domain and frequency domains.

Through the chapters presented, we made several contributions. In the context of micromagnetics, we developed techniques to handle large-scale simulations governed by the Landau–Lifshitz–Gilbert (LLG) equation. We implemented these techniques in an efficient micromagnetic framework. By introducing interpolation-based approaches and addressing periodic boundary conditions, we enabled efficient FEM-based simulations for time- and frequency-domain problems. Furthermore, the integration of our methods enabled the study of novel magnetic structures, such as arrays of spin-transfer torque nano-oscillators (STNOs) and offered insights into their synchronization behavior. In the domain of CXDI, we achieved real-time processing capabilities by proposing a high-performance algorithm tailored for experimental data, thereby broadening the impact of HPC techniques in areas where massive datasets must be processed.

Taken together, our work demonstrates how methodical algorithmic foundations, efficient use of computing hardware, and domain-specific optimizations can unlock major performance gains and deepen our understanding of physical phenomena. The techniques and frameworks developed in this dissertation contribute to the ongoing evolution of hardware, not only accelerating current scientific and engineering tasks but also empowering researchers to address increasingly complex, data-intensive challenges. As hardware technology and computational models continue to advance, the methodologies presented here offer a foundation on which future work can build, driving forward progress in fields ranging from micromagnetic simulations to high-throughput experimental analyses, and shaping the role of HPC in solving the grand challenges of tomorrow’s computational science.

REFERENCES

[1] A. Friedman, Micromagnetics, 182 (1992).

[2] L. LANDAU and E. LIFSHITZ, On the theory of the dispersion of magnetic permeability in ferromagnetic bodies, Perspectives in Theoretical Physics 51 (1992).

[3] T. L. Gilbert, A phenomenological theory of damping in ferromagnetic materials, IEEE Trans Magn **40**, 3443 (2004).

[4] Handbook of Magnetism and Advanced Magnetic Materials, Handbook of Magnetism and Advanced Magnetic Materials (2007).

[5] B. Livshitz, A. Boag, H. N. Bertram, and V. Lomakin, Nonuniform grid algorithm for fast calculation of magnetostatic interactions in micromagnetics, J Appl Phys **105**, 07D541 (2009).

[6] J. A. Rackers and J. W. Ponder, Classical Pauli repulsion: An anisotropic, atomic multipole model, J Chem Phys **150**, 084104 (2019).

[7] M. E. Fisher, Magnetism in One-Dimensional Systems—The Heisenberg Model for Infinite Spin, Am J Phys **32**, 343 (1964).

[8] E. B. Poulsen, A. R. Insinga, and R. Bjørk, Direct exchange calculation for unstructured micromagnetic meshes, J Magn Magn Mater **551**, 169093 (2022).

[9] G. S. Abo, Y. K. Hong, J. Park, J. Lee, W. Lee, and B. C. Choi, Definition of magnetic exchange length, IEEE Trans Magn **49**, 4937 (2013).

[10] T. Kim, I. H. Cha, Y. J. Kim, G. W. Kim, A. Stashkevich, Y. Roussigné, M. Belmeguenai, S. M. Chérif, A. S. Samardak, and Y. K. Kim, Ruderman–Kittel–Kasuya–Yosida-type interfacial Dzyaloshinskii–Moriya interaction in heavy metal/ferromagnet heterostructures, Nature Communications 2021 12:1 **12**, 1 (2021).

[11] D. Azuma, Magnetic materials, Wide Bandgap Power Semiconductor Packaging: Materials, Components, and Reliability 97 (2018).

[12] Z. Lin and V. Lomakin, Linearized frequency domain Landau-Lifshitz-Gilbert equation formulation, AIP Adv **13**, (2023).

[13] S. Zhang and Z. Li, Roles of Nonequilibrium Conduction Electrons on the Magnetization Dynamics of Ferromagnets, Phys Rev Lett **93**, 127204 (2004).

[14] C. Ragusa, M. D'Aquino, C. Serpico, B. Xie, M. Repetto, G. Bertotti, and D. Ansalone, Full micromagnetic numerical simulations of thermal fluctuations, IEEE Trans Magn **45**, 3919 (2009).

[15] W. F. Brown, Magnetoelastic Interactions, **9**, (1966).

[16] A. Prabhakar and D. D. Stancil, Spin waves: Theory and applications, Spin Waves: Theory and Applications 1 (2009).

[17] S. O. Demokritov and A. N. Slavin, Spin Waves, Handbook of Magnetism and Magnetic Materials: Volume 1,2 **1**, 281 (2021).

[18] A. Slavin and V. Tiberkevich, Nonlinear auto-oscillator theory of microwave generation by spin-polarized current, IEEE Trans Magn **45**, 1875 (2009).

[19] J. Miao, R. L. Sandberg, and C. Song, *Coherent X-Ray Diffraction Imaging*, IEEE Journal on Selected Topics in Quantum Electronics.

[20] T. Latychevskaia, Iterative phase retrieval in coherent diffractive imaging: practical issues, (2018).

[21] K. S. Raines, S. Salha, R. L. Sandberg, H. Jiang, J. A. Rodríguez, B. P. Fahimian, H. C. Kapteyn, J. Du, and J. Miao, Three-dimensional structure determination from a single view, Nature **463**, 214 (2010).

[22] M. A. Pfeifer, G. J. Williams, I. A. Vartanyants, R. Harder, and I. K. Robinson, Three-dimensional mapping of a deformation field inside a nanocrystal, (2006).

[23] Y. Nishino, Y. Takahashi, N. Imamoto, T. Ishikawa, and K. Maeshima, Three-Dimensional Visualization of a Human Chromosome Using Coherent X-Ray Diffraction, (n.d.).

[24] J. A. Rodriguez et al., Three-dimensional coherent X-ray diffractive imaging of whole frozen-hydrated cells, **2**, 575 (2015).

[25] C.-C. Chen, C.-H. Lu, D. Chien, J. Miao, and T. K. Lee, Three-dimensional image reconstruction of radiation-sensitive samples with x-ray diffraction microscopy, Phys Rev B **84**, 24112 (2011).

[26] H. Jiang et al., Three-Dimensional Coherent X-Ray Diffraction Imaging of Molten Iron in Mantle Olivine at Nanoscale Resolution, Phys Rev Lett **110**, 205501 (2013).

[27] T. Li et al., Three-Dimensional Quantitative Coherent Diffraction Imaging of Staphylococcus aureus Treated with Peptide-Mineralized Au-Cluster Probes, Anal Chem **94**, 13136 (2022).

[28] J. Miao, C.-C. Chen, C. Song, Y. Nishino, Y. Kohmura, T. Ishikawa, D. Ramunno-Johnson, T.-K. Lee, and S. H. Risbud, Three-Dimensional GaN-Ga 2 O 3 Core Shell Structure Revealed by X-Ray Diffraction Microscopy, (2006).

[29] J. Miao, T. Ishikawa, I. K. Robinson, and M. M. Murnane, Beyond crystallography: Diffractive imaging using coherent X-ray light sources, Science (1979) **348**, 530 (2015).

[30] T. Hatsui and H. Graafsma, X-ray imaging detectors for synchrotron and XFEL sources, Urn:Issn:2052-2525 **2**, 371 (2015).

[31] V. Elser, Phase Retrieval by Iterated Projections, 2003.

[32] Y. Shechtman, Y. C. Eldar, O. Cohen, H. N. Chapman, J. Miao, and M. Segev, Phase Retrieval with Application to Optical Imaging: A contemporary overview; Phase Retrieval with Application to Optical Imaging: A contemporary overview, IEEE Signal Process Mag **32**, (2015).

[33] J. Miao, T. Ishikawa, B. Johnson, E. H. Anderson, B. Lai, and K. O. Hodgson, High Resolution 3D X-Ray Diffraction Microscopy, (n.d.).

[34] J. R. Fienup, Phase retrieval algorithms: a comparison, Appl Opt **21**, 2758 (1982).

[35] H. N. Chapman et al., High-Resolution Ab Initio Three-Dimensional x-Ray Diffraction Microscopy, 2006.

[36] M. Van Heel and M. Schatz, Fourier shell correlation threshold criteria, J Struct Biol **151**, 250 (2005).

[37] A. N. Slavin and P. Kabos, Approximate theory of microwave generation in a current-driven magnetic nanocontact magnetized in an arbitrary direction, IEEE Trans Magn **41**, 1264 (2005).

[38] A. Litvinenko, A. Sidi El Valli, V. Iurchuk, S. Louis, V. Tyberkevych, B. Dieny, A. N. Slavin, and U. Ebels, Ultrafast GHz-Range Swept-Tuned Spectrum Analyzer with 20 ns Temporal Resolution Based on a Spin-Torque Nano-Oscillator with a Uniformly Magnetized “Free” Layer, Nano Lett **22**, 1874 (2022).

[39] H. S. Choi et al., Spin nano–oscillator–based wireless communication, Sci Rep **4**, 5486 (2014).

[40] V. H. González, A. Litvinenko, A. Kumar, R. Khymyn, and J. Åkerman, Spintronic devices as next-generation computation accelerators, Curr Opin Solid State Mater Sci **31**, 101173 (2024).

[41] H. Farkhani, T. Bohnert, M. Tarequzzaman, D. Costa, A. Jenkins, R. Ferreira, and F. Moradi, Spin-torque-nano-oscillator based neuromorphic computing assisted by laser, Proceedings - 2019 14th IEEE International Conference on Design and Technology of Integrated Systems In Nanoscale Era, DTIS 2019 (2019).

[42] T. Böhnert, Y. Rezaeiyan, M. S. Claro, L. Benetti, A. S. Jenkins, H. Farkhani, F. Moradi, and R. Ferreira, Weighted spin torque nano-oscillator system for neuromorphic computing, Communications Engineering 2023 2:1 **2**, 1 (2023).

[43] T. Kanao, H. Suto, K. Mizushima, H. Goto, T. Tanamoto, and T. Nagasawa, Reservoir computing on spin-torque oscillator array, Phys Rev Appl **12**, 024052 (2019).

[44] M. romera et al., Vowel recognition with four coupled spin-torque nano-oscillators, Nature (2018).

[45] A. N. Slavin and V. S. Tiberkevich, Theory of mutual phase locking of spin-torque nanosized oscillators, Phys Rev B Condens Matter Mater Phys **74**, (2006).

[46] T. Kendziorczyk, S. O. Demokritov, and T. Kuhn, Spin-wave-mediated mutual synchronization of spin-torque nano-oscillators: A micromagnetic study of multistable phase locking, Phys Rev B Condens Matter Mater Phys **90**, 54414 (2014).

[47] A. A. Awad, P. Dürrenfeld, A. Houshang, M. Dvornik, E. Iacocca, R. K. Dumas, and J. Åkerman, Long-range mutual synchronization of spin Hall nano-oscillators, Nat Phys **13**, 292 (2017).

[48] D. Nikitin, C. Canudas-De-Wit, P. Frasca, and U. Ebels, Synchronization of Spin-Torque Oscillators via Continuation Method, (n.d.).

[49] M. Zahedinejad, A. A. Awad, S. Muralidhar, R. Khymyn, H. Fulara, H. Mazraati, M. Dvornik, and J. Åkerman, Two-dimensional mutually synchronized spin Hall nano-oscillator arrays for neuromorphic computing, Nat Nanotechnol **15**, 47 (2020).

[50] R. J. Mailloux, *Phased Array Antenna Handbook* (Artech house, 2017).

[51] D. Kalkstein and P. Soven, A Green's function theory of surface states, Surf Sci **26**, 85 (1971).

[52] Y. K. Sirenko and S. Strom, Modern theory of gratings, Resonant Scattering: Analysis Techniques and Phenomena (2010).

[53] S. T. Peng, T. Tamir, and H. L. Bertoni, Theory of periodic dielect waveguides, IEEE Trans Microw Theory Tech **23**, 123 (1975).

[54] L. Greengard and V. Rokhlin, A fast algorithm for particle simulations, J Comput Phys **73**, 325 (1987).

[55] A. D. Baczewski, D. L. Dault, and B. Shanker, Accelerated Cartesian Expansions for the Rapid Solution of Periodic Multiscale Problems, IEEE Trans Antennas Propag **60**, 4281 (2012).

[56] W. Hackbusch and B. N. Khoromskij, A sparse H-matrix arithmetic. Part II: application to multi-dimensional problems, Computing **64**, 21 (2000).

[57] A. Boag, E. Michielssen, and A. Brandt, Nonuniform polar grid algorithm for fast field evaluation, IEEE Antennas Wirel Propag Lett **1**, 142 (2002).

[58] S. Li, B. Livshitz, and V. Lomakin, Fast evaluation of Helmholtz potential on graphics processing units (GPUs), J Comput Phys **229**, 8463 (2010).

[59] J. Meng, A. Boag, V. Lomakin, and E. Michielssen, A multilevel Cartesian non-uniform grid time domain algorithm, J Comput Phys **229**, 8430 (2010).

[60] C. Liu, K. Aygün, and A. E. Yılmaz, A parallel FFT-accelerated layered-medium integral-equation solver for electronic packages, International Journal of Numerical Modelling: Electronic Networks, Devices and Fields **33**, e2684 (2020).

[61] E. Bleszynski, M. Bleszynski, and T. Jaroszewicz, AIM: Adaptive integral method for solving large-scale electromagnetic scattering and radiation problems, Radio Sci **31**, 1225 (1996).

[62] J. R. Phillips and J. K. White, A precorrected-FFT method for electrostatic analysis of complicated 3-D structures, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **16**, 1059 (1997).

[63] R. Pei, T. Askham, L. Greengard, and S. Jiang, A fast method for imposing periodic boundary conditions on arbitrarily-shaped lattices in two dimensions, J Comput Phys **474**, 111792 (2023).

[64] D. S. Shamshirgar, J. Bagge, and A.-K. Tornberg, Fast Ewald summation for electrostatic potentials with arbitrary periodicity, J Chem Phys **154**, 164109 (2021).

[65] L. af Klinteberg, D. S. Shamshirgar, and A.-K. Tornberg, Fast Ewald summation for free-space Stokes potentials, Res Math Sci **4**, 1 (2017).

[66] J. Bagge and A.-K. Tornberg, Fast Ewald summation for Stokes flow with arbitrary periodicity, J Comput Phys **493**, 112473 (2023).

[67] F. Capolino, D. R. Wilton, and W. A. Johnson, Efficient computation of the 3D Green's function for the Helmholtz operator for a linear array of point sources using the Ewald method, J Comput Phys **223**, 250 (2007).

[68] S. Li, D. A. Van Orden, and V. Lomakin, Fast Periodic Interpolation Method for Periodic Unit Cell Problems, IEEE Trans Antennas Propag **58**, 4005 (2010).

[69] D. Van Orden and V. Lomakin, Rapidly Convergent Representations for Periodic Green's Functions of a Linear Array in Layered Media, IEEE Trans Antennas Propag **60**, 870 (2012).

[70] Z. Hu, Infinite Boundary Terms of Ewald Sums and Pairwise Interactions for Electrostatics in Bulk and at Interfaces, J Chem Theory Comput **10**, 5254 (2014).

[71] S. L. Marshall, A periodic Green function for calculation of coloumbic lattice potentials, Journal of Physics: Condensed Matter **12**, 4575 (2000).

[72] K. M. Lebecki, M. J. Donahue, and M. W. Gutowski, Periodic boundary conditions for demagnetization interactions in micromagnetic simulations, J Phys D Appl Phys **41**, 175005 (2008).

[73] A. and H. P. and A. C. and S. D. Bruckner Florian and Ducevic, Strayfield calculation for micromagnetic simulations using true periodic boundary conditions, Sci Rep **11**, 9202 (2021).

[74] R. Chang, S. Li, M. V Lubarda, B. Livshitz, and V. Lomakin, FastMag: Fast micromagnetic simulator for complex magnetic structures (invited), J Appl Phys **109**, 07D358 (2011).

[75] L. Barnett Alex and Greengard, A new integral representation for quasi-periodic scattering problems in two dimensions, BIT Numer Math **51**, 67 (2011).

[76] V. Lomakin and E. Michielssen, Enhanced transmission through metallic plates perforated by arrays of subwavelength holes and sandwiched between dielectric slabs, Phys. Rev. B **71**, 235117 (2005).

[77] V. Lomakin and E. Michielssen, Beam Transmission Through Periodic Subwavelength Hole Structures, IEEE Trans Antennas Propag **55**, 1564 (2007).

[78] A. Hessel and A. A. Oliner, A New Theory of Wood's Anomalies on Optical Gratings, Appl. Opt. **4**, 1275 (1965).

[79] V. Lomakin and E. Michielssen, Transmission of transient plane waves through perfect electrically conducting plates perforated by periodic arrays of subwavelength holes, IEEE Trans Antennas Propag **54**, 970 (2006).

[80] A. Bagnérés-Viallix and P. Baras, On a method to calculate the demagnetizing field in a micromagnetic structure, J Appl Phys **69**, 4599 (1991).

[81] J. Stöhr and H. C. Siegmann, Magnetism, Solid-State Sciences. Springer, Berlin, Heidelberg **5**, 236 (2006).

[82] S. Li, R. Chang, A. Boag, and V. Lomakin, Fast Electromagnetic Integral-Equation Solvers on Graphics Processing Units, IEEE Antennas Propag Mag **54**, 71 (2012).

[83] D. Van Orden and V. Lomakin, Rapidly Convergent Representations for 2D and 3D Green's Functions for a Linear Periodic Array of Dipole Sources, IEEE Trans Antennas Propag **57**, 1973 (2009).

[84] W. C. Chew, J.-M. Jin, C.-C. Lu, E. Michielssen, and J. M. Song, Fast solution methods in electromagnetics, IEEE Trans Antennas Propag **45**, 533 (1997).

[85] P. Giannozzi et al., QUANTUM ESPRESSO: a modular and open-source software project for quantum simulations of materials, J Phys Condens Matter **21**, (2009).

[86] J. Hafner, Ab-initio simulations of materials using VASP: Density-functional theory and beyond, J Comput Chem **29**, 2044 (2008).

[87] D. Bostick and M. L. Berkowitz, The implementation of slab geometry for membrane-channel molecular dynamics simulations, Biophys J **85**, 97 (2003).

[88] W. Weber, P. H. Hünenberger, and J. Andrew McCammon, Molecular Dynamics Simulations of a Polyalanine Octapeptide under Ewald Boundary Conditions: Influence of Artificial Periodicity on Peptide Conformation, Journal of Physical Chemistry B **104**, 3668 (2000).

[89] V. Subramanian and A. E. Yilmaz, An EM-CKT simulator for analyzing transient scattering from nonlinearly loaded periodic structures, 2014 USNC-URSI Radio Science Meeting (Joint with AP-S Symposium), USNC-URSI 2014 - Proceedings 71 (2014).

[90] S. T. Peng, T. Tamir, and H. L. Bertoni, Theory of Periodic Dielectric Waveguides, IEEE Trans Microw Theory Tech **23**, 123 (1975).

[91] A. L. Wysocki and V. P. Antropov, Micromagnetic simulations with periodic boundary conditions: Hard-soft nanocomposites, J Magn Magn Mater **428**, 274 (2017).

[92] D. Berkov and N. Gorn, Quasistatic remagnetization processes in two-dimensional systems with random on-site anisotropy and dipolar interaction: Numerical simulations, Phys Rev B **57**, 14332 (1998).

[93] K. H. . Huebner, The finite element method for engineers, 720 (2001).

[94] R. E. Tarjan, Efficiency of a Good But Not Linear Set Union Algorithm, Journal of the ACM (JACM) **22**, 215 (1975).

[95] B. Livshitz, A. Boag, H. N. Bertram, and V. Lomakin, Nonuniform grid algorithm for fast calculation of magnetostatic interactions in micromagnetics, Citation: Journal of Applied Physics **105**, 7 (2009).

[96] F. Ai and V. Lomakin, Fast Fourier Transform periodic interpolation method for superposition sums in a periodic unit cell, Comput Phys Commun **304**, 109291 (2024).

[97] S. Fu, R. Chang, I. Volvach, M. Kuteifan, M. Menarini, and V. Lomakin, Block Inverse Preconditioner for Implicit Time Integration in Finite Element Micromagnetic Solvers, IEEE Trans Magn **55**, (2019).

[98] J. Scott and M. Tuma, Incomplete Factorizations, Necas Center Series **2023**, 185 (2023).

[99] E. H. Frei, S. Shtrikman, and D. Treves, Critical Size and Nucleation Field of Ideal Ferromagnetic Particles, Physical Review **106**, 446 (1957).

[100] A. Aharoni and S. Shtrikman, Magnetization Curve of the Infinite Cylinder, Physical Review **109**, 1522 (1958).

[101] B. A. Kalinikos and A. N. Slavin, Theory of dipole-exchange spin wave spectrum for ferromagnetic films with mixed exchange boundary conditions, Journal of Physics C: Solid State Physics **19**, 7013 (1986).

[102] Z. Lin, I. Volvach, X. Wang, and V. Lomakin, Eigenvalue-Based Micromagnetic Analysis of Switching in Spin-Torque-Driven Structures, Phys Rev Appl **17**, 034016 (2022).