University of California, Merced

Investigation into the numerical simulation of strongly magnetized charged particle dynamics by exponential integration

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

in

Applied Mathematics

by

Tri Nguyen

Committee in charge:
Mayya Tokman, Chair
Ilon Joseph
John Loffeld
Changho Kim
Maxime Theillard

2024

This is to certify that I have examined a copy of the dissertation by

Tri Nguyen

and found it satisfactory in all respects, and that any and all revisions required by the dissertation committee have been made.

Applied Mathematics
Dissertation Committee Chair:

_____
Mayya Tokman

Dissertation Committee:

_____
Ilon Joseph

Dissertation Committee:

_____
John Loffeld

Dissertation Committee:

_____
Changho Kim

Dissertation Committee:

_____
Maxime Theillard

_____
Date

Contents

List of Figures

List of Tables

Acknowledgements

First, I wish to thank my advisor Mayya Tokman and my mentor Ilon Joseph. I am deeply grateful for their guidance, patience, and support. I also express my sincere appreciation to John Loffeld, Changho Kim, and Maxime Theillard for their assistance, thoughtful discussions, and service on my committee. Finally, I warmly thank all the friends and loved ones who have come my way throughout this journey.

The work in this dissertation was supported in part by the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract No. DE-AC52-07NA72344 and the National Science Foundation Award No. 1840265 and Award No. 2012875.

Curriculum Vitae

Tri (Alex) Nguyen

University of California, Merced
5200 N Lake Rd
Merced, CA 95343
tnguyen478@ucmerced.edu

Education

- **Ph.D., Applied Mathematics**
  University of California, Merced
  Advisor: Mayya Tokman
  2016–Present (Expected Summer 2024)

- **M.S. Mathematics, Applied Mathematics Emphasis**
  California State Polytechnic University, Pomona
  Advisor: Hubertus von Bremen
  2014–2016, 2020

- **B.S. Economics & Mathematics** (with Distinction), *Cum Laude*
  University of California, San Diego
  2010–2013

Relevant Experience

- **Graduate Teaching Assistant**
  University of California, Merced
  2016–Present

- **Graduate Student Researcher**
  Applied Mathematics
  University of California, Merced
  2019–2021

- **Graduate Student Intern**
  Fusion Energy Sciences Program
  Lawrence Livermore National Laboratory
  Summer 2019, Summer 2020

Publications

- T.P. Nguyen, I. Joseph, and M. Tokman, *Exploring exponential time integration for strongly magnetized charged particle motion*, Comput. Phys. Commun., 304 (2024), 109294, DOI 10.1016/j.cpc.2024.109294.

Presentations

- *Simulating Strongly Magnetized Charged Particle Dynamics by Nyström Type Exponential Integration*, 2024 SIAM Annual Meeting, Spokane, WA, Jul 8–12, 2024.
- *A Mathematical Model for the Simultaneous Hermaphrodite Mating Problem*, 2023 AMS Spring Eastern Sectional Meeting, Virtual, Apr 1–2, 2023.
- *Pushing Particles with Exponential Integrators*, 62nd Annual Meeting of the APS Division of Plasma Physics, Virtual, Nov 9–13, 2020.
- *Exponential Integration for Stiff Problems in Plasma Physics*, 2019 SIAM Conference on Analysis of PDEs, La Quinta, CA, Dec 11–14, 2019.
- *Simulating Charged Particle Dynamics with Exponential Integrators*, 61st Annual Meeting of the APS Division of Plasma Physics, Ft Lauderdale, FL, Oct 21–25, 2018.

Teaching (UC Merced, listed by course number)

MATH 11	Spring 2024	Calculus I
MATH 21	Spring 2023	Calculus I for Physical Sciences and Engineering
MATH 22	Fall 2016, Spring 2017, Fall 2022	Calculus II for Physical Sciences and Engineering
MATH 23	Fall 2017, Spring 2018	Vector Calculus
MATH 24	Summer 2024	Linear Algebra and Differential Equations
MATH 32	Summer I and II 2017	Probability and Statistics
MATH 125	Fall 2021	Intermediate Differential Equations
MATH 131	Fall 2018, Spring 2019	Numerical Methods for Scientists and Engineers
MATH 140	Spring 2022	Matematical Methods for Optimization
MATH 141	Fall 2023	Linear Analysis

Fellowships, Scholarships, Grants, and Awards

- **Applied Math Research Summer Fellowship**
  University of California, Merced
  2018, 2021, 2022, 2023

- **National Science Foundation DIRAC RTG Fellowship**
  University of California Merced
  2019–2021

- **Emil R. Herzog Scholarship in Mathematics**
  California State Polytechnic University, Pomona
  2015

- **Errett A. Bishop Memorial Scholarship**
  University of California, San Diego
  2012

Miscellaneous

Citizenship	U.S.A.
Computer Skills	C/C++, Mathematica, MATLAB, Python
Languages	English (Native), Spanish (Fluent)
Military Service	United States Marine Corps (Honorable Discharge)

Abstract

Title: Investigation into the numerical simulation of strongly magnetized charged particle dynamics by exponential integration
Name: Tri Nguyen
Degree: Ph.D.
University: University of California, Merced
Year: 2024
Committee Chair: Professor Mayya Tokman

The problem of solving for charged particle dynamics under the influence of electromagnetic fields is a fundamental component of particle simulation models to simulate plasma physics. This task - known as the particle pushing problem - is especially challenging when the plasma is strongly magnetized due to numerical stiffness arising from the wide range of time scales between highly oscillatory gyromotion and long term macroscopic behavior of the system, thus calling for computationally efficient numerical methods.

This dissertation investigates an alternative approach to numerically simulate strongly magnetized charged particle dynamics using exponential integration techniques. We first derive exponential integrators to solve strongly magnetized particle pushing problems and implement a novel algorithm to evaluate matrix φ functions that is computationally efficient for low dimensional problems. We then extend this work by deriving Nyström type exponential integrators that effectively solve the particle pushing problem in its second-order form directly. Numerical experiments comparing these exponential integrators against two conventional particle pushing algorithms demonstrate that exponential integrators are superior for linear and weakly nonlinear problems and are competitive for nonlinear problems. In particular, the Nyström type exponential integrators exhibit significant computational savings over the standard exponential integrators. These results demonstrate that exponential integrators are a promising alternative to numerically solve strongly magnetized particle pushing problems.

Chapter 1

Introduction

Computational simulation is an essential tool in plasma physics research. Simulations provide an alternative to physical experimentation, which can often be impractical or impossible. Moreover, simulations can be used to plan, design, and optimize physical experiments before their actual implementation.

In low density regimes, a natural computational framework to simulate plasma is by modeling it as a collection of charged particles [1, 4, 22], where at this microscopic scale the dominant forces acting on the particles are due to electromagnetic fields. A key component of this particle simulation framework is the problem of solving for charged particle dynamics under the influence of electromagnetic fields, a task known as the particle pushing problem. Since a large number of particles must be modeled in order for the simulation to be realistic, the particle pushing problem is a computationally intensive component of any particle simulation method.

In the case of strongly magnetized plasma, the particle pushing problem is particularly challenging due to multi-scale behavior of the system. At the fastest temporal scale of the system, charged particles gyrate about magnetic field lines in highly oscillatory gyromotion. However, the long term macroscopic behavior of plasma evolves on a significantly slower scale that can be orders of magnitude slower than the gyroperiod. As a consequence of this wide range of time scales, strongly magnetized particle pushing problems are numerically stiff. An example of this multi-scale phenomenon is illustrated in figure 1.1, which shows a cross-sectional view of a magnetic confinement device called a tokamak. Here, one can see that the scale of interest is the overall banana-shaped trajectory of the particle rather than the fast scale gyromotion.

Conventional numerical time integrators fall into two broad categories: explicit methods and implicit methods. Explicit methods calculate the solution at the next time step based upon already computed solutions at previous times. In other words, the update formula for the solution is an explicit expression of known values. Implicit methods, by contrast, calculate the solution at the next time step using not yet computed values of the solution at the future time. Hence, the update formula is an implicit equation with respect to the desired solution which has to be solved.

Explicit methods, although computationally cheap per time step, suffer from inferior numerical stability. Consequently, stiff problems impose a severe constraint on

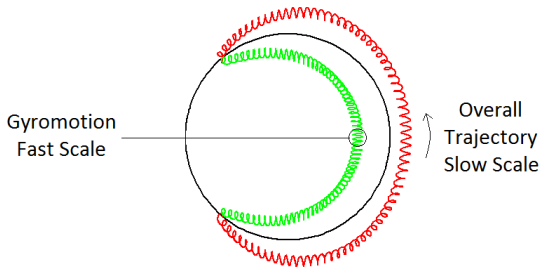


Figure 1.1: Trapped banana orbit

the maximum allowable step size for explicit methods, thus necessitating an excessive number of time steps and computation to approximate the solution over sufficiently long time intervals. Implicit methods, on the other hand, have better numerical stability properties and, thus, can take larger time steps. However, the numerical inversion of the implicit update formula can result in being prohibitively computationally expensive. Due to these limitations, there is a continuous need for the development of integrators with better stability properties than explicit methods and improved computational efficiency with respect to implicit methods for the stiff particle pushing problem.

This dissertation investigates an alternative approach to numerical simulation of the strongly magnetized particle dynamics using exponential integration. Exponential integrators form a class of numerical time stepping methods that solve linear problems exactly and are A-stable [21, 39, 40]. With these favorable accuracy and stability properties, exponential integration is a promising alternative to conventional methods to mitigate numerical stiffness and deliver accurate solutions for strongly magnetized particle pushing problems.

1.1 Background

1.1.1 Particle-In-Cell Methods

Particle-in-cell (PIC) methods have been extensively used to simulate plasmas in low density regimes [1, 4, 22]. In the PIC approach the spatial domain is discretized into computational cells and particle trajectories are advanced over discrete time intervals. PIC methods treat particle motion from the Lagrangian perspective in the sense that particle dynamics throughout the cells are computed by solving kinematic equations of motion. The electromagnetic fields, which are a combination of (known) externally applied fields and the self-consistent fields generated from the charged particle motion, are computed only at the cell edges. In this respect, the computation of the electromagnetic fields is treated from the Eulerian frame of reference. Starting with a known initial dynamical state of the particle and the electromagnetic fields at the cell edges, one iteration of the PIC algorithm proceeds through executing the

following set of directives:

1. Interpolate the electromagnetic fields from the cell edges to each particle's position.
2. Update each particle's dynamical state (position and velocity/momentum) by integrating the equations of motion.
3. Deposit each particle's charge back onto the cell edges using a shape function.
4. Update the self-consistent electromagnetic fields by solving Maxwell's equations.

Figure 1.2 shows a schematic of the PIC algorithm. Note that step two of the PIC algorithm is the particle pushing problem.

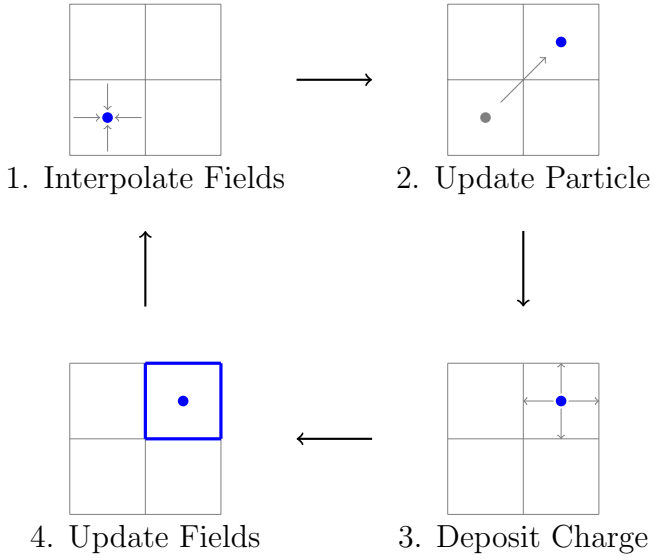


Figure 1.2: Schematic of the PIC algorithm.

1.1.2 The Particle Pushing Problem

The equation of motion governing charged particle dynamics under the influence of a magnetic field $\boldsymbol{B}$ and an electric field $\boldsymbol{E}$ is given by the Lorentz force

$$m\frac{d\boldsymbol{v}}{dt} = q\left(\boldsymbol{E} + \boldsymbol{v}\times\boldsymbol{B}\right), \tag{1.1}$$

where $\boldsymbol{v}$, q, and m are the particle velocity, electric charge, and mass, respectively. Denoting particle position by $\boldsymbol{x}$, equation (1.1) is equivalently expressed by the Newtonian system of equations:

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{v}, \tag{1.2a}$$

$$\frac{d\boldsymbol{v}}{dt} = \frac{q}{m}(\boldsymbol{E} + \boldsymbol{v}\times\boldsymbol{B}). \tag{1.2b}$$

Alternatively, the particle pushing problem has an equivalent formulation in terms of the position $\boldsymbol{x}$ and the (conjugate) momentum denoted by $\boldsymbol{p}$. The total energy of the system is given by the following Hamiltonian function

$$H(\boldsymbol{x}, \boldsymbol{p}) = \frac{1}{2m}\|\boldsymbol{p} - q\,\mathbf{A}(\boldsymbol{x})\|^2 + q\,V(\boldsymbol{x}), \tag{1.3}$$

where $\mathbf{A}(\boldsymbol{x})$ is the magnetic vector potential and $V(\boldsymbol{x})$ is the electric scalar potential that respectively yield the electromagnetic fields

$$\boldsymbol{B} = \nabla_{\boldsymbol{x}} \times \mathbf{A}(\boldsymbol{x}) \qquad \text{and} \qquad \boldsymbol{E} = -\nabla_{\boldsymbol{x}} V(\boldsymbol{x}).$$

Hamilton's equations then give the equations of motion:

$$\frac{d\boldsymbol{x}}{dt} = \quad \nabla_{\boldsymbol{p}} H(\boldsymbol{x}, \boldsymbol{p}) = \frac{1}{m}\big(\boldsymbol{p} - q\,\mathbf{A}(\boldsymbol{x})\big), \tag{1.4a}$$

$$\frac{d\boldsymbol{p}}{dt} = -\nabla_{\boldsymbol{x}} H(\boldsymbol{x}, \boldsymbol{p}) = \frac{q}{m}\left(\frac{\partial \mathbf{A}}{\partial \boldsymbol{x}}\right)^T \big(\boldsymbol{p} - q\,\mathbf{A}(\boldsymbol{x})\big) + q\,\boldsymbol{E}. \tag{1.4b}$$

Observe that both the Newtonian form (1.2) and Hamiltonian form (1.4) of the particle pushing problem are systems of first-order differential equations.

1.1.3 Charged Particle Motion in Electromagnetic Fields

This section describes the effects of various electromagnetic field configurations on the motion of a charged particle. In this dissertation, only the simplest types of particle motion are discussed:

- acceleration due to an electric field $\boldsymbol{E}$,
- gyromotion due to a uniform magnetic field $\boldsymbol{B}$,
- the $\boldsymbol{E} \times \boldsymbol{B}$ drift due to constant magnetic and electric fields perpendicular to each other, and
- the grad-B drift due to a non-uniform magnetic field with a perpendicular gradient term.

Focusing on these test problems will help us to thoroughly study the performance of the new exponential numerical approach. For a discussion on particles motion due to the effects of other types of non-uniformity in the electromagnetic fields, we direct the reader to references [21, 23, 30].

Electric Field - Acceleration

The simplest electromagnetic field configuration is one with no magnetic field and a uniform electric field $\boldsymbol{E}$. In this case the force acting on the particle is

$$m\frac{d\boldsymbol{v}}{dt} = q\,\boldsymbol{E}.$$

Thus, the particle experiences acceleration along the direction parallel to the electric field. Note that the sign of the charge determines the orientation of the acceleration: a particle with positive charge accelerates parallel to $\boldsymbol{E}$ while a particle with negative charge accelerates anti-parallel to $\boldsymbol{E}$.

Uniform Magnetic Field - Gyromotion

We next describe particle motion in a uniform magnetic field $\boldsymbol{B}$ and no electric field. Here, the Lorentz force equation (1.1) reduces to

$$\frac{d\boldsymbol{v}}{dt} = \frac{q}{m}(\boldsymbol{v} \times \boldsymbol{B}). \tag{1.5}$$

which implies that the magnetic field redirects particle velocity in a perpendicular direction while its speed $v = \|\boldsymbol{v}\|$ remains constant.

To show that the particle speed is unaffected by a magnetic field, we look at the kinetic energy of the particle:

$$\begin{aligned}\frac{1}{2}mv^2 &= \frac{1}{2}m\|\boldsymbol{v}\|^2 \\ &= \frac{1}{2}m(\boldsymbol{v}\cdot\boldsymbol{v}).\end{aligned}$$

Differentiating with respect to time gives

$$\begin{aligned}\frac{d}{dt}\left(\frac{1}{2}mv^2\right) &= \frac{d}{dt}\left(\frac{1}{2}m(\boldsymbol{v}\cdot\boldsymbol{v})\right) \\ &= m\,\boldsymbol{v}\cdot\frac{d\boldsymbol{v}}{dt} \\ &= m\,\boldsymbol{v}\cdot(\boldsymbol{v}\times\boldsymbol{B}).\end{aligned}$$

Noting that the cross product term $\boldsymbol{v}\times\boldsymbol{B}$ is perpendicular to $\boldsymbol{v}$, we see that the dot product term vanishes. Hence,

$$\frac{d}{dt}\left(\frac{1}{2}mv^2\right) = 0.$$

That is, the kinetic energy is constant, which implies the particle speed v remains unchanged.

We next show that the magnetic field redirects particle velocity in a perpendicular direction. Specifically, the magnetic field rotates particle velocity in the plane perpendicular to the field. Without loss of generality, suppose the magnetic field is in the z direction, $\boldsymbol{B} = B\,\hat{\boldsymbol{z}}$. Decomposing equation (1.5) into x, y, z components, we have:

$$\frac{dv_x}{dt} = \quad \omega\, v_y, \tag{1.6a}$$

$$\frac{dv_y}{dt} = -\omega\, v_x, \tag{1.6b}$$

$$\frac{dv_z}{dt} = \quad 0, \tag{1.6c}$$

where $\omega = qB/m$. Equations (1.6a) and (1.6b) imply that the magnetic field affects particle motion in the plane perpendicular to $\boldsymbol{B}$ while equation (1.6c) states that the velocity parallel to $\boldsymbol{B}$ is constant.

To derive particle motion in the perpendicular plane, we differentiate equations (1.6a) and (1.6b) once more with respect to time:

$$\frac{d^2v_x}{dt^2} = \quad \omega\,\frac{dv_y}{dt},$$

$$\frac{d^2v_y}{dt^2} = -\omega\,\frac{dv_x}{dt}.$$

Then substituting equations (1.6a) and (1.6b) into the right-hand side and after rearrangement, we get:

$$\frac{d^2v_x}{dt^2} + \omega^2\, v_x = 0,$$

$$\frac{d^2v_y}{dt^2} + \omega^2\, v_y = 0,$$

These are linear second-order differential equations with oscillatory solutions. Hence, the solutions for velocity can be written as:

$$v_x(t) = v_\perp \cos(\omega\, t + \phi) \tag{1.7a}$$

$$v_y(t) = v_\perp \sin(\omega\, t + \phi) \tag{1.7b}$$

$$v_z(t) = v_\parallel, \tag{1.7c}$$

where $v_\parallel$ and $v_\perp = \sqrt{v_x^2 + v_y^2}$ are constant speeds parallel and perpendicular to $\boldsymbol{B}$, respectively, and ϕ is a phase angle determined by the initial velocity.

To obtain expressions for the particle position, we simply integrate velocity to get:

$$x(t) = x_0 + \frac{v_\perp}{\omega}\sin(\omega\, t + \phi) - \frac{v_\perp}{\omega}\sin\phi, \tag{1.8a}$$

$$y(t) = y_0 - \frac{v_\perp}{\omega}\cos(\omega\, t + \phi) + \frac{v_\perp}{\omega}\cos\phi, \tag{1.8b}$$

$$z(t) = z_0 + v_\parallel t. \tag{1.8c}$$

Generalizing to an arbitrary magnetic field, charged particles gyrate about magnetic field lines in oscillatory gyromotion in the plane perpendicular to the magnetic field, where $\omega = qB/m$ is the gyrofrequency, $T = 2\pi/\omega$ is the gyroperiod, and $r = v_\perp/|\omega|$ is the gyroradius. The gyromotion of a positively charged particle is illustrated in figure 1.3.

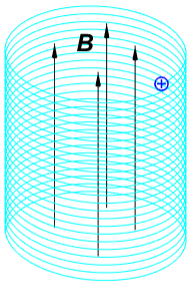


Figure 1.3: Gyromotion of a positively charged particle in a magnetic field $\boldsymbol{B}$.

$\boldsymbol{E} \times \boldsymbol{B}$ Drift

We now examine particle motion in a constant magnetic field $\boldsymbol{B}$ with a constant electric field $\boldsymbol{E}_\perp$ perpendicular to $\boldsymbol{B}$. As the particle goes through gyromotion from the $\boldsymbol{B}$ field, note that on one half of the gyro-orbit $\boldsymbol{E}_\perp$ decelerates the particle thereby decreasing $v_\perp$. Since $v_\perp$ decreases, the gyroradius r decreases as well. On the other half of the gyro-orbit $\boldsymbol{E}_\perp$ accelerates the particle so that $v_\perp$ increases, which in turn increases the gyroradius r. The net effect is that (in the plane perpendicular to $\boldsymbol{B}$) the gyromotion does not form closed orbits, but instead yields a drift motion perpendicular to both $\boldsymbol{E}_\perp$ and $\boldsymbol{B}$ as shown in figure 1.4.

To analyze this drift motion, suppose without loss of generality that $\boldsymbol{B} = B\,\hat{\boldsymbol{z}}$ and $\boldsymbol{E}_\perp$ is a constant electric field perpendicular to $\boldsymbol{B}$. If we conceptualize particle motion as circular motion (gyromotion) about the average particle position (the guiding center), then particle velocity can be expressed by

$$\boldsymbol{v} = \boldsymbol{v}_r + \boldsymbol{v}_R,$$

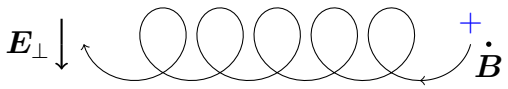


Figure 1.4: Drift motion from an electric field perpendicular $\boldsymbol{E}_\perp$ to the $\boldsymbol{B}$ field.

where $\boldsymbol{v}_r$ is the gyromotion velocity about the guiding center, and $\boldsymbol{v}_R$ is the guiding center velocity. In other words, the particle velocity is a superposition of the gyromotion velocity and the guiding center velocity. Then the total force on the particle is

$$\begin{aligned} m\frac{d\boldsymbol{v}}{dt} &= m\left(\frac{d\boldsymbol{v}_r}{dt} + \frac{d\boldsymbol{v}_R}{dt}\right) \\ &= q\big((\boldsymbol{v}_r + \boldsymbol{v}_R) \times \boldsymbol{B}\big) + q\,\boldsymbol{E}_\perp \\ &= q(\boldsymbol{v}_r \times \boldsymbol{B}) + q(\boldsymbol{v}_R \times \boldsymbol{B}) + q\,\boldsymbol{E}_\perp. \end{aligned}$$

Decomposing into components corresponding to the gyromotion and the guiding center, we have:

$$\frac{d\boldsymbol{v}_r}{dt} = \frac{q}{m}(\boldsymbol{v}_r \times \boldsymbol{B}), \tag{1.9a}$$

$$\frac{d\boldsymbol{v}_R}{dt} = \frac{q}{m}(\boldsymbol{v}_R \times \boldsymbol{B}) + \frac{1}{m}\boldsymbol{E}_\perp. \tag{1.9b}$$

Noting that equation (1.9a) is the gyromotion equation discussed in the previous section, we focus our analysis on equation (1.9b). Supposing the guiding center velocity $\boldsymbol{v}_R$ is constant, we have

$$\frac{d\boldsymbol{v}_R}{dt} = 0$$

implying that

$$-\frac{q}{m}\boldsymbol{E}_\perp = \boldsymbol{v}_R \times \boldsymbol{B} = \boldsymbol{\Omega}\,\boldsymbol{v}_R, \qquad \text{where } \boldsymbol{\Omega} = \begin{bmatrix} 0 & \omega \\ -\omega & 0 \end{bmatrix}.$$

Solving for the guiding center velocity $\boldsymbol{v}_R$, we get

$$\begin{aligned}
\boldsymbol{v}_R &= -\frac{q}{m}\boldsymbol{\Omega}^{-1}\boldsymbol{E}_\perp \\
&= -\frac{q}{m}\begin{bmatrix} 0 & -1/\omega \\ 1/\omega & 0 \end{bmatrix}\begin{bmatrix} E_x \\ E_y \end{bmatrix} \\
&= \frac{1}{B^2}\begin{bmatrix} 0 & B \\ -B & 0 \end{bmatrix}\begin{bmatrix} E_x \\ E_y \end{bmatrix} \\
&= \frac{\boldsymbol{E}_\perp \times \boldsymbol{B}}{B^2}.
\end{aligned} \tag{1.10}$$

Equation (1.10) tells us that $\boldsymbol{E}_\perp$ induces a constant drift velocity in the direction perpendicular to both $\boldsymbol{E}_\perp$ and $\boldsymbol{B}$. Generalizing for an arbitrary constant electric field $\boldsymbol{E}$, let

$$\boldsymbol{E} = \boldsymbol{E}_\parallel + \boldsymbol{E}_\perp,$$

where $\boldsymbol{E}_\parallel$ and $\boldsymbol{E}_\perp$ are the components parallel and perpendicular to $\boldsymbol{B}$, respectively. Then

$$\begin{aligned}
\boldsymbol{E} \times \boldsymbol{B} &= (\boldsymbol{E}_\parallel + \boldsymbol{E}_\perp) \times \boldsymbol{B} \\
&= (\boldsymbol{E}_\parallel \times \boldsymbol{B}) + (\boldsymbol{E}_\perp \times \boldsymbol{B}) \\
&= \boldsymbol{E}_\perp \times \boldsymbol{B},
\end{aligned}$$

where the cross product term $\boldsymbol{E}_\parallel \times \boldsymbol{B}$ vanishes because $\boldsymbol{E}_\parallel$ is parallel to $\boldsymbol{B}$. Hence, equation (1.10) generalizes for arbitrary constant electric field $\boldsymbol{E}$ and we see that the drift velocity induced by an arbitrary constant electric field $\boldsymbol{E}$ is

$$\boldsymbol{v}_E = \frac{\boldsymbol{E} \times \boldsymbol{B}}{B^2}. \tag{1.11}$$

In other words, particle motion in a magnetic field and an electric field is composed of gyromotion and drift motion called $\boldsymbol{E} \times \boldsymbol{B}$ drift.

Note that the above analysis considers the special case of a constant force due to a uniform $\boldsymbol{E}$ field. Generalizing to an arbitrary constant force $\boldsymbol{F}$, we have the following expression for drift motion induced by $\boldsymbol{F}$:

$$\boldsymbol{v}_R = \frac{1}{q}\left(\frac{\boldsymbol{F} \times \boldsymbol{B}}{B^2}\right). \tag{1.12}$$

Non-Uniform Magnetic Field - Grad-B Drift

We now consider particle motion in a non-uniform magnetic field. We consider the particular case where the change in the magnetic field is very small as the particle traverses over one gyro-orbit. Formally, we assume the gyroradius is much smaller than the characteristic scale length of the change in the magnetic field, which is mathematically expressed by

$$\frac{r\|\nabla B\|}{B} \sim \epsilon \ll 1.$$

Recall that the gyroradius is inversely proportional to the magnetic field strength. Consequently, the gyroradius is smaller as the particle orbits in a region where the magnetic field is stronger. Conversely, when the particle orbits in a region where the magnetic field is weaker, the gyroradius is larger. Similar to the $\boldsymbol{E} \times \boldsymbol{B}$ drift, the net effect is that the gyromotion does not form a closed orbit resulting in a trajectory with drift motion perpendicular to both the magnetic field $\boldsymbol{B}$ and its gradient ∇B, as shown in figure 1.5. This drift due to the non-uniformity of the magnetic field is called the grad-B drift.

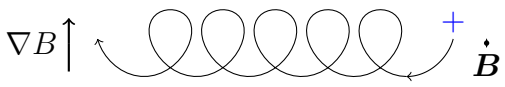


Figure 1.5: Grad-B drift motion

The equation of motion with a non-uniform magnetic field is expressed by a non-linear differential equation, which in general does not admit a closed form solution. However, under the assumption that the variation in the magnetic field is small over the gyro-orbit, we can find an approximate solution. To analyze this grad-B drift motion, we expand the magnetic field about the guiding center to get

$$\boldsymbol{B} = \boldsymbol{B}_0 + (\boldsymbol{r} \cdot \nabla)\boldsymbol{B} + \cdots .$$

Without loss of generality, let us take the origin to be at the guiding center and suppose

$$\boldsymbol{B} = \left(B_0 + y\frac{\partial B}{\partial y} + \cdots \right) \hat{\boldsymbol{z}}.$$

Note that since the guiding center is located at the origin, we have

$$x = x_R + x_r = x_r,$$
$$y = y_R + y_r = y_r,$$

where $(x_R, y_R)^T = (0,0)^T$ are the coordinates of the guiding center and $(x_r, y_r)^T$ are the coordinates of the particle gyrating about the guiding center. Then the Lorentz force is given by

$$\begin{aligned} m\frac{d\boldsymbol{v}}{dt} &= q\left[(\boldsymbol{v}_r \times \boldsymbol{B}) + (\boldsymbol{v}_R \times \boldsymbol{B})\right] \\ &= q\left[\left(\boldsymbol{v}_r \times \left(B + y\frac{\partial B}{\partial y} + \cdots\right)\hat{\boldsymbol{z}}\right) + \left(\boldsymbol{v}_R \times \left(B + y\frac{\partial B}{\partial y} + \cdots\right)\hat{\boldsymbol{z}}\right)\right] \\ &= m\omega\left[\left(\boldsymbol{v}_r \times \left(1 + \frac{y}{B}\frac{\partial B}{\partial y} + \cdots\right)\hat{\boldsymbol{z}}\right) + \left(\boldsymbol{v}_R \times \left(1 + \frac{y}{B}\frac{\partial B}{\partial y} + \cdots\right)\hat{\boldsymbol{z}}\right)\right]. \end{aligned}$$

By assumption,

$$\frac{r\|\nabla B\|}{B} = \frac{y}{B}\frac{\partial B}{\partial y} \sim \epsilon \ll 1.$$

Hence,

$$\begin{aligned} m\frac{d\boldsymbol{v}}{dt} =& m\omega\left[\left(\boldsymbol{v}_r \times \left(1 + \frac{y}{B}\frac{\partial B}{\partial y}\right)\hat{\boldsymbol{z}}\right) + \left(\boldsymbol{v}_R \times \left(1 + \frac{y}{B}\frac{\partial B}{\partial y}\right)\hat{\boldsymbol{z}}\right)\right] \\ &+ \mathcal{O}(\epsilon^2) \\ =& m\omega\left[(\boldsymbol{v}_r \times \hat{\boldsymbol{z}}) + \left(\boldsymbol{v}_r \times \frac{y}{B}\frac{\partial B}{\partial y}\hat{\boldsymbol{z}}\right) + (\boldsymbol{v}_R \times \hat{\boldsymbol{z}})\right. \\ &\left. + \left(\boldsymbol{v}_R \times \frac{y}{B}\frac{\partial B}{\partial y}\hat{\boldsymbol{z}}\right)\right] + \mathcal{O}(\epsilon^2). \end{aligned}$$

We now define the averaging operator $\langle\cdot\rangle$ by

$$\langle\cdot\rangle = \frac{1}{T}\int_0^T (\cdot)\, dt.$$

Then averaging over one gyro-orbit gives us

$$\begin{aligned} m\left\langle\frac{d\boldsymbol{v}}{dt}\right\rangle =& m\omega\left[\langle\boldsymbol{v}_r \times \hat{\boldsymbol{z}}\rangle + \left\langle\boldsymbol{v}_r \times \frac{y}{B}\frac{\partial B}{\partial y}\hat{\boldsymbol{z}}\right\rangle + \langle\boldsymbol{v}_R \times \hat{\boldsymbol{z}}\rangle + \left\langle\boldsymbol{v}_R \times \frac{y}{B}\frac{\partial B}{\partial y}\hat{\boldsymbol{z}}\right\rangle\right] \\ &+ \mathcal{O}(\epsilon^2) \\ =& m\omega\left[(\langle\boldsymbol{v}_r\rangle \times \hat{\boldsymbol{z}}) + \left\langle\boldsymbol{v}_r \times \frac{y}{B}\frac{\partial B}{\partial y}\hat{\boldsymbol{z}}\right\rangle + (\boldsymbol{v}_R \times \hat{\boldsymbol{z}})\right. \\ &\left. + \left(\boldsymbol{v}_R \times \left\langle\frac{y}{B}\frac{\partial B}{\partial y}\hat{\boldsymbol{z}}\right\rangle\right)\right] + \mathcal{O}(\epsilon^2). \end{aligned}$$

Observe that

$$\left\langle \frac{d\boldsymbol{v}}{dt} \right\rangle = \frac{d\boldsymbol{v}_R}{dt} = 0, \quad \langle \boldsymbol{v}_r \rangle = 0, \quad \text{and} \quad \left\langle \frac{y}{B}\frac{\partial B}{\partial y}\hat{\boldsymbol{z}} \right\rangle = 0.$$

Therefore, the effective force acting on the guiding center is

$$m\frac{d\boldsymbol{v}_R}{dt} = m\omega \left[\left\langle \boldsymbol{v}_r \times \frac{y}{B}\frac{\partial B}{\partial y}\hat{\boldsymbol{z}} \right\rangle + (\boldsymbol{v}_R \times \hat{\boldsymbol{z}}) \right] + \mathcal{O}(\epsilon^2). \tag{1.13}$$

This implies that to a leading order

$$\boldsymbol{v}_R \times \hat{\boldsymbol{z}} = -\omega \left\langle \boldsymbol{v}_r \times \frac{y}{B}\frac{\partial B}{\partial y} \right\rangle$$

Since we set the origin at the guiding center, by equations (1.7a) and (1.8b) we see that

$$y = y_r = -\frac{v_{rx}}{\omega_0},$$

which gives us

$$\begin{aligned}
\frac{d\boldsymbol{v}_R}{dt} &= -\omega \left(\boldsymbol{v}_r \times \frac{y_r}{B}\frac{\partial B}{\partial y} \right) \\
&= \boldsymbol{v}_r \times \frac{v_{rx}}{B}\frac{\partial B}{\partial y} \\
&= \frac{1}{B}\frac{\partial B}{\partial y}\left[v_{rx}v_{ry}\,\hat{\boldsymbol{x}} - v_{rx}^2\,\hat{\boldsymbol{y}} \right] \\
&= \frac{v_\perp^2}{B}\frac{\partial B}{\partial y}\left[\sin(\omega_0 t + \phi)\cos(\omega_0 t + \phi)\,\hat{\boldsymbol{x}} - \cos^2(\omega_0 t + \phi)\,\hat{\boldsymbol{y}} \right]
\end{aligned} \tag{1.14}$$

where for the components of the gyromotion velocity we made the substitutions

$$\begin{aligned}
v_{rx} &= v_\perp \cos(\omega_0 t + \phi), \\
v_{ry} &= v_\perp \sin(\omega_0 t + \phi).
\end{aligned}$$

Averaging equation (1.14) over one gyro-orbit and multiplying by m, we get

$$\begin{aligned}
m\left\langle \frac{dv_{Rx}}{dt} \right\rangle &= \langle F_{\nabla x} \rangle \\
&= \langle \boldsymbol{v}_r \times \hat{\boldsymbol{z}} \rangle_x \\
&= \frac{v_\perp^2}{B}\frac{\partial B}{\partial y}\langle \sin(\omega_0\, t + \phi)\cos(\omega_0\, t + \phi)\rangle \\
&= 0, \\
m\left\langle \frac{dv_{Ry}}{dt} \right\rangle &= \langle F_{\nabla y} \rangle \\
&= \langle \boldsymbol{v}_r \times \hat{\boldsymbol{z}} \rangle_y \\
&= -\frac{v_\perp^2}{B}\frac{\partial B}{\partial y}\langle \cos^2(\omega_0\, t + \phi)\rangle \\
&= -\frac{v_\perp^2}{2B}\frac{\partial B}{\partial y}, \\
m\left\langle \frac{dv_{Rz}}{dt} \right\rangle &= \langle F_{\nabla z} \rangle \\
&= \langle \boldsymbol{v}_r \times \hat{\boldsymbol{z}} \rangle_z \\
&= 0,
\end{aligned}$$

where $\boldsymbol{F}_\nabla = (F_{\nabla x}, F_{\nabla y}, F_{\nabla z})^T$ is the effective grad-B drift force. Substituting the average of this force $\langle \boldsymbol{F}_\nabla \rangle$ into equation (1.12) gives the leading order approximation for the grad-B drift velocity

$$\boldsymbol{v}_R = -\left(\frac{1}{2}mv_\perp^2\right)\left(\frac{1}{q\,B}\right)\frac{\partial B}{\partial y}\,\hat{\boldsymbol{x}}.$$

Generalizing to an arbitrary magnetic field $\boldsymbol{B}$, the drift velocity is

$$\begin{aligned}
\boldsymbol{v}_\nabla &= -\left(\frac{\frac{1}{2}mv_\perp}{q}\right)\frac{\boldsymbol{B}\times\nabla B}{B^3} \\
&= -\left(\frac{W_\perp}{q}\right)\frac{\boldsymbol{B}\times\nabla B}{B^3},
\end{aligned} \tag{1.15}$$

where $W_\perp = \frac{1}{2}mv_\perp$ is the kinetic energy perpendicular to the magnetic field $\boldsymbol{B}$.

To reiterate, the particular particle motions described above ($\boldsymbol{E}$ field acceleration, gyromotion, $\boldsymbol{E}\times\boldsymbol{B}$ drift motion, and grad-B drift motion) were chosen for test problems to validate the time integrators proposed in this dissertation.

1.1.4 Conventional Particle Pushing Framework

The text of this section is a reprint of the material as it appears in T.P. Nguyen, I. Joseph and M. Tokman, Exploring exponential time integration for strongly magnetized charged particle motion, *Computer Physics Communications*, 304 (2024), 109294, doi: https://doi.org/10.1016/j.cpc.2024.109294.

The standard approach to numerical particle pushing approximates the Newtonian equations of motion (1.2) with the finite-difference model

$$\frac{\boldsymbol{x}_{n+1} - \boldsymbol{x}_n}{h} = \boldsymbol{v}_{n+1/2}, \tag{1.16a}$$

$$\frac{\boldsymbol{v}_{n+1/2} - \boldsymbol{v}_{n-1/2}}{h} = \frac{q}{m}\left(\boldsymbol{E}_n + \frac{\boldsymbol{v}_{n+1/2} + \boldsymbol{v}_{n-1/2}}{2} \times \boldsymbol{B}_n\right), \tag{1.16b}$$

where h is a fixed time step size and the subscripts n, $n\pm1/2$, $n+1$ denote times t_n, $t_n\pm$h/2, t_n+h, respectively. Position and the electromagnetic fields are computed at integer time nodes while velocity is computed at half-integer time nodes. This staggering of position and velocity by one-half time step gives a leapfrog-like, centered-difference, time reversible scheme with second-order accuracy. Observe that the second equation (1.16b) is implicit in $\boldsymbol{v}_{n+1/2}$ and, hence, numerically stable. However, the step size h must be sufficiently small such that the electric field $\boldsymbol{E}$ and magnetic field $\boldsymbol{B}$ are approximately constant over the time interval $[t_n, t_n + h]$ to yield accurate solutions.

It follows from equation (1.16a) that the finite-difference model approximates the second derivative of position with the centered difference formula:

$$\frac{d^2\boldsymbol{x}}{dt^2} \approx \frac{\boldsymbol{x}_{n+1} - 2\boldsymbol{x}_n + \boldsymbol{x}_{n-1}}{h^2}.$$

Birdsall and Langdon [1] pointed out that in order for this model to properly capture harmonic motion, the time step size for equations (1.16a) and (1.16b) must obey the restriction:

$$h < \frac{2}{|\omega|}, \qquad \omega = \frac{qB}{m}.$$

The update formula for the particle position is given by a simple rearrangement of (1.16a):

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + h\,\boldsymbol{v}_{n+1/2}.$$

For the Lorentz force equation (1.16b), note that the right-hand side is composed of an electric push term and a magnetic rotation term due to the electric field $\boldsymbol{E}_n$ and magnetic field $\boldsymbol{B}_n$, respectively. Also observe that the updated velocity $\boldsymbol{v}_{n+1/2}$ is given implicitly, which requires inversion of the equation to get an explicit expression for $\boldsymbol{v}_{n+1/2}$. Two common algorithms to resolve these tasks and update the particle velocity are the Buneman [6] and the Boris [2] particle pushers. Currently, the Boris particle pusher is the most frequently used algorithm for PIC applications [34].

Buneman Particle Pusher

The Buneman particle pushing algorithm decomposes the action of the electric field $\boldsymbol{E}$ on particle velocity into components parallel and perpendicular to the magnetic field $\boldsymbol{B}$. In the presence of an electric field $\boldsymbol{E}$ and a magnetic field $\boldsymbol{B}$, the particle experiences a so-called $\boldsymbol{E} \times \boldsymbol{B}$ drift velocity (perpendicular to both fields)

$$\boldsymbol{v}_{\text{drift}} = \frac{\boldsymbol{E} \times \boldsymbol{B}}{B^2}, \quad B = \|\boldsymbol{B}\|.$$

The Buneman algorithm subtracts this drift from the particle velocities at time nodes $t_{n-1/2}$ and $t_{n+1/2}$ thereby defining two intermediate velocities:

$$\begin{aligned} \boldsymbol{v}^- &= \boldsymbol{v}_{n-1/2} - \boldsymbol{v}_{\text{drift}}, \\ \boldsymbol{v}^+ &= \boldsymbol{v}_{n+1/2} - \boldsymbol{v}_{\text{drift}}. \end{aligned}$$

Substituting $\boldsymbol{v}_{n-1/2}$ and $\boldsymbol{v}_{n+1/2}$ into equation (1.16b) then yields

$$\frac{\boldsymbol{v}^+ - \boldsymbol{v}^-}{h} = \frac{q}{m}\left(\boldsymbol{E}_{\|} + \frac{\boldsymbol{v}^+ + \boldsymbol{v}^-}{2} \times \boldsymbol{B}\right).$$

The above formula is composed of acceleration parallel to the magnetic field (the $\boldsymbol{E}_{\|}$ term) and a rotation of the velocity perpendicular to the magnetic field (the cross product term). For a uniform magnetic field with magnitude B, the angle of magnetic rotation over time step h is

$$\theta = h\omega, \qquad \omega = \frac{qB}{m}.$$

The Buneman algorithm updates the velocity from $\boldsymbol{v}^-$ to $\boldsymbol{v}^+$ by the formula

$$\boldsymbol{v}^+ = \cos\theta\, \boldsymbol{v}^- - \sin\theta \left(\frac{\boldsymbol{B}}{B} \times \boldsymbol{v}^-\right).$$

As a historical note, the Bunemam algorithm was introduced in 1967 [6] during which time the evaluation of transcendental functions was computationally expensive. To reduce computational cost, the algorithm makes use of the small angle approximation

$$w = \frac{h}{2}\frac{qB}{m} \approx \tan\left(\frac{\theta}{2}\right).$$

Then, using half-angle trigonometric identities, $\sin\theta$ and $\cos\theta$ are computed as follows:

$$\begin{aligned} s &= \frac{2\,w}{1+w^2} = \sin\theta, \\ c &= \frac{1-w^2}{1+w^2} = \cos\theta. \end{aligned}$$

Algorithm 1: Buneman Velocity Push

Input: h, q, m, $\boldsymbol{B}_n$, $B = \|\boldsymbol{B}_n\|$, $\boldsymbol{E}_n$, $\boldsymbol{v}_{n-1/2}$

Output: $\boldsymbol{v}_{n+1/2}$

1: $\boldsymbol{v}_{\text{drift}} \leftarrow \dfrac{\boldsymbol{E}_n \times \boldsymbol{B}_n}{B^2}$

2: $\boldsymbol{v}^- \leftarrow \boldsymbol{v}_{n-1/2} - \boldsymbol{v}_{\text{drift}}$

3: $w \leftarrow \dfrac{h}{2}\dfrac{qB}{m}$

4: $s \leftarrow \dfrac{2\,w}{1+w^2}$

5: $c \leftarrow \dfrac{1-w^2}{1+w^2}$

6: $\boldsymbol{v}^+ \leftarrow c\,\boldsymbol{v}^- - s\left(\dfrac{\boldsymbol{B}_n}{B} \times \boldsymbol{v}^-\right)$

7: $\boldsymbol{v}_{n+1/2} \leftarrow \boldsymbol{v}^+ + \boldsymbol{v}_{\text{drift}}$

Boris Particle Pusher

The Boris algorithm takes an alternative approach to the velocity update by decoupling the electric push and magnetic rotation in (1.16b). The discussion presented here is taken from [1]. The algorithm defines two intermediate velocities $\boldsymbol{v}^-$ and $\boldsymbol{v}^+$ by the relations

$$\boldsymbol{v}_{n-1/2} = \boldsymbol{v}^- - \frac{h}{2}\frac{q}{m}\boldsymbol{E}_n,$$
$$\boldsymbol{v}_{n+1/2} = \boldsymbol{v}^+ + \frac{h}{2}\frac{q}{m}\boldsymbol{E}_n.$$

Substituting the above expressions into the Lorentz force equation (1.16b) cancels the $\boldsymbol{E}_n$ term resulting in the magnetic rotation equation

$$\frac{\boldsymbol{v}^+ - \boldsymbol{v}^-}{h} = \frac{q}{m}\left(\frac{\boldsymbol{v}^+ + \boldsymbol{v}^-}{2} \times \boldsymbol{B}_n\right). \tag{1.17}$$

Thus, the actions due to the electric field $\boldsymbol{E}_n$ and the magnetic field $\boldsymbol{B}_n$ are decoupled and velocity is updated in a Strang-like splitting scheme as follows:

i. First-half electric push $\quad \boldsymbol{v}^- = \boldsymbol{v}_{n-1/2} + \dfrac{h}{2}\dfrac{q}{m}\boldsymbol{E}_n;$

ii. Magnetic rotation $\quad \dfrac{\boldsymbol{v}^+ - \boldsymbol{v}^-}{h} = \dfrac{q}{m}\left(\dfrac{\boldsymbol{v}^+ + \boldsymbol{v}^-}{2} \times \boldsymbol{B}_n\right);$

iii. Second-half electric push $\quad \boldsymbol{v}_{n+1/2} = \boldsymbol{v}^+ + \dfrac{h}{2}\dfrac{q}{m}\boldsymbol{E}_n.$

Figure 1.6 illustrates an example of the velocity update in a configuration where the magnetic field is pointing out of the plane of the page, the electric field points from left to right, and the initial particle velocity $\boldsymbol{v}_{n-1/2}$ is perpendicular to the magnetic field.

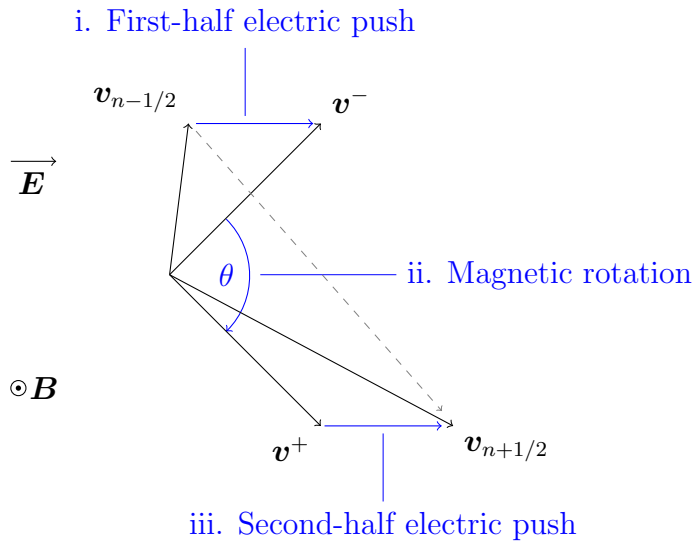


Figure 1.6: Boris velocity update

Observe that the magnetic rotation equation (1.17) is an implicit expression in $\boldsymbol{v}^+$ and, therefore, requires inversion to get an explicit expression. The Boris algorithm achieves this inversion as follows. First, an intermediate velocity $\boldsymbol{v}'$ is defined to be the vector that bisects the magnetic rotation angle θ in the plane perpendicular to the magnetic field $\boldsymbol{B}$. Furthermore, $\boldsymbol{v}'$ is specified such that a right triangle is formed with $\boldsymbol{v}'$ as the hypotenuse and $\boldsymbol{v}^-$ as one of the legs. This implies that there exist a scalar w such that the other leg of the triangle is given by

$$\boldsymbol{v}^- \times w\,\hat{\boldsymbol{B}},$$

where $\hat{\boldsymbol{B}}$ is the unit vector in the direction of $\boldsymbol{B}$; see figure 1.7. Letting $\alpha = \theta/2$, we see that

$$\tan\alpha = \frac{|\boldsymbol{v}^- \times w\hat{\boldsymbol{B}}|}{|\boldsymbol{v}^-|} = \frac{|\boldsymbol{v}^-|w}{|\boldsymbol{v}^-|} = w.$$

Hence, by straightforward vector addition

$$\boldsymbol{v}' = \boldsymbol{v}^- + \boldsymbol{v}^- \times \tan\alpha\, \hat{\boldsymbol{B}}, \quad \alpha = \frac{\theta}{2} = \frac{h}{2}\frac{qB}{m}.$$

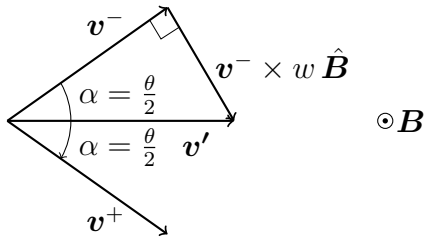


Figure 1.7: Vector $\boldsymbol{v}'$ bisects the magnetic rotation angle θ. A right triangle is formed by the vectors $\boldsymbol{v}^-$, $\boldsymbol{v}'$, and $\boldsymbol{v}^- \times w\,\hat{\boldsymbol{B}}$ for some scalar w.

The algorithm next solves for the vector $\boldsymbol{v}^+ - \boldsymbol{v}^-$ by making use of the fact that it is perpendicular to both $\boldsymbol{v}'$ and $\boldsymbol{B}$. Hence, there exist some scalar u such that

$$\boldsymbol{v}^+ - \boldsymbol{v}^- = \boldsymbol{v}' \times u\,\hat{\boldsymbol{B}}.$$

To find the value of u, refer to figure 1.8 and observe that

$$\sin\alpha = \frac{\frac{1}{2}|\boldsymbol{v}' \times u\,\hat{\boldsymbol{B}}|}{|\boldsymbol{v}^+|} = \frac{|\boldsymbol{v}'|u}{2|\boldsymbol{v}^+|}.$$

Solving for u gives

$$u = \frac{2|\boldsymbol{v}^+|\sin\alpha}{|\boldsymbol{v}'|}.$$

Substituting $\boldsymbol{v}' = \boldsymbol{v}^- + \boldsymbol{v}^- \times \tan\alpha\hat{\boldsymbol{B}}$ and making use of the fact $|\boldsymbol{v}^+| = |\boldsymbol{v}^-|$ ($\boldsymbol{v}^+$ is $\boldsymbol{v}^-$ rotated by angle θ), we get

$$u = \frac{2\sin\alpha}{\sqrt{1+\tan^2\alpha}} = \frac{2\tan\alpha}{1+\tan^2\alpha}.$$

Thus, the update from $\boldsymbol{v}^-$ to $\boldsymbol{v}^+$ is given by the formula

$$\boldsymbol{v}^+ = \boldsymbol{v}^- + \boldsymbol{v}' \times \frac{2\tan\alpha}{1+\tan^2\alpha}\hat{\boldsymbol{B}}.$$

Similar to the Buneman algorithm, the Boris algorithm was introduced at a time (1970) when the evaluation of transcendental functions was computationally expensive. Therefore, implementations of the Boris algorithm typically use the small angle approximation

$$w = \alpha \approx \tan\alpha$$

in step 2 of algorithm 2.

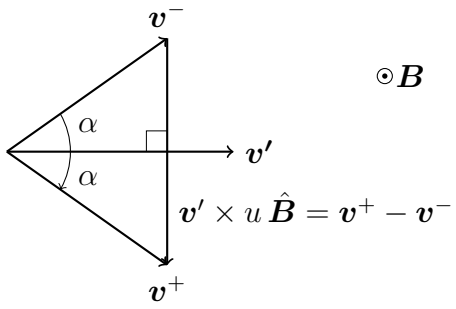


Figure 1.8: Vector $\boldsymbol{v}' \times u\,\hat{\boldsymbol{B}}$ is equal to $\boldsymbol{v}^+ - \boldsymbol{v}^-$ and perpendicular to both $\boldsymbol{v}'$ and $\boldsymbol{B}$.

Algorithm 2: Boris Velocity Push

Input: h, q, m, $\boldsymbol{B}_n$, $B = \|\boldsymbol{B}_n\|$, $\boldsymbol{E}_n$, $\boldsymbol{v}_{n-1/2}$

Output: $\boldsymbol{v}_{n+1/2}$

1: $\boldsymbol{v}^- \leftarrow \boldsymbol{v}_{n-1/2} + \dfrac{h}{2}\dfrac{q}{m}\boldsymbol{E}_n$

2: $w \leftarrow \dfrac{h}{2}\dfrac{qB}{m}$

3: $\boldsymbol{v}' \leftarrow \boldsymbol{v}^- + \boldsymbol{v}^- \times \dfrac{1}{B}\,w\,\boldsymbol{B}_n$

4: $\boldsymbol{v}^+ \leftarrow \boldsymbol{v}^- + \boldsymbol{v}' \times \dfrac{1}{B}\dfrac{2\,w}{1+w^2}\boldsymbol{B}_n$

5: $\boldsymbol{v}_{n+1/2} \leftarrow \boldsymbol{v}^+ + \dfrac{h}{2}\dfrac{q}{m}\boldsymbol{E}_n$

Other Particle Pushing Algorithms

The investigation of computationally efficient numerical particle pushers continues to be an active research field. Notable among the more recent developments are the energy-conserving, asymptotic preserving scheme [35] and the filtered Boris algorithm [19]. The first method is a modified implicit Crank-Nicolson scheme that converses energy and incorporates an effective force in the velocity update that captures the leading order grad-B drift motion in non-uniform magnetic fields. This effective force is carefully chosen such that it approximates the grad-B force acting on the guiding center in a gyro-averaged sense. The filtered Boris algorithm, on the other hand, modifies the standard Boris pusher by introducing so-called filter functions to more accurately resolve the fast oscillations in particle velocity due to strong magnetic fields. Different choices of filter functions and choices of the positions where the

magnetic field is evaluated yield different variants of the filtered Boris algorithm. It is interesting to note that for problems with constant magnetic and electric fields, the filtered Boris algorithm reduces to a type of exponential integrator that solves the problem exactly. However, this dissertation shall demonstrate that the exponential integrators proposed herein are exact solvers for problems with a constant magnetic field and electric fields that are linear functions of the particle position in addition to constant electric fields. Both the modified Crank-Nicolson scheme and the filtered Boris algorithm (for the general case of arbitrary magnetic fields) are implicit methods and are more complex to implement than the standard Boris algorithm. Hence, they have the advantage of allowing for larger time step sizes for problems with non-uniform electromagnetic fields, but at the cost of being more computationally expensive than the standard Boris pusher.

1.2 Dissertation Structure

The organization of this dissertation is as follows. Chapter 2 explores exponential integration as a novel numerical particle pushing method. A framework for deriving an exponential integrator is presented along with two specific example schemes from the Exponential Propagation Iterative methods of Runge-Kutta type (EPIRK) class [39, 40]: a second-order Exponential Propagation (EP2) method and a third-order Exponential Propagation method of Runge-Kutta type (EPRK3). A key to the efficiency of our numerical approach is the use of the Lagrange-Sylvester Interpolation Polynomial formula to compute the exponential-like φ matrix functions. For low dimensional problems such as simulating charged particle dynamics, evaluation of the exponential matrix functions outperforms the common Krylov subspace projection-based techniques. Results of numerical experiments comparing the exponential integrators against the conventional Boris and Buneman pushers show that exponential integrators yield superior performance for linear and weakly nonlinear problems, and are competitive with the conventional pushers for strongly nonlinear problems.

Chapter 3 builds upon the previous chapter by taking the standard exponential integration framework as a template to derive even more computationally efficient Nyström type exponential methods that integrate the particle pushing problem in its second-order form (1.1) directly instead as a system of first-order equations. In particular, we derive a second-order Exponential Propagation method of Runge-Kutta-Nyström type (EPRKN2) and a third-order Exponential Propagation method of Runge-Kutta-Nyström type (EPRKN3). Results of numerical experiments are presented showing a significant improvement in the computation speeds of the EPRKN2 and EPRKN3 methods over the standard EP2 and EPRK3 schemes.

The final chapter discusses, summarizes and concludes the work presented in this dissertation. Directions for future research are also outlined.

Chapter 2

Exploring exponential time integration for strongly magnetized charged particle motion

The text of this chapter is a reprint of the material as it appears in T.P. Nguyen, I. Joseph and M. Tokman, Exploring exponential time integration for strongly magnetized charged particle motion, *Computer Physics Communications*, 304 (2024), 109294, doi: https://doi.org/10.1016/j.cpc.2024.109294.

2.1 Introduction

Solving for charged particle dynamics is a key problem in particle-in-cell (PIC) simulations of plasma physics, a task known as the particle pushing problem. Since realistic simulations call for the modeling of a vast number of particles, the problem is computationally intensive. This task is especially challenging when the plasma is strongly magnetized in which case charged particles gyrate about magnetic field lines in highly oscillatory gyromotion. In contrast, the macroscopic evolution of the system occurs on a time scale orders of magnitude slower. The presence of such a wide range of time scales in the system results in the numerical stiffness of the equations modeling the dynamics. Moreover, accuracy requirements of simulations typically demand resolution at the scale of the gyromotion, which necessitate small time steps for conventional time integration schemes. These difficulties, therefore, call for computationally efficient numerical particle pushing methods.

The standard approach to solving the particle pushing problem numerically is to discretize the equations of motion with a finite-difference model from which the dynamical state of the particle is advanced by a time stepping algorithm [1, 22]. Two well-known examples of this conventional approach are the Boris [2] and Buneman [6] particle pushers. Both methods stagger particle position and velocity by one-half time step resulting in a leapfrog-like centered-difference scheme that gives second-order accuracy in time. The Boris algorithm, in particular, currently enjoys status

as the de facto particle pusher [34]. A fundamental requirement in this framework is that electromagnetic fields be approximately constant over each time step size. Consequently, problems with large field gradients demand small step sizes to maintain accurate solutions, which results in excessive computational expense.

The investigation of computationally efficient numerical particle pushers continues to be an active research field. For example, investigation into numerical methods that address this time step size restriction include [3, 10, 11, 13, 14, 17, 19, 35, 41].

This chapter explores an alternative approach to numerical particle pushing using a technique called exponential integration. Exponential integrators approximate the solution of a nonlinear dynamical system in terms of exponential-like functions of matrices which are either Jacobians of the system or their approximations. Exponential methods offer several desirable features. By construction, exponential time integrators solve the linear portion of the particle pushing problem exactly thus accounting for the electric field gradient component of the solution. While traditional particle pushers such as the Boris and Buneman algorithms assume that the electric field gradient is nearly zero over the course of the time step, exponential integration methods allow for a non-zero gradient and enable larger time steps to be taken. In addition, since computing individual particle trajectories is a low dimensional problem, it is possible to evaluate the exponential-like functions of the Jacobians required by exponential methods with relatively low computational cost. In this chapter we exploit the good stability properties of the exponential integration methods and the low-dimensionality of the particle pushing problem to propose exponential integrators for calculating the dynamics of particles under the influence of a strong constant magnetic field and spatially varying electric fields. Similar to the modified Crank-Nicolson scheme [35] and the filtered Boris algorithm [19], the exponential integrators presented here can compute accurate solutions using larger step sizes but are more complex and computationally expensive than conventional particle pushers. However, these exponential integrator particle pushers are explicit methods in contrast to the Crank-Nicolson and filtered Boris pushers. We emphasize that since this is an initial exploration into the relatively novel approach of numerical particle pushing by exponential integration, this study focuses on problems with a uniform magnetic field as a first step and defer investigation into problems with non-uniform magnetic fields for future work.

The organization of this chapter is as follows: Section 2 describes the equations of motion of the particle pushing problem. Section 3 presents exponential integrators used for solving these equations. A computational technique to evaluate the exponential-like matrix functions which constitutes the main computational expense of an exponential integrator is discussed in section 4. Numerical experiments for several test problems comparing exponential integrators with the Boris and Buneman algorithms are presented in section 5. Finally, section 6 summarizes and concludes this chapter. A proof of the theorem justifying our method to compute matrix functions is presented in an appendix.

2.2 Exponential Integrator Particle Pusher

Observe that if the particle state is known at time $t = t_n$, then the particle pushing problem is of the form

$$\frac{d\boldsymbol{u}}{dt} = \boldsymbol{f}(\boldsymbol{u}), \quad \boldsymbol{u}_n = \boldsymbol{u}(t_n), \tag{2.1}$$

where $\boldsymbol{f}(\boldsymbol{u})$ is the right-hand side function of the equations of motion. Taking a first-order Taylor expansion of the right-hand side function of (2.1) about the known state $\boldsymbol{u}_n$, we get

$$\frac{d\boldsymbol{u}}{dt} = \boldsymbol{f}(\boldsymbol{u}_n) + \boldsymbol{A}_n(\boldsymbol{u} - \boldsymbol{u}_n) + \boldsymbol{r}(\boldsymbol{u}), \tag{2.2}$$

where

$$\boldsymbol{A}_n = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\right|_{\boldsymbol{u}=\boldsymbol{u}_n}$$

is the Jacobian matrix and

$$\boldsymbol{r}(\boldsymbol{u}) = \boldsymbol{f}(\boldsymbol{u}) - \boldsymbol{f}(\boldsymbol{u}_n) - \boldsymbol{A}_n(\boldsymbol{u} - \boldsymbol{u}_n) \tag{2.3}$$

is the nonlinear remainder term. Multiplying equation (2.2) by the integrating factor $\exp(-t\boldsymbol{A}_n)$ and then integrating over the time interval $[t_n, t_n + h]$, we obtain the integral equation

$$\boldsymbol{u}(t_n + h) = \boldsymbol{u}_n + h\,\varphi_1(h\boldsymbol{A}_n)\boldsymbol{f}(\boldsymbol{u}_n) + \int_{t_n}^{t_n+h} e^{\boldsymbol{A}_n(t_n+h-t)}\boldsymbol{r}(\boldsymbol{u}(t))\,dt, \tag{2.4}$$

where $\varphi_1(h\boldsymbol{A}_n)$ is a matrix function defined by the MacLaurin series expansion of the scalar analytic function

$$\varphi_1(z) = \frac{e^z - 1}{z} = \int_0^1 e^{z(1-\tau)}\,d\tau$$

applied to the matrix argument $h\boldsymbol{A}_n$. Letting $t = t_n + \tau h$, equation (2.4) is equivalently expressed by

$$\boldsymbol{u}(t_n + h) = \boldsymbol{u}_n + h\,\varphi_1(hA_n)\,\boldsymbol{f}(\boldsymbol{u}_n) + h\int_0^1 e^{h\boldsymbol{A}_n(1-\tau)}\,\boldsymbol{r}(\boldsymbol{u}(t_n + \tau h))\,d\tau. \tag{2.5}$$

Equation (2.5) is a starting point from which an exponential integrator can be derived as follows. Let $\boldsymbol{u}_n$ be a numerical solution obtained at a previous integration step and let h be a specified time step size. Then formula (2.5) gives the exact solution at the next time step $\boldsymbol{u}(t_n + h)$. To approximate $\boldsymbol{u}(t_n + h)$ we can construct an exponential integrator by accomplishing the following two tasks:

(i) Develop a quadrature rule to approximate the nonlinear integral term

$$h \int_0^1 e^{h\boldsymbol{A}_n(1-\tau)} \boldsymbol{r}(\boldsymbol{u}(t_n + \tau h))\, d\tau;$$

(ii) Construct a technique to compute exponential-like matrix functions, called φ functions.

The Exponential Propagation Iterative Methods of Runge-Kutta-type (EPIRK) framework has been shown to allow construction of efficient exponential methods that reduce computational cost per time step compared to other exponential integrators [39, 40]. These methods have been shown to be computationally efficient for several applications including MHD modeling [12]. Thus, this is the first class of exponential methods we will explore for solving the particle pushing problem.

The formulation for a general EPIRK method is given by the following ansatz:

$$\boldsymbol{U}_i = \boldsymbol{u}_0 + a_{i1}\psi_{i1}(g_{i1}h\boldsymbol{A}_0)\, h\boldsymbol{f}(\boldsymbol{u}_0) + \sum_{j=2}^{i} a_{ij}\psi_{ij}(g_{ij}h\boldsymbol{A}_0)\, \Delta^{(j-1)}\boldsymbol{r}(\boldsymbol{u}_0), \tag{2.6a}$$

$$i = 1, 2, \ldots, s-1,$$

$$\boldsymbol{u}_1 = \boldsymbol{u}_0 + b_1\psi_{s1}(g_{s1}h\boldsymbol{A}_0)\, h\boldsymbol{f}(\boldsymbol{u}_0) + \sum_{j=2}^{s} b_j\psi_{sj}(g_{sj}h\boldsymbol{A}_0)\, h\Delta^{(j-1)}\boldsymbol{r}(\boldsymbol{u}_0), \tag{2.6b}$$

where the matrix ψ_{ij} functions are defined by the scalar functions

$$\psi_{ij}(z) = \sum_{k=1}^{s} p_{ijk}\, \varphi_k(z)$$

with

$$\varphi_k(z) = \int_0^1 e^{z(1-\tau)} \frac{\tau^{k-1}}{(k-1)!}\, d\tau, \qquad k = 1, 2, \ldots,$$

and the vectors $\Delta^{(j-1)}\boldsymbol{r}(\boldsymbol{u}_0)$ are the $j-1$th forward differences of the nonlinear remainder function (2.3) computed on the nodes $\boldsymbol{u}_0, \boldsymbol{U}_1, \boldsymbol{U}_2, \ldots, \boldsymbol{U}_{s-1}$. Here, the first through the $j-1$th forward differences of the nonlinear remainder function are defined by:

$$\begin{aligned}
\Delta\boldsymbol{r}(\boldsymbol{u}_0) &= \boldsymbol{r}(\boldsymbol{U}_1) - \boldsymbol{r}(\boldsymbol{u}_0), \\
\Delta^2\boldsymbol{r}(\boldsymbol{u}_0) &= \Delta\boldsymbol{r}(\boldsymbol{U}_1) - \Delta\boldsymbol{r}(\boldsymbol{u}_0) \\
&= \boldsymbol{r}(\boldsymbol{U}_2) - 2\boldsymbol{r}(\boldsymbol{U}_1) + \boldsymbol{r}(\boldsymbol{u}_0), \\
\vdots \quad &\quad \vdots \quad \vdots \\
\Delta^{j-1}\boldsymbol{r}(\boldsymbol{u}_0) &= \Delta^{j-2}\boldsymbol{r}(\boldsymbol{U}_{j-1}) - \Delta^{j-2}\boldsymbol{r}(\boldsymbol{u}_0) \\
&= \sum_{i=0}^{j-1} (-1)^i \binom{j-1}{\mathrm{i}} \boldsymbol{r}(\boldsymbol{U}_{j-1-i}), \quad \text{where } \boldsymbol{U}_0 = \boldsymbol{u}_0.
\end{aligned}$$

The coefficients $a_{ij}, g_{ij}, b_j, p_{ijk}$ are determined by satisfying the desired order conditions. This procedure has been used to derive the following two methods:

- Second-order exponential propagation method [40]

$$\boldsymbol{u}_{n+1} = \boldsymbol{u}_n + h\,\varphi_1(h\boldsymbol{A}_n)\boldsymbol{f}(\boldsymbol{u}_n), \tag{EP2}$$

- Third-order exponential propagation, Runge-Kutta type method [37]

$$\begin{aligned} \boldsymbol{U_1} &= \boldsymbol{u}_n + h\,\varphi_1\left(\tfrac{3}{4}\,h\boldsymbol{A}_n\right)\,\boldsymbol{f}(\boldsymbol{u}_n), \\ \boldsymbol{R}_1 &= \boldsymbol{f}(\boldsymbol{U_1}) - \boldsymbol{f}(\boldsymbol{u}_n) - \boldsymbol{A}_n(\boldsymbol{U_1} - \boldsymbol{u}_n), \\ \boldsymbol{u}_{n+1} &= \boldsymbol{u}_n + h\,\varphi_1(h\boldsymbol{A}_n)\,\boldsymbol{f}(\boldsymbol{u}_n) + 2\,h\,\varphi_3(h\boldsymbol{A}_n)\,\boldsymbol{R}_1. \end{aligned} \tag{EPRK3}$$

Note that in principle exponential integrators solve linear differential equations exactly. As a consequence, the region of stability for exponential integrators is the left-half of the complex plane; i.e. exponential integrators are A-stable.

2.3 Computing the Matrix Functions

In any exponential integration scheme the most computationally expensive step is the evaluation of each action of an exponential-like matrix function φ. For small matrices, approximation techniques such as a finite Taylor polynomial, Padé approximation, or scaling and squaring have been common approaches to compute the matrix function $\varphi(\boldsymbol{A})$. However, these methods are quite computationally expensive and are usually only used when the computational cost of evaluating matrix functions is not important. (For a detailed discussion on the computational issues of various methods to evaluate the exponential of a matrix, see [28,29].) Other methods include the Leja method [8] and Krylov subspace projection methods. Krylov subspace projection methods, in particular, have been shown to be computationally efficient techniques to approximate the action the matrix φ function on a vector when the matrix is large [16,31].

The particle pushing problem, however, is a low dimensional problem. Even in three dimensions only six equations of motion have to be integrated simultaneously to advance a particle's trajectory. Exploiting this low dimensionality, we propose an alternative approach to compute the matrix φ functions by means of evaluating a finite degree matrix polynomial that yields an analytic result. We show that this direct analytic method is computationally efficient for such small problems.

The following theorem [5,15,38] asserts that any analytic matrix function has an exact expression in terms of a finite degree matrix polynomial.

Theorem 1 (Lagrange-Sylvester Interpolation Polynomial Formula)**.** *Let $\boldsymbol{A}$ be an $N \times N$ matrix and let f be a scalar function analytic in a domain containing the spectrum of $\boldsymbol{A}$. Then there exists a unique polynomial p of (at most) degree $N-1$ such that:*

1. *If the eigenvalues of* $\boldsymbol{A}$ *are all distinct, then* p *is the polynomial that interpolates* f *on the spectrum of* $\boldsymbol{A}$*;*

2. *If* $\boldsymbol{A}$ *has repeated eigenvalues, then* p *is the polynomial that interpolates* f *on the spectrum of* $\boldsymbol{A}$*. In addition, for each eigenvalue* λ_j *with multiplicity* r_j*, the polynomial* p *also satisfies* $r_j - 1$ *osculating conditions in the sense that all derivatives up to order* $r_j - 1$ *of both* p *and* f *agree with each other at the interpolation node* λ_j*. In other words:*

$$\begin{array}{rcll} p(\lambda_j) & = & f(\lambda_j) & \textit{interpolation condition,} \\ p'(\lambda_j) & = & f'(\lambda_j) & \textit{1st osculating condition,} \\ p''(\lambda_j) & = & f''(\lambda_j) & \textit{2nd osculating condition,} \\ \vdots & \vdots & \vdots & \vdots \\ p^{(r_j-1)}(\lambda_j) & = & f^{(r_j-1)}(\lambda_j) & r_j - 1^{th} \textit{ osculating condition,} \end{array}$$

where the superscript denotes the order of the derivative with respect to λ*.*

In either case, the polynomial p *applied to the matrix argument* A *is equivalent to the matrix function* $f(A)$*. That is,*

$$p(A) = f(A).$$

Proof. See appendix A. □

Our method applies this theorem to calculate the matrix exponential-like φ_k functions, which is presented in Algorithm 3. Note that the numerical computation of the scalar $\varphi_k(z)$ functions for $k \geq 1$ is subject to catastrophic cancellation for small argument values $z = h\lambda_j$. To overcome this issue, we employ the Cauchy integral formula suggested by Kassam and Trefethen [25]. Furthermore, our particular implementation of the Lagrange-Sylvester formula employs Newton divided differences [7, 26] to calculate the interpolation polynomial. That is, we seek the polynomial of the form

$$\begin{aligned} p(\lambda) = b_0 + b_1(\lambda - \lambda_1) + b_2(\lambda - \lambda_1)(\lambda - \lambda_2) + \ldots \\ + b_{N-1}(\lambda - \lambda_1) \cdots (\lambda - \lambda_{N-1}) \end{aligned}$$

that agrees with the φ_k function on the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_N$, where the polynomial coefficients are given by the Newton divided differences:

$$\begin{array}{rcl} b_0 & = & \varphi_k[\lambda_1], \\ b_1 & = & \varphi_k[\lambda_1, \lambda_2], \\ b_2 & = & \varphi_k[\lambda_1, \lambda_2, \lambda_3], \\ \vdots & \vdots & \vdots \\ b_{N-1} & = & \varphi_k[\lambda_1, \ldots, \lambda_N], \end{array} \quad .$$

Here, the Newton divided differences on the right-hand side are defined as follows. The zeroth divided difference is

$$\varphi_k[\lambda_i] = \varphi_k(\lambda_i).$$

The first divided difference is

$$\varphi_k[\lambda_i, \lambda_{i+1}] = \begin{cases} \varphi'(\lambda_{i+1}) & \text{if } \lambda_i = \lambda_{i+1}, \\ \dfrac{\varphi_k[\lambda_{i+1}] - \varphi_k[\lambda_i]}{\lambda_{i+1} - \lambda_i} & \text{otherwise.} \end{cases}$$

The second divided difference is

$$\varphi_k[\lambda_i, \lambda_{i+1}, \lambda_{i+2}] = \begin{cases} \dfrac{1}{2!}\varphi_k''(\lambda_i) & \text{if } \lambda_i = \lambda_{i+1} = \lambda_{i+2}, \\ \dfrac{\varphi_k[\lambda_{i+1,i+2}] - \varphi_k[\lambda_i, \lambda_{i+1}]}{\lambda_{i+2} - \lambda_i} & \text{otherwise.} \end{cases}$$

By recursive definition, the j^{th} divided difference is

$$\varphi_k[\lambda_i, \ldots, \lambda_{i+j}]$$
$$= \begin{cases} \dfrac{1}{j!}\varphi_k^{(j)}(\lambda_{i+j}) & \text{if } \lambda_i, \ldots, \lambda_{i+j} \text{ are all equal,} \\ \dfrac{\varphi_k[\lambda_{i+1}, \ldots, \lambda_{i+j}] - \varphi_k[\lambda_i, \ldots, \lambda_{i+j-1}]}{\lambda_{i+j} - \lambda_i} & \text{otherwise,} \end{cases}$$

where the superscript denotes the order of the derivative of the φ_k function with respect to λ. Thus, our algorithm is a generalization of Method 10 in [28, 29], which computes the matrix exponential, using an interpolation polynomial calculated with Newton divided differences.

To illustrate the computational efficiency of Algorithm 3 we compared two implementations of the second-order EP2 and third-order EPRK3 exponential integrators using (i) a Krylov subspace projection method called KIOPS [16], and (ii) the Lagrange-Sylvester formula in MATLAB. Both implementations of the exponential integrators are used to solve the Hamiltonian equations of motion over the time interval $[0, 100]$ for a particle of unit mass and unit charge in a uniform magnetic field $\boldsymbol{B} = 100\,\hat{\boldsymbol{z}}$ with electric fields

$$\boldsymbol{E} = -\frac{100}{3}\begin{bmatrix} x^3 \\ y^3 \end{bmatrix} \quad \text{for the two-dimensional model}$$

and

$$\boldsymbol{E} = -\frac{1}{3}\begin{bmatrix} 100x^3 \\ 100y^3 \\ 10z^3 \end{bmatrix} \quad \text{for the three-dimensional model.}$$

Algorithm 3: Lagrange-Sylvester Formula to compute the matrix function $\varphi_k(h\boldsymbol{A})$

1: Solve for the eigenvalues of $\boldsymbol{A}$.
2: Solve for the interpolation polynomial p such that for each eigenvalue λ_j:

$$\begin{aligned} p(\lambda_j) &= \varphi_k(h\lambda_j), \\ p'(\lambda_j) &= \varphi_k'(h\lambda_j), \\ p''(\lambda_j) &= \varphi_k''(h\lambda_j), \\ \vdots &\quad \vdots \quad\quad \vdots \\ p^{(r_j-1)}(\lambda_j) &= \varphi_k^{(r_j-1)}(h\lambda_j), \end{aligned}$$

where $r_j \geq 1$ is the multiplicity of λ_j and the superscript denotes the order of the derivative with respect to λ.
3: Evaluate the matrix polynomial $p(\boldsymbol{A})$, which is equal to $\varphi_k(h\boldsymbol{A})$.

The computed solutions were compared against a reference solution obtained by the MATLAB `ode113` solver with error tolerances set to 10^{-12} for `RelTol` (relative error tolerance) and 10^{-12} for `AbsTol` (absolute error tolerance). Relative error of the exponential integrator solution is defined as

$$\text{error} = \frac{\|\boldsymbol{x}^* - \boldsymbol{x}\|}{\|\boldsymbol{x}^*\|},$$

where $\boldsymbol{x}^*$ is the particle position of the reference solution and $\boldsymbol{x}$ is the particle position of the exponential integrator solution, both evaluated at the final time $t = 100$, and $\|\cdot\|$ denotes the Euclidean norm.

Figure 2.1 shows precision diagrams (CPU time vs error) comparing two implementations of the EP2 and EPRK3 integrators. As we can see from the figure, the Lagrange-Sylvester formula enables significant computational savings for exponential integration compared to the KIOPS methods (with iteration convergence tolerance set to 1e−9). For the two-dimensional test problem, the EPI2 integrator using KIOPS takes on average seven times longer than the EP2 integrator using the Lagrange-Sylvester formula to compute the final solution. Similarly, the EPIRK3 integrator using KIOPS takes on average 2.7 times longer than the EPRK3 integrator using the Lagrange-Sylvester formula to compute the final solution for the two-dimensional test problem. For the three-dimensional test problem, the EPI2 integrator using KIOPS takes on average four times longer than the EP2 integrator using the Lagrange-Sylvester formula to compute the final solution. Likewise, the EPIRK3 integrator using KIOPS takes on average 2.8 times longer than the EPRK3 integrator using the Lagrange-Sylvester formula to compute the final solution for the three-dimensional test problem. This is expected since the KIOPS technique is designed for large scale problems and we expect the Lagrange-Sylvester formula to be more efficient for these

low-dimensional systems.

We note that the Jacobian matrices of the problem possess some structure. For example, the Newtonian formulation of the problem yields a zero block matrix and an identity block matrix inside the Jacobian. Symmetries also exist in the Hamiltonian form of the Jacobian as well. It is possible that additional computational savings can be derived for both the Lagrange-Sylvester and KIOPS algorithms. We will investigate this direction in the future.

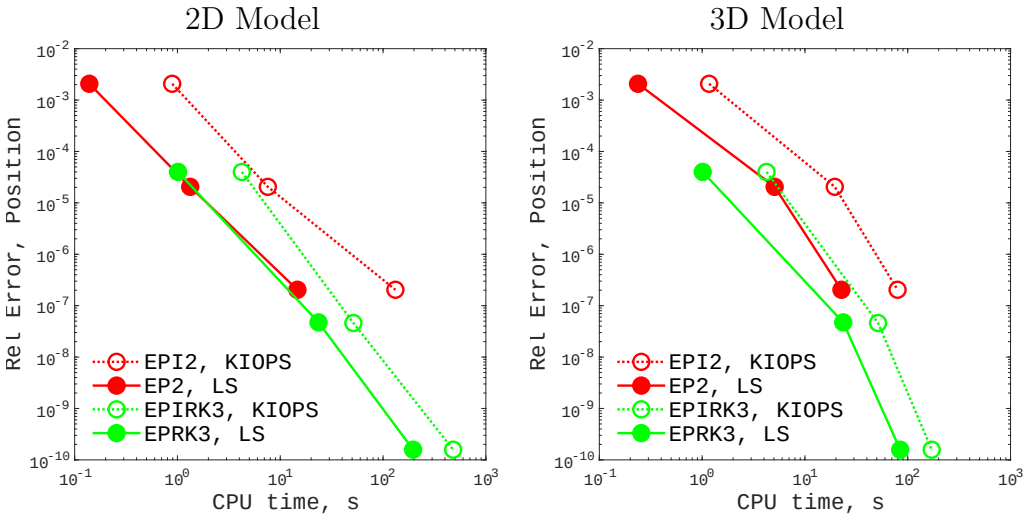


Figure 2.1: Precision diagram showing performances of 2nd and 3rd order exponential integrators subspace projection (dotted lines) and Lagrange-Sylvester formula (solid lines) for step sizes $h = 0.01, 0.001, 0.0001$ over the time interval [0, 100].

2.4 Numerical Experiments

To assess the performance of exponential integrators for the particle pushing problems we used a series of test configurations and compare these integrators to the widely used Boris and Buneman algorithms. We selected the second and third order exponential methods EP2 and EPRK3 to integrate the Hamiltonian form of the equations of motion (1.4). Both exponential integrators are implemented with the Lagrange-Sylvester interpolation formula to compute the matrix φ functions as described in Algorithm 3.

The test problems under examination model a particle of unit mass and unit charge in a uniform in time and space magnetic field aligned in the z direction, $\boldsymbol{B} = 100\,\hat{\boldsymbol{z}}$, and a non-uniform electric field $\boldsymbol{E}$ resulting in anisotropic drift motion of particles along periodic orbits. Specifically, the test problems are set up with electric fields characterized by electric scalar potential wells and hills (in the xy plane) of quadratic, cubic, and quartic forms. To enable comparison across the different potentials, the configurations of the potentials are such that the largest absolute eigenvalue of the

Hessian matrix for each potential form (quadratic, cubic, and quartic) is set to the same value at the initial particle position. Our reasoning is that the largest absolute eigenvalue, which we denote by $|V''|$, gives a measure of the gradient of the electric field and also is a rough estimate of the electric oscillation frequency. The ratio of this eigenvalue to the magnetic field strength defines an ordering parameter that determines different regimes of particle motion [24]. Thus, we conduct numerical experiments for several values of this ratio, i.e. $|V''|/B = 1/100$, $|V''|/B = 1/10$, and $|V''|/B = 1$. The equations were solved over the time interval of $[0, 100]$ which is equivalent to just nearly 1,600 gyroperiods.

To get an estimate of the error in our numerical experiments, we computed approximations to the solutions of test problems using the MATLAB `ode113` integrator with error tolerances set to 10^{-12} for `RelTol` (relative error tolerance) and 10^{-12} for `AbsTol` (absolute error tolerance) and designated it as the reference solution. The relative error of the numerical solution is defined by

$$\text{error } = \frac{\|\boldsymbol{x}^* - \boldsymbol{x}\|}{\|\boldsymbol{x}^*\|},$$

where $\boldsymbol{x}^*$ is the particle position of the reference solution, $\boldsymbol{x}$ is the particle position of the solution of the particle pusher, and $\|\cdot\|$ denotes the Euclidean norm.

We also examined the particle energy over the longer time interval of $[0, 20000]$ corresponding to over 3.18×10^5 gyroperiods. Each electric scalar potential is configured such that at the initial condition the particle energy is unity. Since the particle pushing problem is a Hamiltonian system, energy is a conserved quantity and, therefore, any error in the computed energy gives an indication of the long term accuracy of the particle pusher under examination. The experiments were first performed with all particle pushers using the same step size set to the minimum of $h = 0.01$ or the largest step size such that the energy error is within 10% of the true value. That is,

$$h = \min\{0.01, \underline{h}\},$$

where

$$\underline{h} \coloneqq \max\left\{h > 0 \colon \quad 0.9 \leq \text{energy} \leq 1.1 \quad \text{and} \quad \frac{20000}{h} \in \mathbb{N}\right\}.$$

The experiments were then repeated for the exponential integrators using the largest step such that relative energy error of the EP2 solution is within 10% of the true energy.

For the two dimensional models, we also included two additional experiments. The first experiment examines the performances of the particle pushers for a simple non-uniform magnetic field problem called the grad-B drift problem. The second experiment examines the computed gyroradius of each particle pusher for a linear $\boldsymbol{E} \times \boldsymbol{B}$ drift problem.

All experiments in this section were run on a PC with an Intel Core i7-1255U processor at clock speed 1.7 GHz and 16 GB of RAM and implemented in C++ using the Eigen C++ template library for linear algebra [18] with the exception of

the grad-B drift problem and the gyroradius experiments, which were implemented in MATLAB. All computations were calculated with double precision floating point operations.

2.4.1 Two Dimensional Model

All two-dimensional test problems set initial particle position and velocity at $\boldsymbol{x}_0 = (1, 0)$ and $\boldsymbol{v}_0 = (0, -1)$, respectively. Table 2.1 lists the configurations for the electric scalar potential wells and their corresponding electric fields. Configurations for electric scalar potential hills and the corresponding electric fields are shown in table 2.2.

Plots of the reference solution orbits and precision diagrams for test problems with $|V''|/B = 1/100$, $|V''|/B = 1/10$, and $|V''|/B = 1$ are shown in figures 2.2, 2.3, and 2.4, respectively. They show that the performances of the particle pushers are roughly similar between the potential well problems and the potential hill problems. For the test problems with quadratic potentials, the exponential integrators exhibit superior performance as expected, because the problems are linear for which exponential integrators solve exactly. For the nonlinear test problems with cubic and quartic potentials the computational advantage of the exponential methods is not as dramatic but they are still competitive with the Boris and Buneman particle pushers.

Figures 2.5, 2.6, and 2.7 show the energy plots for test problems with $|V''|/B = 1/100$, $|V''|/B = 1/10$, and $|V''|/B = 1$, respectively. Note that the exponential integrators compute the exact energy in one single time step for linear test problems with quadratic potentials. For the nonlinear test problems with cubic and quartic potentials, we point out several key observations. Since these exponential integrators have not been designed to preserve energy exactly, their computed energies are expected to drift over the time interval. However, the errors in energy of the exponential methods remains within the same bounds of the errors for the Boris and Buneman algorithms for comparable time step sizes. For large step sizes, the drift causes the energy to eventually exceed those bounds. It is also important to note that the EPRK3 integrator performs better than the EP2 integrator in two respects: the EP2 energy drifts are larger than the EPRK3 energy drifts and there is wider variation in the EP2 energies compared to the EPRK3 energies. These results indicate a possibility of construction of higher order exponential methods that can yield sufficient accuracy within the time interval of interest and, if they are carefully designed, could still remain competitive from the efficiency standpoint. We will pursue development of such techniques in our future publications.

		$\lvert V'' \rvert = 1$	$\lvert V'' \rvert = 10$	$\lvert V'' \rvert = 100$
Quadratic Well	V	$\frac{1}{2}(x^2+y^2)$	$-\frac{9}{2}+5(x^2+y^2)$	$-\frac{99}{2}+50(x^2+y^2)$
	$\boldsymbol{E}$	$-\begin{bmatrix} x \\ y \end{bmatrix}$	$-10\begin{bmatrix} x \\ y \end{bmatrix}$	$-100\begin{bmatrix} x \\ y \end{bmatrix}$
Cubic Well	V	$-\frac{2}{3}+x^2+y^2$ $+\frac{1}{6}(x^3+y^3)$	$-\frac{19}{6}+3(x^2+y^2)$ $+\frac{2}{3}(x^3+y^3)$	$-\frac{95}{2}+47(x^2+y^2)$ $+x^3+y^3$
	$\boldsymbol{E}$	$-\begin{bmatrix} 2x-\frac{1}{2}x^2 \\ 2y-\frac{1}{2}y^2 \end{bmatrix}$	$-\begin{bmatrix} 6x-2x^2 \\ 6y-2y^2 \end{bmatrix}$	$-\begin{bmatrix} 94x+3x^2 \\ 94y+3y^2 \end{bmatrix}$
Quartic Well	V	$\frac{5}{12}+\frac{1}{12}(x^4+y^4)$	$-\frac{1}{3}+\frac{5}{6}(x^4+y^4)$	$-\frac{47}{6}+\frac{25}{3}(x^4+y^4)$
	$\boldsymbol{E}$	$-\frac{1}{3}\begin{bmatrix} x^3 \\ y^3 \end{bmatrix}$	$-\frac{10}{3}\begin{bmatrix} x^3 \\ y^3 \end{bmatrix}$	$-\frac{100}{3}\begin{bmatrix} x^3 \\ y^3 \end{bmatrix}$

Table 2.1: Electric scalar potential wells and corresponding electric fields for 2D model test problems

		$\lvert V'' \rvert = 1$	$\lvert V'' \rvert = 10$	$\lvert V'' \rvert = 100$
Quadratic Hill	V	$1 - \frac{1}{2}(x^2 + y^2)$	$\frac{11}{2} - 5(x^2 + y^2)$	$\frac{101}{2} - 50(x^2 + y^2)$
	$\boldsymbol{E}$	$\begin{bmatrix} x \\ y \end{bmatrix}$	$10 \begin{bmatrix} x \\ y \end{bmatrix}$	$100 \begin{bmatrix} x \\ y \end{bmatrix}$
Cubic Hill	V	$\frac{4}{3} - x^2 - y^2$ $+\frac{1}{6}(x^3 + y^3)$	$\frac{25}{6} - 3(x^2 + y^2)$ $-\frac{2}{3}(x^3 + y^3)$	$\frac{97}{2} - 47(x^2 + y^2)$ $-x^3 - y^3$
	$\boldsymbol{E}$	$\begin{bmatrix} 2x - \frac{1}{2}x^2 \\ 2y - \frac{1}{2}y^2 \end{bmatrix}$	$\begin{bmatrix} 6x - 2x^2 \\ 6y - 2y^2 \end{bmatrix}$	$\begin{bmatrix} 94x + 3x^2 \\ 94y + 3y^2 \end{bmatrix}$
Quartic Hill	V	$\frac{7}{12} - \frac{1}{12}(x^4 + y^4)$	$\frac{4}{3} - \frac{5}{6}(x^4 + y^4)$	$\frac{53}{6} - \frac{25}{3}(x^4 + y^4)$
	$\boldsymbol{E}$	$\frac{1}{3} \begin{bmatrix} x^3 \\ y^3 \end{bmatrix}$	$\frac{10}{3} \begin{bmatrix} x^3 \\ y^3 \end{bmatrix}$	$\frac{100}{3} \begin{bmatrix} x^3 \\ y^3 \end{bmatrix}$

Table 2.2: Electric scalar potential hills and corresponding electric fields for 2D model test problems

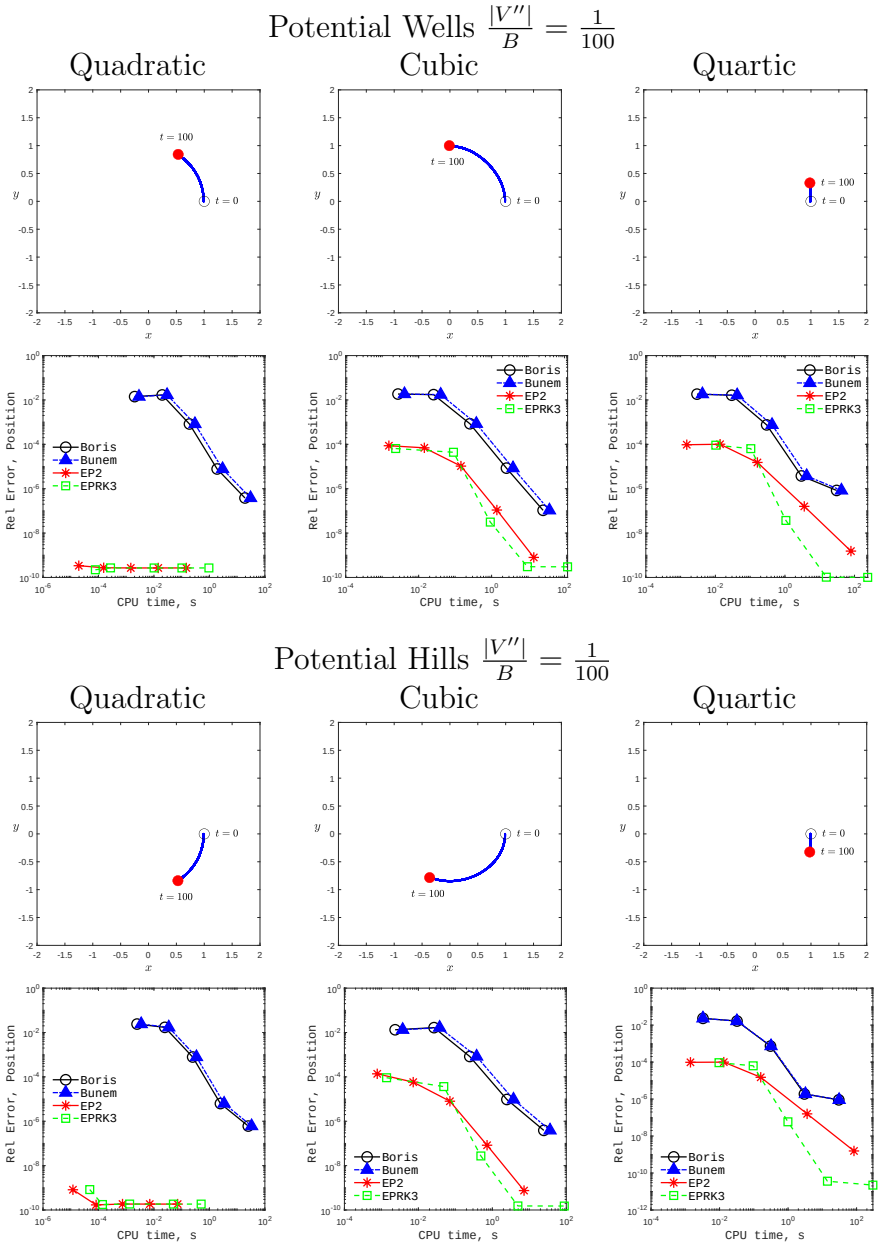


Figure 2.2: Results for 2D test problems with $|V''|/B = 1/100$: potential well reference solution orbits (first row), potential well precision diagrams (second row), potential hill reference solution orbits (third row), and potential hill precision diagrams (fourth row). Boris/Buneman step sizes are $h = 10^{-2}$, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6} for quadratic potential problems and $h = 10^{-3}$, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7} for cubic/quartic potential problems. EP2/EPRK3 step sizes are $h = 100$, 10, 1, 10^{-1}, 10^{-2} for quadratic potential problems and $h = 10^{-1}$, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} for cubic/quartic potential problems.

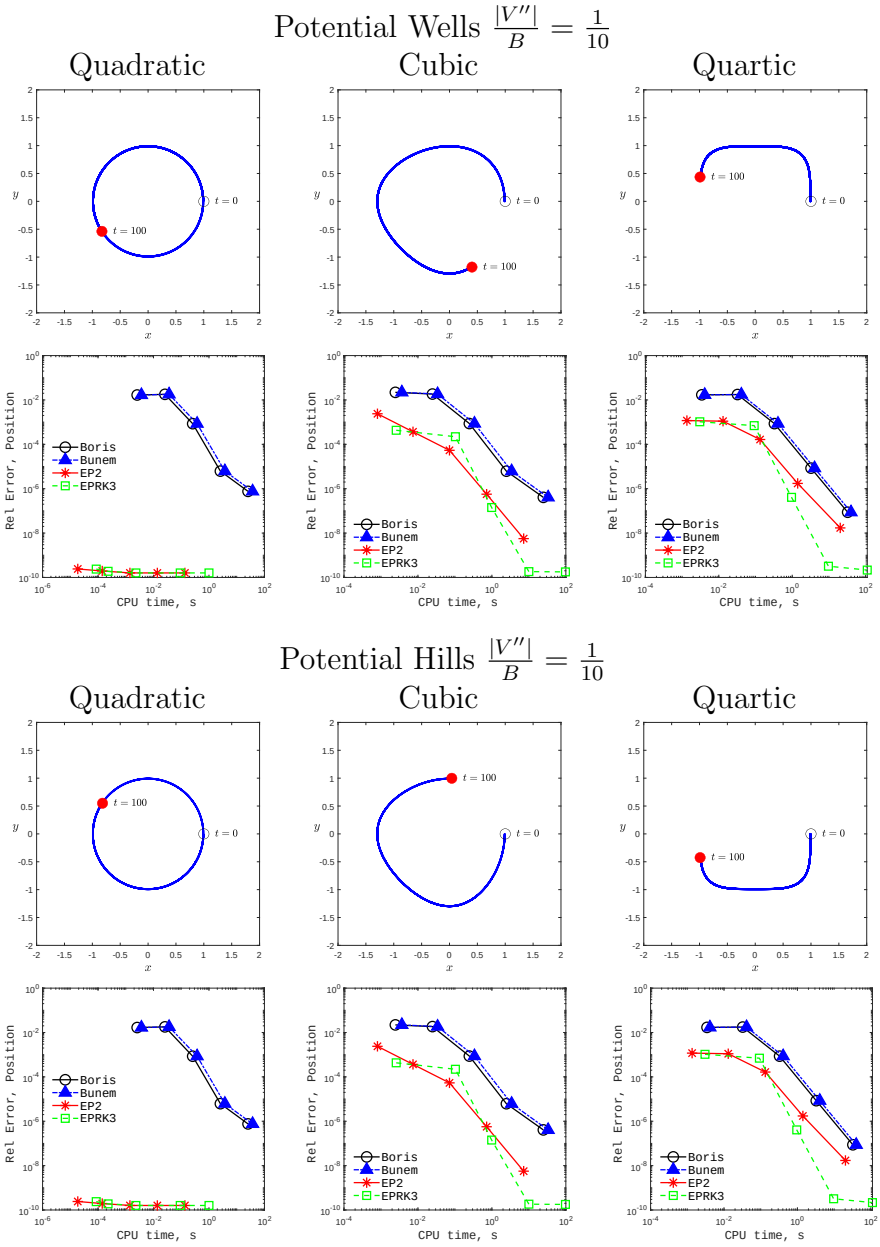


Figure 2.3: Results for 2D test problems with $|V''|/B = 1/10$: potential well reference solution orbits (first row), potential well precision diagrams (second row), potential hill reference solution orbits (third row), and potential hill precision diagrams (fourth row). Boris/Buneman step sizes are $h = 10^{-2}$, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6} for quadratic potential problems and $h = 10^{-3}$, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7} for cubic/quartic potential problems. EP2/EPRK3 step sizes are $h = 100$, 10, 1, 10^{-1}, 10^{-2} for quadratic potential problems and $h = 10^{-1}$, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} for cubic/quartic potential problems.

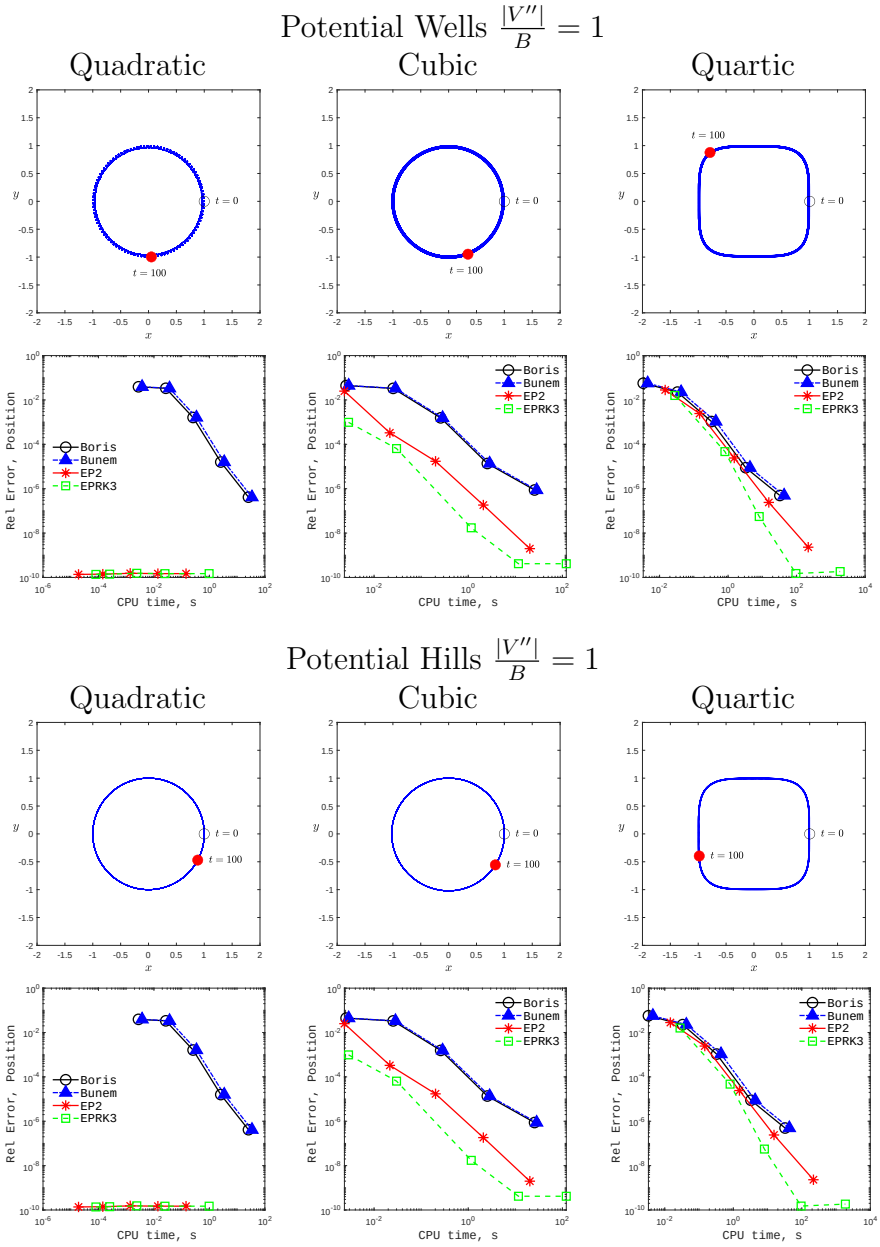
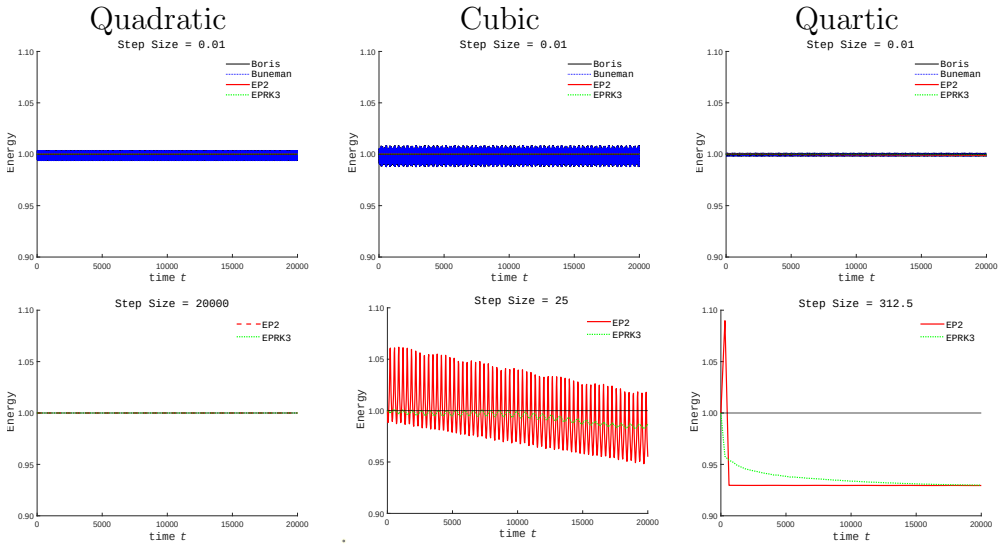


Figure 2.4: Results for 2D test problems with $|V''|/B = 1$: potential well reference solution orbits (first row), potential well precision diagrams (second row), potential hill reference solution orbits (third row), and potential hill precision diagrams (fourth row). Boris/Buneman step sizes are $h = 10^{-2}$, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6} for quadratic potential problems and $h = 10^{-3}$, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7} for cubic/quartic potential problems. EP2/EPRK3 step sizes are $h = 100$, 10, 1, 10^{-1}, 10^{-2} for quadratic potential problems and $h = 10^{-1}$, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} for cubic/quartic potential problems.

Potential Wells, $\frac{|V''|}{B} = \frac{1}{100}$



Potential Hills, $\frac{|V''|}{B} = \frac{1}{100}$

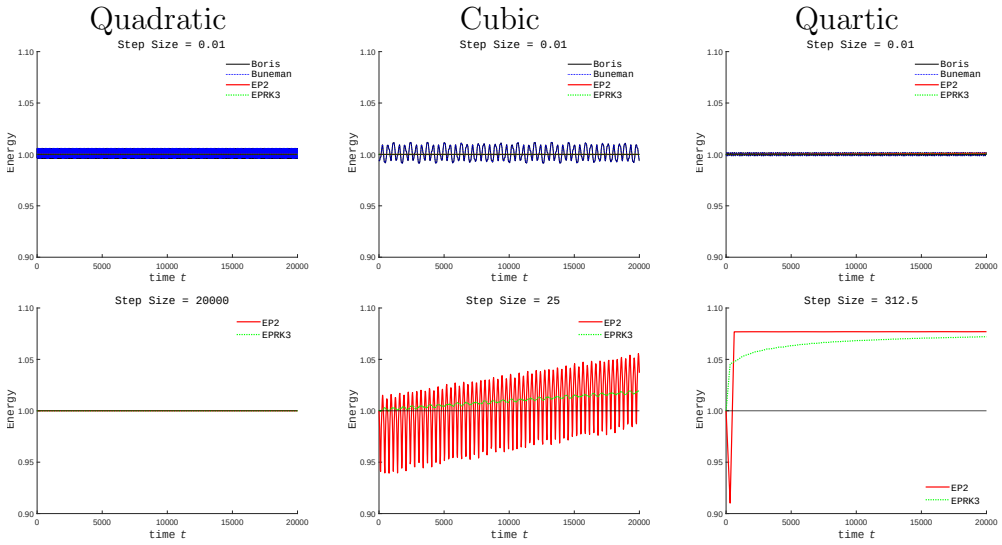
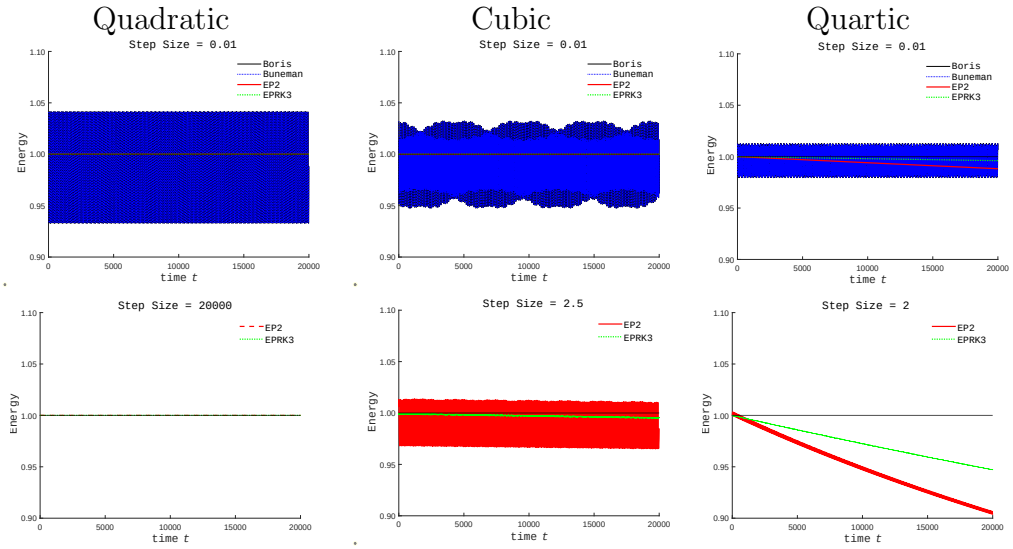


Figure 2.5: Energy of 2D test problems with $|V''|/B = 1/100$

Potential Wells, $\frac{|V''|}{B} = \frac{1}{10}$



Potential Hills, $\frac{|V''|}{B} = \frac{1}{10}$

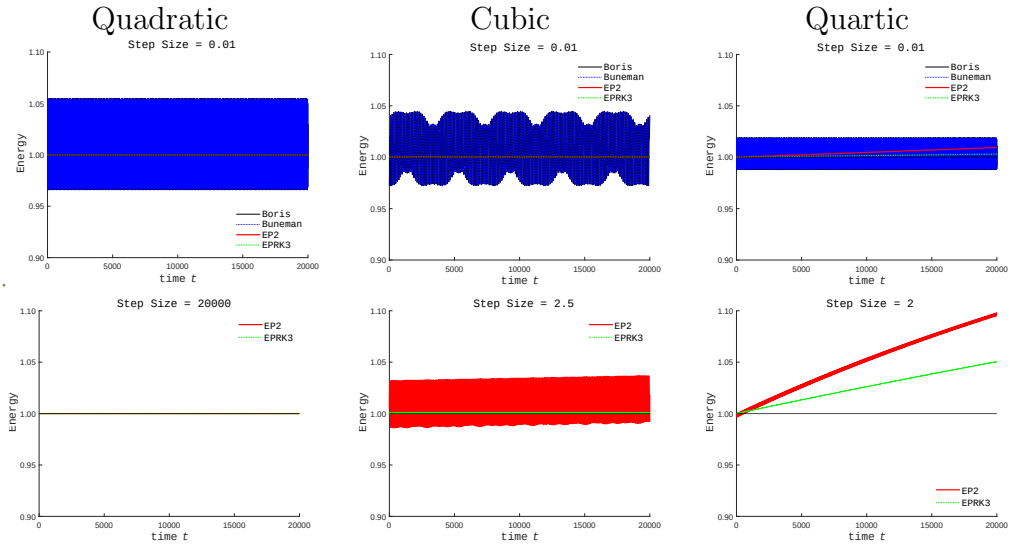


Figure 2.6: Energy of 2D test problems with $|V''|/B = 1/10$

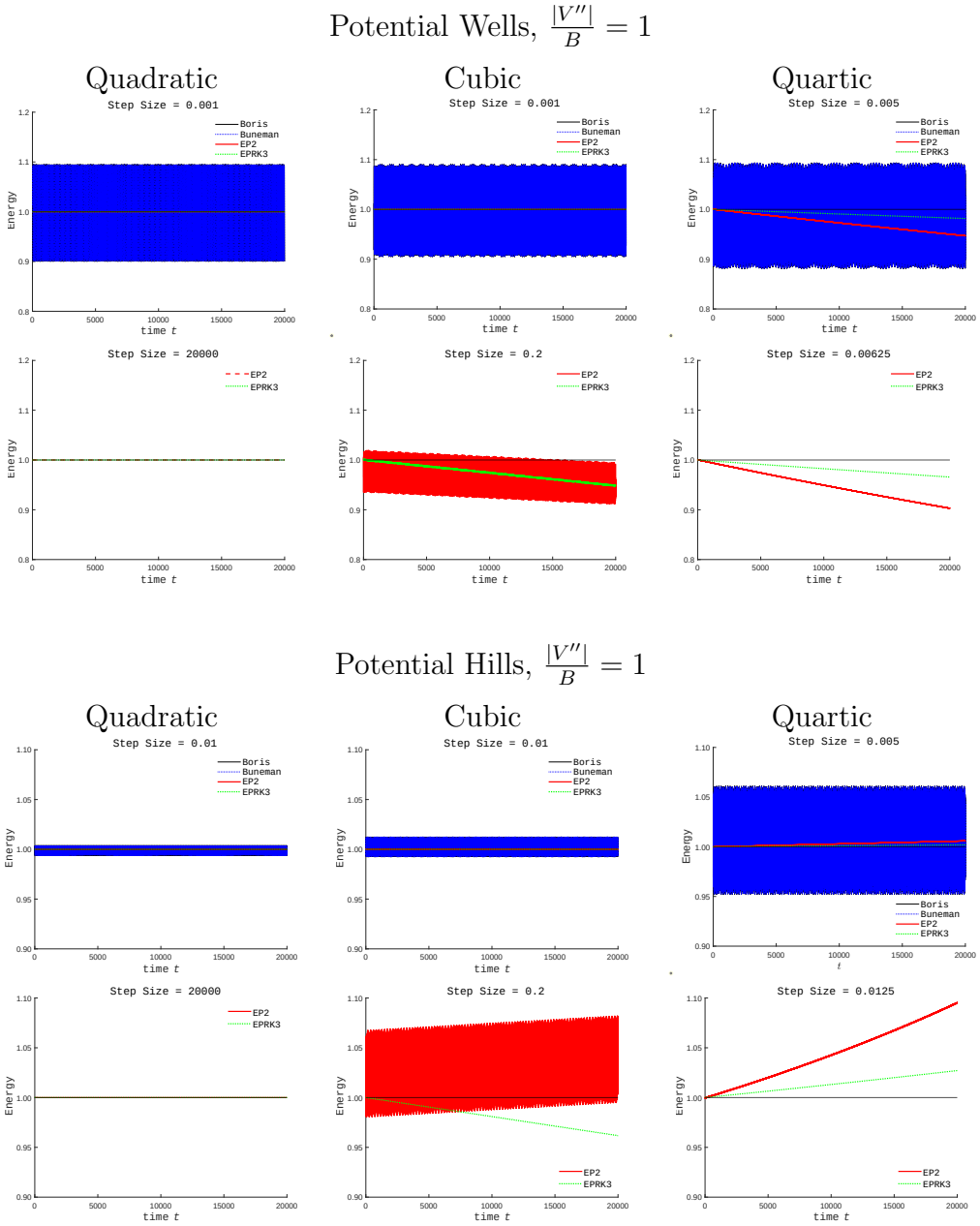


Figure 2.7: Energy of 2D test problems with $|V''|/B = 1$

Non-Uniform Magnetic Field, Grad-B Drift Problem, 2D Model

In this experiment we considered a prototype non-uniform magnetic field configuration with the so-called grad-B drift problem. This problem has a non-uniform magnetic field with linear spatial variation in which the length scale of spatial variation is of much longer than the gyroradius. Formally, we assume

$$\frac{r\|\nabla B\|}{B} \ll 1,$$

where r is the gyroradius, $B = \|\boldsymbol{B}\|$, and ∇B is the magnetic field gradient. Under this assumption, the particle experiences a drift velocity [9,23,30] approximately given by

$$\boldsymbol{v}_{\nabla B} = \frac{1}{2}\frac{v_\perp^2}{\omega}\frac{\boldsymbol{B}\times\nabla B}{B^2},$$

where $v_\perp$ is the particle speed in the plane perpendicular to the magnetic field and $\omega = qB/m$ is the gyrofrequency.

The test problem for this experiment was configured with zero electric field and magnetic field set to

$$\boldsymbol{B} = (100 + \delta B\, y)\hat{\boldsymbol{z}}.$$

Similar to the previous experiments in this section, the grad-B drift experiment considers a particle of mass $m = 1$ and charge $q = 1$ with initial conditions $\boldsymbol{x}_0 = (1, 0)$ and $\boldsymbol{v}_0 = (0, -1)$. Solutions were obtained by integrating the equations of motion over the time interval $[0, 100]$. Figure 3.3 shows plots of the reference solution orbits and the precision diagrams for $\delta B = 0.1$, 1, 10.

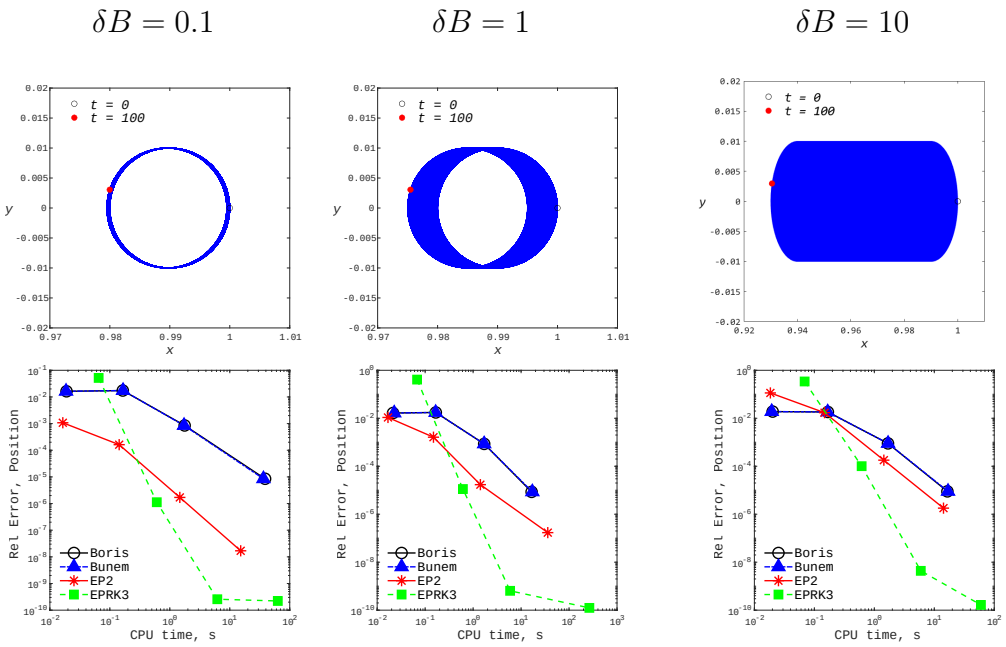


Figure 2.8: Results for grad-B drift problem: reference solution orbits (top row), and precision diagrams (bottom row). Boris/Buneman step sizes are $h = 10^{-2}$, 10^{-3}, 10^{-4}, 10^{-5}. EP2/EPRK3 step sizes are $h = 10^{-1}$, 10^{-2}, 10^{-3}, 10^{-4}.

Gyroradius, 2D Model

This experiment examines the gyroradius of the solutions computed by the numerical particle pushers. Since the Boris algorithm is known to compute an artificially enlarged gyroradius when using large step sizes relative to the gyroperiod [33], it is of interest to see how the exponential integrators perform in this regard. Here, the term "large step size relative to the gyroperiod" (or simply "large" step size) is defined by $\omega h \gg 1$, where $\omega = |q|B/m$ is the gyroperiod and h is the time step size. Conversely, the term "small step size relative to the gyroperiod" (or simply "small" step size) is defined by $\omega h \ll 1$. Here we consider a linear $\boldsymbol{E} \times \boldsymbol{B}$ drift problem with electromagnetic fields

$$\boldsymbol{B} = 100\,\hat{\boldsymbol{z}} \quad \text{and} \quad \boldsymbol{E} = -\begin{bmatrix} 0 \\ 1+y \end{bmatrix}.$$

The gyroradius is $r = 0.01$ for this particular configuration. Using a particle of mass $m = 1$ and charge $q = 1$ with initial conditions $\boldsymbol{x}_0 = (1, 0)$ and $\boldsymbol{v}_0 = (0, -1)$, we integrated the equations of motion over the time interval $[0, 100]$ using a "small" step size $h = 0.001$ and a "large" step size $h = 0.1$. For the "small" and "large" step sizes, these yield $\omega h = 0.1 < 1$ and $\omega h = 10 > 1$, respectively. Results of the experiment are shown in figure 3.2. Observe that all the numerical particle pushers accurately computed the correct gyroradius for the "small" step size $h = 0.001$. However, for the "large" step size $h = 0.1$ both Boris and Buneman algorithms compute a drastically enlarged gyroradius while the exponential integrators compute the correct gyroradius.

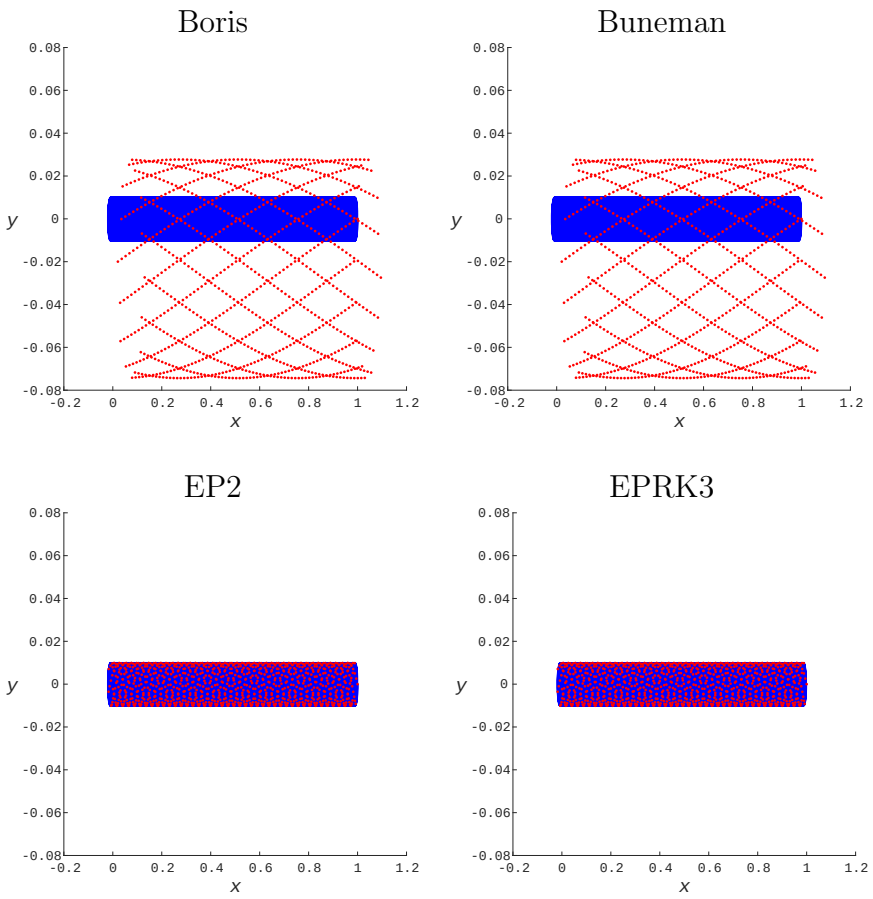


Figure 2.9: Plots of computed trajectories for the $\boldsymbol{E} \times \boldsymbol{B}$ drift problem. Solutions for step size $h = 0.001$ are solid blue and solutions for step size $h = 0.1$ are dotted red.

2.4.2 Three Dimensional Model

All three-dimensional test problems set the initial particle position at $\boldsymbol{x}_0 = (1, 0, 0)$ and the initial particle velocity at $\boldsymbol{v}_0 = (0, -1, 1)$. Configurations for the electric scalar potential wells and their corresponding electric fields are shown in table 2.3. Configurations for the electric scalar potential hills and the corresponding electric fields are shown in table 2.4.

Figures 2.10, 2.11, and 2.12 show plots of the reference solution orbits and precision diagrams for test problems with $|V''|/B = 1/100$, $|V''|/B = 1/10$, and $|V''|/B = 1$, respectively. Note that overall the comparative performance of the exponential methods with traditional particle pushers is similar for three-dimensional problems compared to two-dimensional cases. As in the two-dimensional experiments the exponential methods perform well in the linear case and remain competitive for cubic and quartic potentials. A minor difference for the linear case performance between two-dimensional and three-dimensional is in the slight increase of the error for the largest steps sizes. This is the result of finite precision computations of large analytic formulas involved in evaluation of the eigenvalues of the Jacobian matrix and the polynomials in the Lagrange-Sylvester formula. This error can be reduced or eliminated, if needed, if the calculations are performed using software packages that double the precision of the calculations.

The energy plots for the test problems with $|V''|/B = 1/100$, $|V''|/B = 1/10$, and $|V''|/B = 1$ are shown in figures 2.13, 2.14, and 2.15 are also similar to the two-dimensional case. As in the two-dimensional experiments, the accuracy of the energies of the system computed with the exponential integrators drifts over long time intervals and the magnitude of the drift depends on the time step size. Again, the EPRK3 integrator performs better than the EP2 integrator by exhibiting both less drift and less variation in the computed energies indicating that higher order methods indeed yield more accurate solutions. Thus, comparative performance of all methods is consistent across two- and three-dimensional problems and the numerical results are aligned with theoretically expected performance.

		$\lvert V''\rvert = 1$	$\lvert V''\rvert = 10$	$\lvert V''\rvert = 100$
Quadratic Well	V	$\frac{1}{2}(x^2+y^2)$ $+\frac{1}{20}z^2 - \frac{1}{2}$	$5(x^2+y^2)$ $+\frac{1}{2}z^2 - 5$	$50(x^2+y^2)$ $+5z^2 - 50$
	$\boldsymbol{E}$	$-\begin{bmatrix} x \\ y \\ \frac{1}{10}z \end{bmatrix}$	$-\begin{bmatrix} 10x \\ 10y \\ z \end{bmatrix}$	$-\begin{bmatrix} 100x \\ 100y \\ 10z \end{bmatrix}$
Cubic Well	V	$-\frac{5}{6} + x^2 + y^2$ $-\frac{1}{6}(x^3+y^3)$ $+\frac{1}{10}z^2 - \frac{1}{60}z^3$	$-\frac{11}{3} + 3(x^2+y^2)$ $+\frac{2}{3}(x^3+y^3)$ $+\frac{3}{10}z^2 + \frac{1}{15}z^3$	$47(x^2+y^2)$ $-48 + x^3 + y^3$ $+\frac{1}{10}(47z^2+z^3)$
	$\boldsymbol{E}$	$-\begin{bmatrix} 2x - \frac{1}{2}x^2 \\ 2y - \frac{1}{2}y^2 \\ \frac{1}{5}z - \frac{1}{20}z^2 \end{bmatrix}$	$-\begin{bmatrix} 6x - 2x^2 \\ 6y - 2y^2 \\ \frac{3}{5}z + \frac{1}{5}z^2 \end{bmatrix}$	$-\begin{bmatrix} 94x + 3x^2 \\ 94y + 3y^2 \\ \frac{47}{5}z + \frac{3}{10}z^2 \end{bmatrix}$
Quartic Well	V	$\frac{1}{12}(x^4+y^4)$ $+\frac{1}{120}z^4 - \frac{1}{12}$	$\frac{5}{6}(x^4+y^4)$ $+\frac{1}{12}z^4 - \frac{5}{6}$	$\frac{25}{3}(x^4+y^4)$ $+\frac{5}{6}z^4 - \frac{2}{3}$
	$\boldsymbol{E}$	$-\frac{1}{3}\begin{bmatrix} x^3 \\ y^3 \\ \frac{1}{10}z^3 \end{bmatrix}$	$-\frac{1}{3}\begin{bmatrix} 10x^3 \\ 10y^3 \\ z^3 \end{bmatrix}$	$-\frac{1}{3}\begin{bmatrix} 100x^3 \\ 100y^3 \\ 10z^3 \end{bmatrix}$

Table 2.3: Electric scalar potential wells and corresponding electric fields for 3D model test problems

		$\lvert V''\rvert = 1$	$\lvert V''\rvert = 10$	$\lvert V''\rvert = 100$
Quadratic Hill	V	$-\frac{1}{2}(x^2+y^2)$ $+\frac{1}{20}z^2+\frac{1}{2}$	$-5(x^2+y^2)$ $+\frac{1}{2}z^2+5$	$-50(x^2+y^2)$ $+5z^2+50$
	$\boldsymbol{E}$	$\begin{bmatrix} x \\ y \\ \frac{1}{10}z \end{bmatrix}$	$\begin{bmatrix} 10x \\ 10y \\ z \end{bmatrix}$	$\begin{bmatrix} 100x \\ 100y \\ 10z \end{bmatrix}$
Cubic Hill	V	$\frac{5}{6}+x^2+y^2$ $+\frac{1}{6}(x^3+y^3)$ $+\frac{1}{10}z^2-\frac{1}{60}z^3$	$\frac{11}{3}-3(x^2+y^2)$ $-\frac{2}{3}(x^3+y^3)$ $+\frac{3}{10}z^2+\frac{1}{15}z^3$	$-47(x^2+y^2)$ $+48-x^3-y^3$ $+\frac{1}{10}(47z^2+z^3)$
	$\boldsymbol{E}$	$\begin{bmatrix} 2x-\frac{1}{2}x^2 \\ 2y-\frac{1}{2}y^2 \\ -\frac{1}{5}z+\frac{1}{20}z^2 \end{bmatrix}$	$\begin{bmatrix} 6x-2x^2 \\ 6y-2y^2 \\ -\frac{3}{5}z-\frac{1}{5}z^2 \end{bmatrix}$	$\begin{bmatrix} 94x+3x^2 \\ 94y+3y^2 \\ -\frac{47}{5}z-\frac{3}{10}z^2 \end{bmatrix}$
Quartic Hill	V	$-\frac{1}{12}(x^4+y^4)$ $+\frac{1}{120}z^4+\frac{1}{12}$	$-\frac{5}{6}(x^4+y^4)$ $+\frac{1}{12}(z^4+1)$	$-\frac{25}{3}(x^4+y^4)$ $+\frac{5}{6}z^4+\frac{25}{3}$
	$\boldsymbol{E}$	$\frac{1}{3}\begin{bmatrix} x^3 \\ y^3 \\ -\frac{1}{10}z^3 \end{bmatrix}$	$\frac{1}{3}\begin{bmatrix} 10x^3 \\ 10y^3 \\ -z^3 \end{bmatrix}$	$\frac{1}{3}\begin{bmatrix} 100x^3 \\ 100y^3 \\ -10z^3 \end{bmatrix}$

Table 2.4: Electric scalar potential hills and corresponding electric fields for 3D model test problems

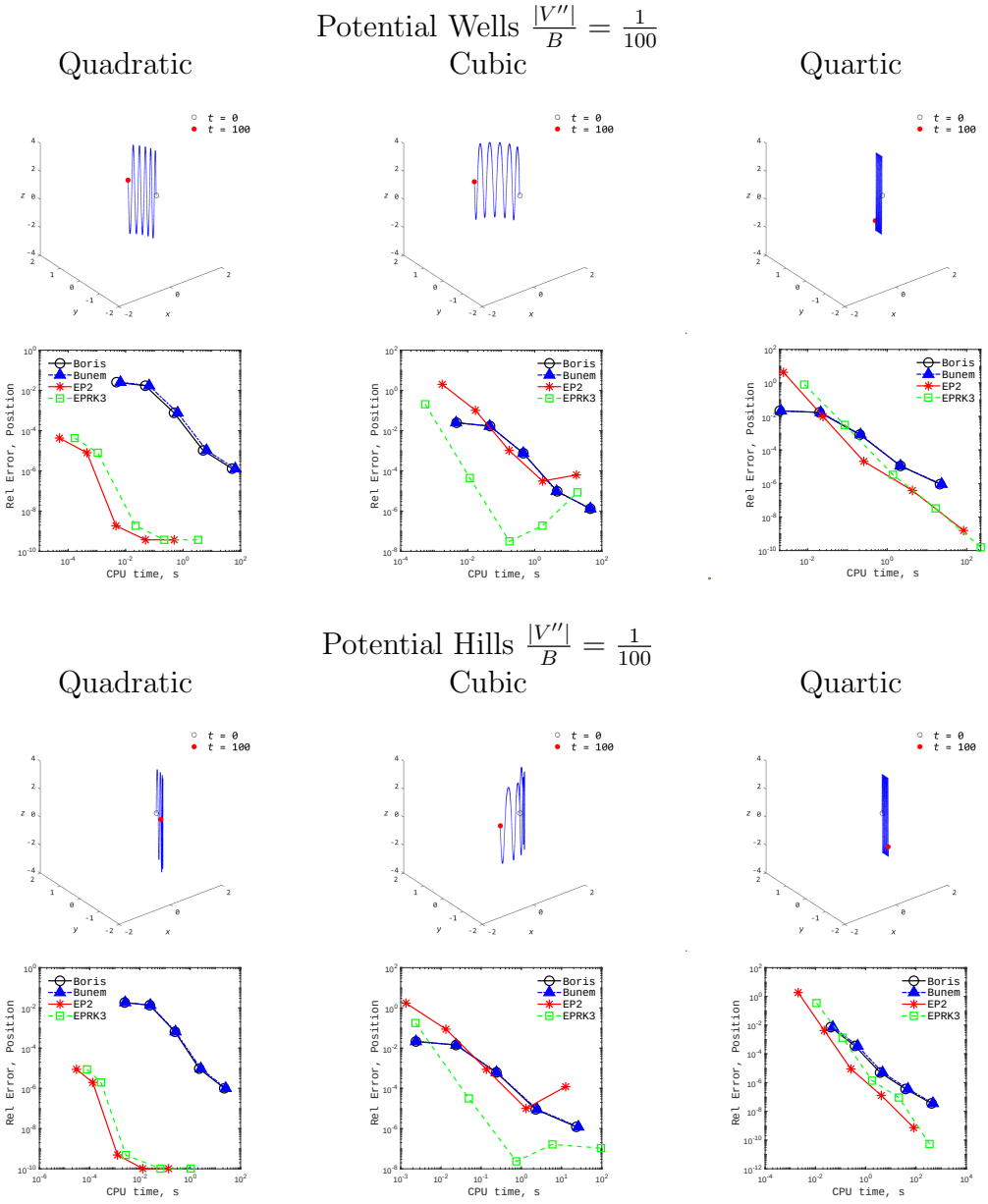


Figure 2.10: Results for 3D test problems with $|V''|/B = 1/100$: potential well reference solution orbits (first row), potential well precision diagrams (second row), potential hill reference solution orbits (third row), and potential hill precision diagrams (fourth row). Boris/Buneman step sizes are $h = 10^{-2}$, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6} for quadratic potential problems and $h = 10^{-3}$, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7} for cubic/quartic potential problems. EP2/EPRK3 step sizes are $h = 100$, 10, 1, 10^{-1}, 10^{-2} for quadratic potential problems and $h = 10^{-1}$, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} for cubic/quartic potential problems.

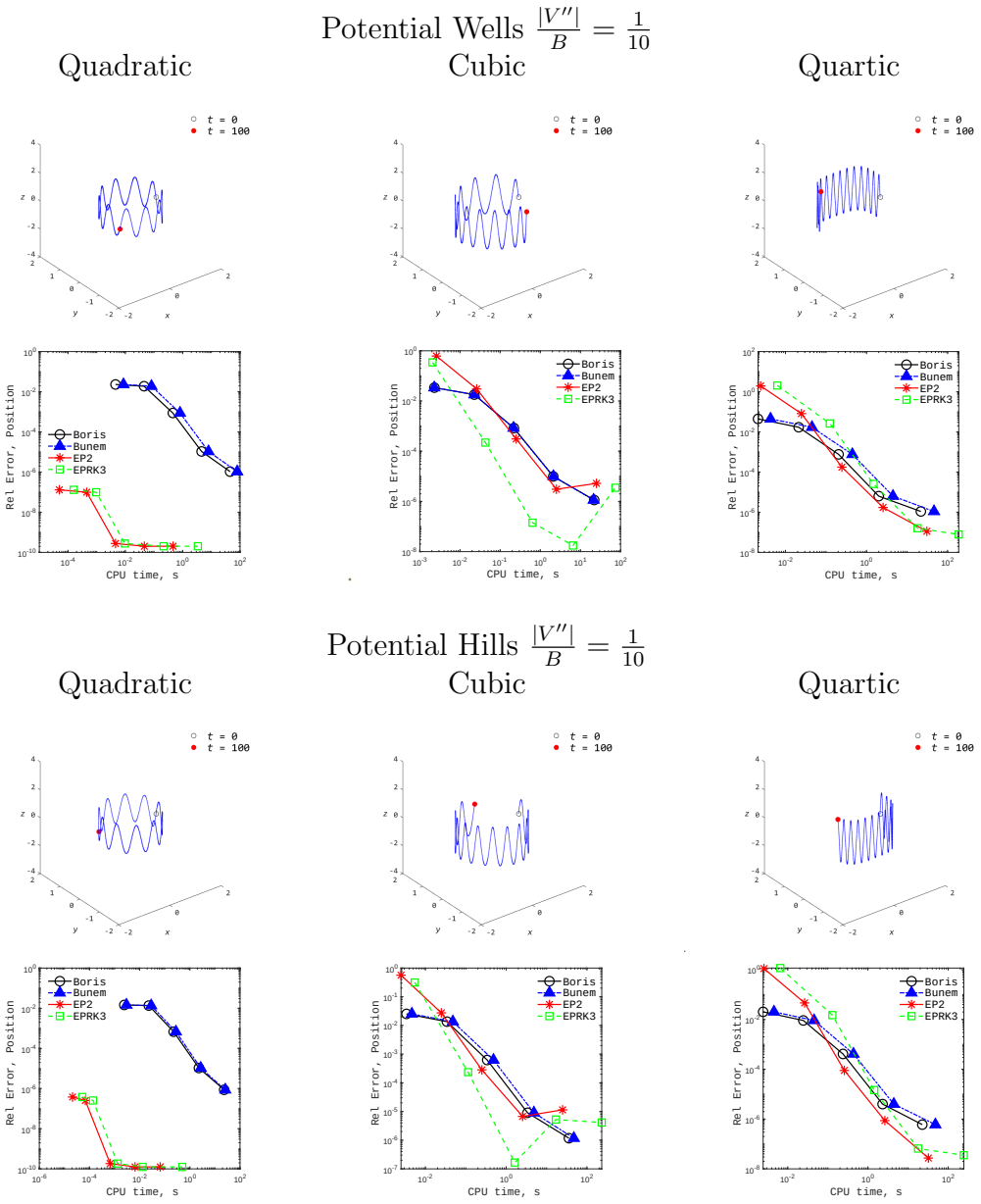


Figure 2.11: Results for 3D test problems with $|V''|/B = 1/10$: potential well reference solution orbits (first row), potential well precision diagrams (second row), potential hill reference solution orbits (third row), and potential hill precision diagrams (fourth row). Boris/Buneman step sizes are $h = 10^{-2}$, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6} for quadratic potential problems and $h = 10^{-3}$, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7} for cubic/quartic potential problems. EP2/EPRK3 step sizes are $h = 100$, 10, 1, 10^{-1}, 10^{-2} for quadratic potential problems and $h = 10^{-1}$, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} for cubic/quartic potential problems.

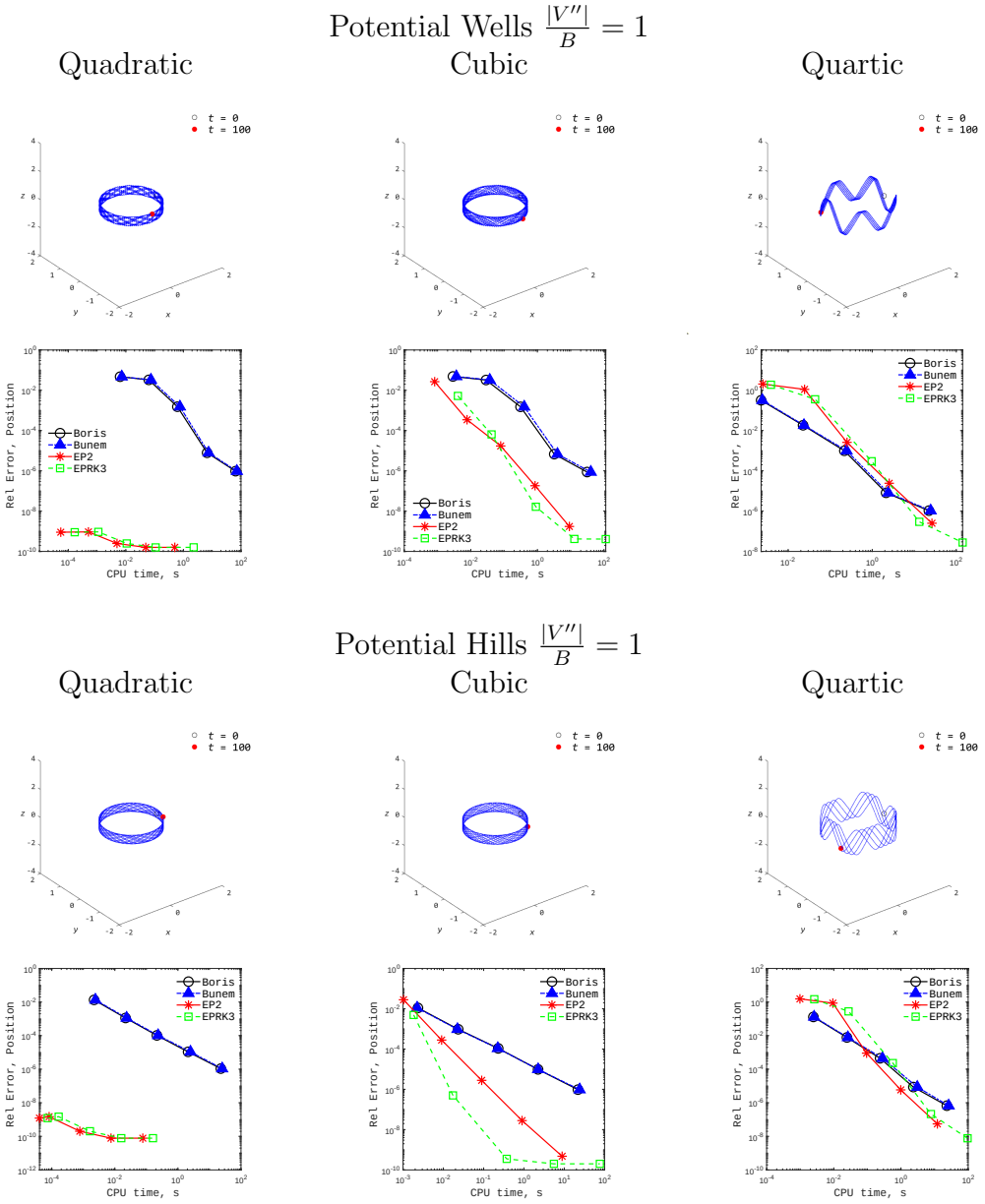


Figure 2.12: Results for 3D test problems with $|V''|/B = 1$: potential well reference solution orbits (first row), potential well precision diagrams (second row), potential hill reference solution orbits (third row), and potential hill precision diagrams (fourth row). Boris/Buneman step sizes are $h = 10^{-2}$, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6} for quadratic potential problems and $h = 10^{-3}$, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7} for cubic/quartic potential problems. EP2/EPRK3 step sizes are $h = 100$, 10, 1, 10^{-1}, 10^{-2} for quadratic potential problems and $h = 10^{-1}$, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} for cubic/quartic potential problems.

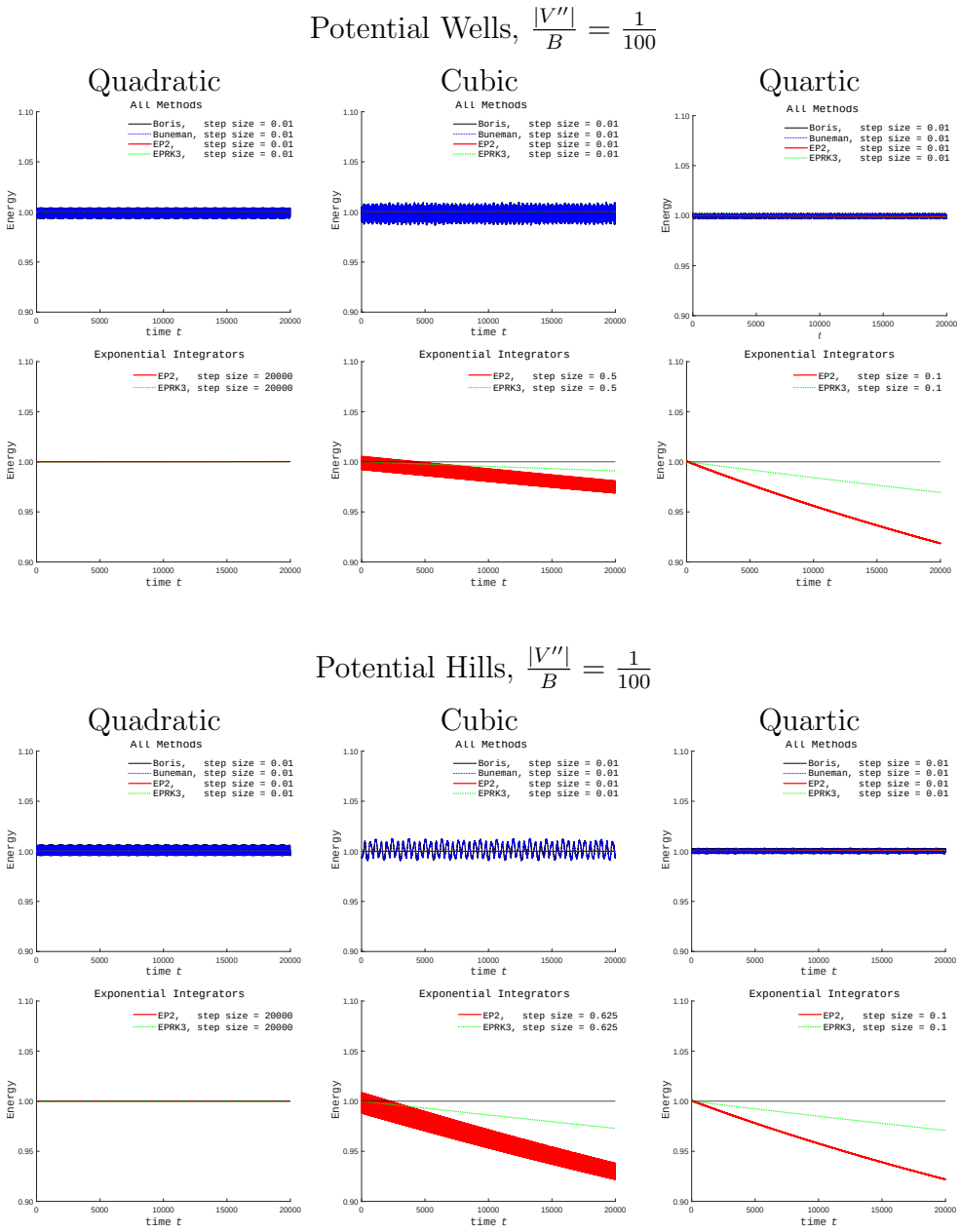


Figure 2.13: Energy of 3D test problems with $|V''|/B = 1/100$

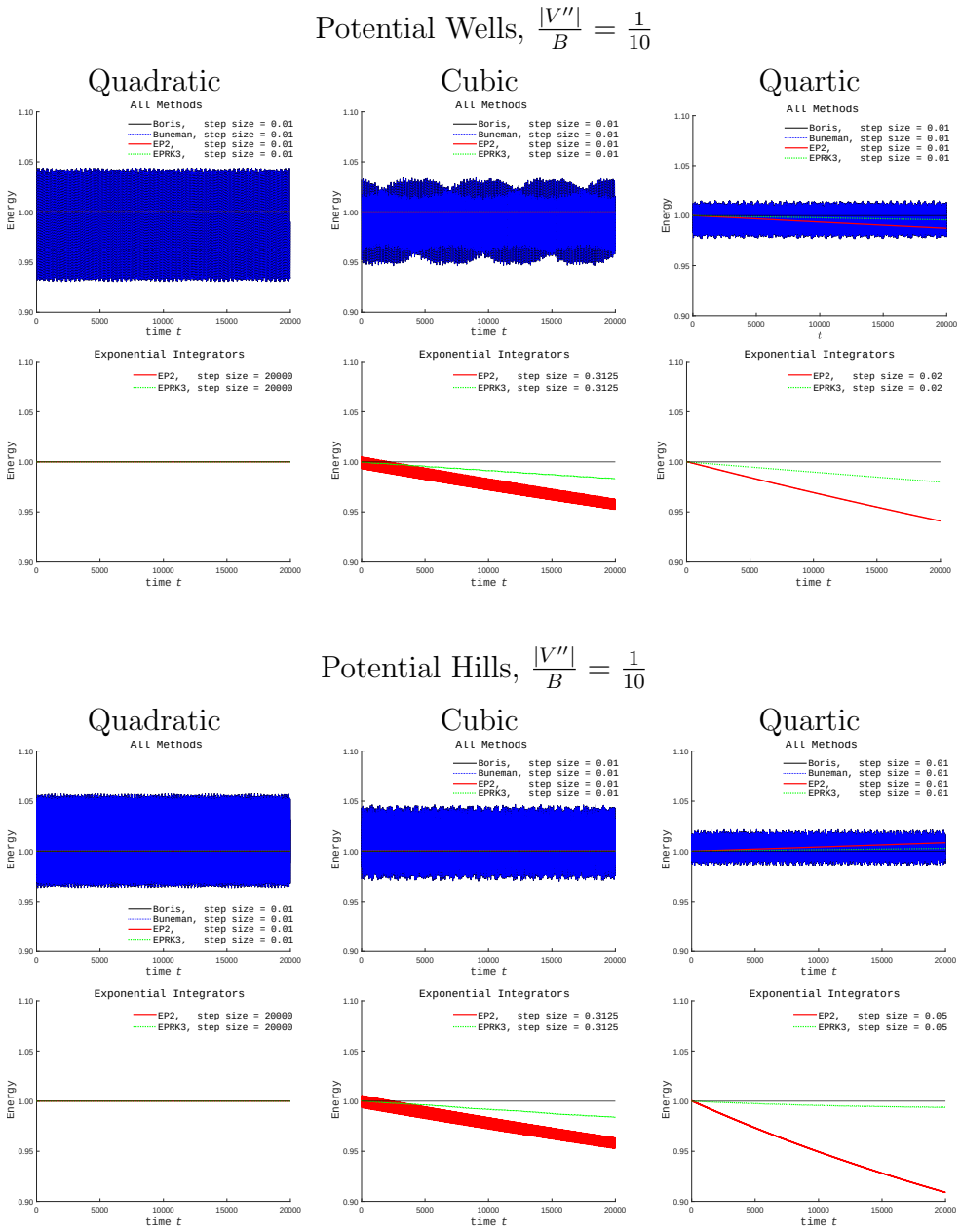
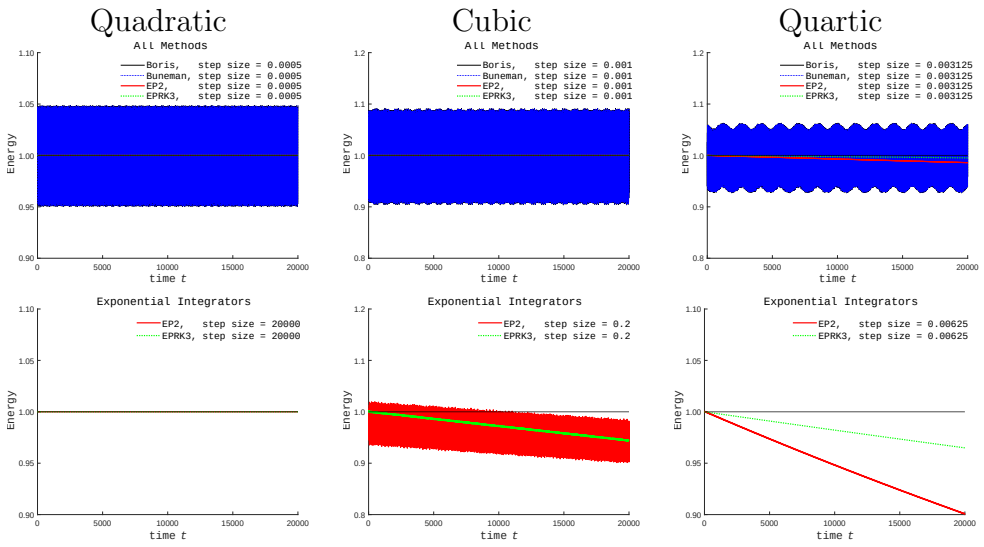


Figure 2.14: Energy of 3D test problems with $|V''|/B = 1/10$

Potential Wells, $\frac{|V''|}{B} = 1$



Potential Hills, $\frac{|V''|}{B} = 1$

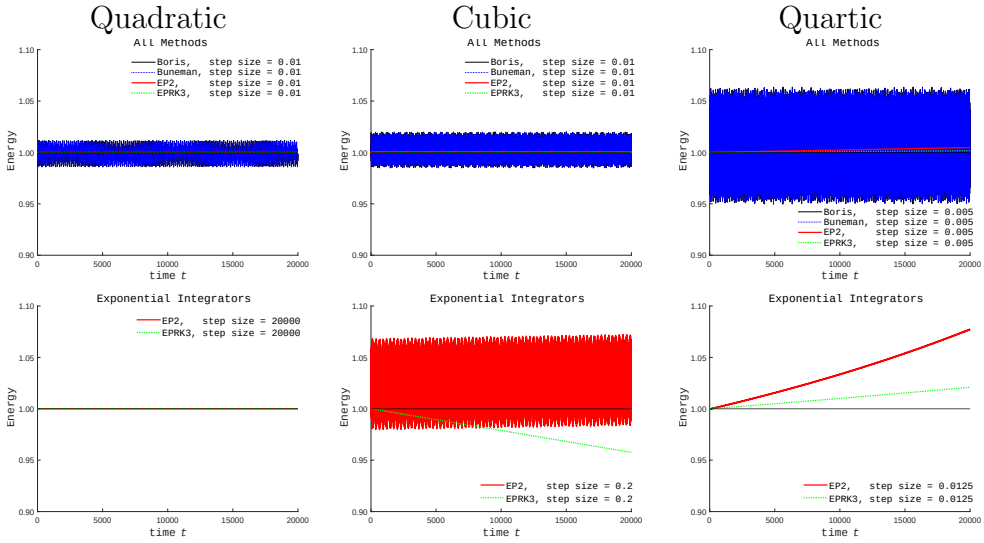


Figure 2.15: Energy of 3D test problems with $|V''|/B = 1$

2.5 Conclusion and Future Work

In this chapter we proposed an alternative approach to the numerical simulation of charged particle dynamics using exponential integrators. An integral part of this algorithm is taking advantage of the low dimensionality of the particle pushing problem and using an analytic method to compute matrix φ functions needed at each step of an exponential scheme. We showed that exponential integrators can be competitive compared to traditional particle pushers when the problem is strongly magnetized. As expected, exponential integrators offer dramatic computational advantages for cases where electric fields are generated by quadratic electric potentials. Since the problem is linear in this case, an exponential integrator with accurate evaluation of matrix φ functions computes a very accurate solution with the error coming primarily from the finite-precision computation of φ that involves the eigenvalue solver and the interpolation polynomial of the function. Compared to the traditional Boris and Buneman algorithms, for these linear problems, we showed that exponential integrators could bring approximately six orders of magnitude gains in computational speed and three orders of magnitude improvements in accuracy simultaneously. For nonlinear problems with the cubic electric potentials we still saw significant computational savings, though not as dramatic as for quadratic problems. To obtain the solution at the same accuracy level, exponential integrators exhibited savings in computation time of about two orders of magnitude for two-dimensional problems and at least an order of magnitude for three-dimensional problems compared to traditional methods. The quartic potentials yielded comparable performance between the exponential integrators and the Boris and Buneman schemes. We also see that higher order exponential methods can improve the computational performance. These points indicate that for highly nonlinear problems like those with a quartic electric potential, it is important to pay particular attention to approximation of the nonlinear integral in the exact solution (2.5) when constructing an exponential integrator. This is further evidenced by the grad-B drift experiments. Exploring different approximations of the nonlinear integral to develop better performing exponential methods for highly nonlinear problems will be one of the research directions we plan to pursue in the future.

Of course, the exponential methods we used have not been designed to be energy preserving and, indeed, a drift in energy is observed in the numerical results. However, we showed improvements in the accuracy of the computed energy as the order of an integrator is increased. This result warrants further research into development of exponential methods of higher order that would potentially exhibit better energy preservation.

Additionally, we showed that for a linear $\boldsymbol{E} \times \boldsymbol{B}$ drift problem the exponential integrators accurately compute the gyroradius regardless of the step size value as expected. By contrast, both the Boris and Buneman pushers artificially enlarge the gyroradius for a large step size relative to the gyroradius.

Investigating the performance of the exponential particle pushers as they are embedded within an overall PIC integrator is another research direction we plan to

pursue. For example, low order spatial discretizations of the electric field can result in the potentials dominated by the quadratic terms. It would be interesting to study whether the computational advantages of exponential integrators for such quadratically dominated potentials would persist even if we account for the redefining of potentials as the particles cross the cell boundaries.

To summarize, we have shown preliminary results that offer some evidence of the numerical advantages of the new numerical approach we propose. This work also highlighted possible directions for improving the exponential integration-based methods making them more suitable for highly nonlinear particle pushing problems and we plan to pursue these directions in our future research.

Chapter 3

Nyström type exponential integrators for strongly magnetized charged particle motion

3.1 Introduction

In the previous chapter, we investigated exponential integration as an alternative approach to numerical solve strongly magnetized particle pushing problems. Results from our numerical experiments demonstrate that exponential integrators yield superior performance for linear and weakly nonlinear problems and are competitive for strongly nonlinear problems when compared to the conventional Boris and Buneman schemes. Thus, exponential integration was shown to be a promising approach to solving numerically stiff particle pushing problems in strongly magnetized plasma.

This chapter builds upon the previous results and presents new integrators that allow further improvement in the computational efficiency of exponential methods for particle pushing problems. Specifically, we take the standard exponential integration framework as a starting template and derive Nyström-type methods induced by partitioning the standard exponential integrators into components corresponding to particle position $\boldsymbol{x}$ and velocity $\boldsymbol{v}$. These Nyström-type exponential integrators exploit the mathematical structure of the Newtonian formulation of particle pushing problem yielding computationally efficient methods that effectively integrates the equation of motion as a second-order problem.

The organization of this chapter is as follows. Section 2 reviews the equations describing charged particle motion in electromagnetic fields. Section 3 discusses the standard exponential integration framework. Section 4 presents our approach to deriving Nyström-type exponential integrators. Numerical results are presented in section 5. Finally, we summarize our results, present conclusions, and discuss future research.

3.2 Newtonian Equations of Motion

Recall that the dynamics of a particle of mass m and charge q moving under the Lorentz force exerted by a magnetic field $\boldsymbol{B}$ and an electric field $\boldsymbol{E}$ is given by

$$m\,\frac{d\boldsymbol{v}}{dt} = q\,(\boldsymbol{E} + \boldsymbol{v} \times \boldsymbol{B}). \tag{3.1}$$

Making the observation that velocity $\boldsymbol{v}$ is simply the time derivative of the position $\boldsymbol{x}$, the Newtonian form of the particle pushing problem is equivalently expressed by the first-order system

$$\begin{cases} \dfrac{d\boldsymbol{x}}{dt} &= \boldsymbol{v}, \\ \dfrac{d\boldsymbol{v}}{dt} &= \dfrac{q}{m}(\boldsymbol{E} + \boldsymbol{v} \times \boldsymbol{B}). \end{cases} \tag{3.2}$$

3.3 Exponential Integration Framework

This section reviews the framework to derive a standard exponential integrator. Consider an initial value problem of the form:

$$\frac{d\boldsymbol{u}}{dt} = \boldsymbol{F}(\boldsymbol{u}), \qquad \boldsymbol{u}_n = \boldsymbol{u}(t_n). \tag{3.3}$$

We take a first-order Taylor expansion of the right-hand side function about the initial condition $\boldsymbol{u}_n$ to obtain

$$\frac{d\boldsymbol{u}}{dt} = \boldsymbol{A}_n\boldsymbol{u} + \boldsymbol{r}(\boldsymbol{u}), \tag{3.4}$$

where $\boldsymbol{A}_n = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{u}}\big|_{\boldsymbol{u}=\boldsymbol{u}_n}$ is the Jacobian matrix (evaluated at $\boldsymbol{u} = \boldsymbol{u}_n$) and

$$\boldsymbol{r}(\boldsymbol{u}) = \boldsymbol{F}(\boldsymbol{u}) - \boldsymbol{F}(\boldsymbol{u}_n) - \boldsymbol{A}_n(\boldsymbol{u} - \boldsymbol{u}_n) \tag{3.5}$$

is the nonlinear remainder term. We now multiply equation (3.4) by the integrating factor $\exp(-t\boldsymbol{A}_n)$ and integrate from $t = t_n$ to $t = t_n + h$ to obtain the variation of constants formula

$$\boldsymbol{u}(t_n + h) = \boldsymbol{u}_n + h\,\varphi_1(h\boldsymbol{A}_n)\boldsymbol{F}(\boldsymbol{u}_n) + h\int_0^1 e^{h(1-\tau)\boldsymbol{A}_n}\boldsymbol{r}(\boldsymbol{u})\,d\tau, \tag{3.6}$$

where $\varphi_1(h\boldsymbol{A}_n)$ is a matrix function defined by

$$\varphi_1(z) = \frac{e^z - 1}{z} = \sum_{j=0}^{\infty} \frac{1}{(j+1)!} z^j.$$

Observe that if we let $\boldsymbol{u}_n$ denote a solution at time $t = t_n$ and let h be a specified time step size, then equation (3.6) is an exact analytic formula for the solution at

the next time step $\boldsymbol{u}(t_n + h)$. Hence, a numerical approximation to the variation of constants formula (3.6) yields an exponential integrator. For specific details on deriving exponential integrators, reference [39] discusses multistep type exponential integrators and references [39,40] describes Runge-Kutta type exponential integrators.

For the purpose of integrating the particle pushing problem, this chapter considers exponential integrators from a class of methods called the Exponential Propagation Iterative methods of Runge-Kutta type (EPIRK) [39]. The EPIRK class of exponential integrators has has shown to be more computationally efficient per time step compared to other types of exponential integrators for numerous applications including magnetohydrodynamics (MHD) [12].

In this chapter we consider two specific examples from the EPIRK class. The first integrator is the second-order Exponential Propagation method [39]

$$\boldsymbol{u}_{n+1} = \boldsymbol{u}_n + h\,\varphi_1(h\boldsymbol{A}_n)\boldsymbol{F}(\boldsymbol{u}_n). \tag{EP2}$$

The second integrator is the third-order Exponential Propagation method, Runge-Kutta type [37, 40]

$$\begin{aligned} \boldsymbol{U}_1 &= \boldsymbol{u}_n + h\,\varphi_1\left(\tfrac{3}{4}h\boldsymbol{A}_n\right)\boldsymbol{F}(\boldsymbol{u}_n), \\ \boldsymbol{R}_1 &= \boldsymbol{F}(\boldsymbol{U}_1) - \boldsymbol{F}(\boldsymbol{u}_n) - \boldsymbol{A}_n(\boldsymbol{U}_1 - \boldsymbol{u}_n), \\ \boldsymbol{u}_{n+1} &= \boldsymbol{u}_n + h\,\varphi_1(h\boldsymbol{A}_n)\boldsymbol{F}(\boldsymbol{u}_n) + 2h\,\varphi_3(h\boldsymbol{A}_n)\boldsymbol{R}_1. \end{aligned} \tag{EPRK3}$$

We conclude this section by pointing out that exponential integrators solve linear problems exactly, which makes them A-stable.

3.4 Nyström Methods

For the particular cases of mechanical systems governed by Newton's second law of motion in which the force acting on an object is proportional to the second derivative of its position, the governing equation of motion is given by a second-order initial value problem of the form:

$$\begin{cases} \boldsymbol{x}''(t) = \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{x}'\right), \\ \boldsymbol{x}(t_0) = \boldsymbol{x}_0, \qquad \boldsymbol{x}'(t_0) = \boldsymbol{x}_0', \end{cases}$$

where the prime notation denotes the derivative. Conventional numerical ODE solvers typically integrate first-order initial value problems. Therefore, applying these conventional methods to solve second-order problems requires transforming them to an equivalent first-order system expressed by

$$\frac{d}{dt}\begin{bmatrix}\boldsymbol{x} \\ \boldsymbol{x}'\end{bmatrix} = \begin{bmatrix}\boldsymbol{x}' \\ \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{x}')\end{bmatrix}, \qquad \begin{bmatrix}\boldsymbol{x}(t_0) \\ \boldsymbol{x}'(t_0)\end{bmatrix} = \begin{bmatrix}\boldsymbol{x}_0 \\ \boldsymbol{x}_0'\end{bmatrix},$$

However, Nyström [32] discovered a computationally efficient approach to construct integrators that solve second-order problems directly. These methods are accordingly called Nyström methods.

3.4.1 Nyström Type Exponential Integrators

Following Nyström's approach, we exploit the mathematical structure of the Newtonian formulation of the particle pushing problem and derive a Nyström-type exponential integrator. In particular, we employ the idea of using partitioned Runge-Kutta methods to induce Runge-Kutta-Nyström (RKN) integrators [20, 27, 36] and adopt this approach to derive Nyström-type exponential integrators induced by the partitioned exponential schemes.

We start by defining the function

$$\boldsymbol{f}_L = \boldsymbol{\Omega}\,\boldsymbol{v} + \frac{q}{m}\boldsymbol{E}, \tag{3.7}$$

where $\boldsymbol{\Omega}$ is the skew-symmetric matrix such that

$$\boldsymbol{\Omega}\,\boldsymbol{v} = \frac{q}{m}\boldsymbol{v} \times \boldsymbol{B}.$$

Then equation (3.2) in the context of a particle pushing problem becomes

$$\begin{cases} \dfrac{d\boldsymbol{x}}{dt} = \boldsymbol{v}, \\ \dfrac{d\boldsymbol{v}}{dt} = \boldsymbol{f}_L(\boldsymbol{x}, \boldsymbol{v}). \end{cases}$$

Next we define the following vectors:

$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{v} \end{bmatrix}, \quad \boldsymbol{F}(\boldsymbol{u}) = \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{f}_L(\boldsymbol{x}, \boldsymbol{v}) \end{bmatrix}, \quad \text{and} \quad \boldsymbol{u}_0 = \begin{bmatrix} \boldsymbol{x}(t_0) \\ \boldsymbol{v}(t_0) \end{bmatrix}.$$

This allows us to express the particle pushing problem in the form given by equation (2.1).

We now partition the standard exponential integration framework to derive a partitioned exponential integrator scheme. Partitioning the vectors of the problem into $\boldsymbol{x}$ and $\boldsymbol{v}$ components gives us

$$\boldsymbol{u} = \left[\begin{array}{c} \boldsymbol{x} \\ \hdashline \boldsymbol{v} \end{array}\right], \qquad \boldsymbol{F}(\boldsymbol{u}) = \left[\begin{array}{c} \boldsymbol{v} \\ \hdashline \boldsymbol{f}_L(\boldsymbol{x}, \boldsymbol{v}) \end{array}\right], \qquad \text{and} \qquad \boldsymbol{r} = \left[\begin{array}{c} \boldsymbol{r}_x \\ \hdashline \boldsymbol{r}_v \end{array}\right].$$

Likewise, we partition the matrices of the problem into $d \times d$ block components corresponding to $\boldsymbol{x}$ and $\boldsymbol{v}$, where d is the dimension of $\boldsymbol{x}$ and $\boldsymbol{v}$:

$$\boldsymbol{A} = \left[\begin{array}{c:c} \boldsymbol{O} & \boldsymbol{I} \\ \hdashline \boldsymbol{H} & \boldsymbol{\Omega} \end{array}\right], \qquad \boldsymbol{\varphi}_k(\boldsymbol{A}) = \left[\begin{array}{c:c} \overline{\boldsymbol{\Psi}}_k(\boldsymbol{A}) & \overline{\boldsymbol{\Upsilon}}_k(\boldsymbol{A}) \\ \hdashline \underline{\boldsymbol{\Psi}}_k(\boldsymbol{A}) & \underline{\boldsymbol{\Upsilon}}_k(\boldsymbol{A}) \end{array}\right].$$

Here $\boldsymbol{O}$ and $\boldsymbol{I}$ are the zero and identity matrices, respectively; $\boldsymbol{H} = \partial \boldsymbol{f}_L / \partial \boldsymbol{x}$ is the Jacobian matrix of $\boldsymbol{f}_L$ with respect to particle position $\boldsymbol{x}$; and $\overline{\boldsymbol{\Psi}}_k$, $\overline{\boldsymbol{\Upsilon}}_k$, $\underline{\boldsymbol{\Psi}}_k$, and $\underline{\boldsymbol{\Upsilon}}_k$

are block matrix functions. Expressing the variation-of-constants formula (3.6) in vector form, we have

$$\begin{aligned}\begin{bmatrix}\boldsymbol{x}(t_n+h)\\ \boldsymbol{v}(t_n+h)\end{bmatrix} &= \begin{bmatrix}\boldsymbol{x}_n\\ \boldsymbol{v}_n\end{bmatrix} + h\begin{bmatrix}\overline{\boldsymbol{\Psi}}_1(h\boldsymbol{A}_n) & \overline{\boldsymbol{\Upsilon}}_1(h\boldsymbol{A}_n)\\ \underline{\boldsymbol{\Psi}}_1(h\boldsymbol{A}_n) & \underline{\boldsymbol{\Upsilon}}_1(h\boldsymbol{A}_n)\end{bmatrix}\begin{bmatrix}\boldsymbol{v}_n\\ \boldsymbol{f}_L(\boldsymbol{x}_n,\boldsymbol{v}_n)\end{bmatrix}\\ &\quad + h\int_0^1 \begin{bmatrix}\overline{\boldsymbol{\Psi}}_0(h(1-\tau)\boldsymbol{A}_n) & \overline{\boldsymbol{\Upsilon}}_0(h(1-\tau)\boldsymbol{A}_n)\\ \underline{\boldsymbol{\Psi}}_0(h(1-\tau)\boldsymbol{A}_n) & \underline{\boldsymbol{\Upsilon}}_0(h(1-\tau)\boldsymbol{A}_n)\end{bmatrix}\begin{bmatrix}\boldsymbol{r}_x\\ \boldsymbol{r}_v\end{bmatrix} d\tau,\end{aligned}$$

where

$$\begin{bmatrix}\overline{\boldsymbol{\Psi}}_0(h(1-\tau)\boldsymbol{A}_n) & \overline{\boldsymbol{\Upsilon}}_0(h(1-\tau)\boldsymbol{A}_n)\\ \underline{\boldsymbol{\Psi}}_0(h(1-\tau)\boldsymbol{A}_n) & \underline{\boldsymbol{\Upsilon}}_0(h(1-\tau)\boldsymbol{A}_n)\end{bmatrix} = \varphi_0(h(1-\tau)\boldsymbol{A}_n) \coloneqq \exp(h(1-\tau)\boldsymbol{A}_n).$$

In other words, the analytic solutions for $\boldsymbol{x}$ and $\boldsymbol{v}$ at time $t = t_0 + h$ are:

$$\begin{aligned}\boldsymbol{x}(t_n+h) &= \boldsymbol{x}_n + h\overline{\boldsymbol{\Psi}}_1(h\boldsymbol{A}_n)\boldsymbol{v}_n + h\overline{\boldsymbol{\Upsilon}}_1(h\boldsymbol{A}_n)\boldsymbol{f}_L(\boldsymbol{x}_n,\boldsymbol{v}_n)\\ &\quad + h\int_0^1 \overline{\boldsymbol{\Psi}}_0(h\boldsymbol{A}_n)\boldsymbol{r}_x\,d\tau + h\int_0^1 \overline{\boldsymbol{\Upsilon}}_0(h\boldsymbol{A}_n)\boldsymbol{r}_v\,d\tau,\end{aligned} \tag{3.8a}$$

$$\begin{aligned}\boldsymbol{v}(t_n+h) &= \boldsymbol{v}_n + h\underline{\boldsymbol{\Psi}}_1(h\boldsymbol{A}_n)\boldsymbol{v}_n + h\underline{\boldsymbol{\Upsilon}}_1(h\boldsymbol{A}_n)\boldsymbol{f}_L(\boldsymbol{x}_n,\boldsymbol{v}_n)\\ &\quad + h\int_0^1 \underline{\boldsymbol{\Psi}}_0(h\boldsymbol{A}_n)\boldsymbol{r}_x\,d\tau + h\int_0^1 \underline{\boldsymbol{\Upsilon}}_n(h\boldsymbol{A}_n)\boldsymbol{r}_v\,d\tau.\end{aligned} \tag{3.8b}$$

Similar to deriving a standard exponential integrator, we now let $\boldsymbol{x}_n$ and $\boldsymbol{v}_n$ denote the numerical solutions at time $t = t_n$ for position and velocity, respectively, and let h be a specified time step size. Then applying appropriate quadrature rules to the nonlinear integral terms in (3.8) gives us numerical approximations to the solutions $\boldsymbol{x}$ and $\boldsymbol{v}$ at the next time step. In other words, we derive Nyström-type exponential integrators induced by partitioning the standard exponential integration framework.

Since we already have formulas for second-order and third-order exponential integrators, we can readily derive schemes for second-order and third-order Runge-Kutta-Nyström-type exponential integrators. Decomposing the second-order EP2 method into $\boldsymbol{x}$ and $\boldsymbol{v}$ components gives the second-order Nyström type exponential integrator EPRKN2 particle pusher:

$$\begin{aligned}\boldsymbol{x}_{n+1} &= \boldsymbol{x}_n + h\,\overline{\boldsymbol{\Psi}}_1(h\boldsymbol{A}_n)\boldsymbol{v}_n + h\,\overline{\boldsymbol{\Upsilon}}_1(h\boldsymbol{A}_n)\boldsymbol{f}_L(\boldsymbol{x}_n,\boldsymbol{v}_n),\\ \boldsymbol{v}_{n+1} &= \boldsymbol{v}_n + h\,\underline{\boldsymbol{\Psi}}_1(h\boldsymbol{A}_n)\boldsymbol{v}_n + h\,\underline{\boldsymbol{\Upsilon}}_1(h\boldsymbol{A}_n)\boldsymbol{f}_L(\boldsymbol{x}_n,\boldsymbol{v}_n)\end{aligned} \tag{EPRNK2}$$

Likewise, decomposing the third-order EPRK3 exponential integrator into $\boldsymbol{x}$ and $\boldsymbol{v}$ components gives us the third-order Runge-Kutta-Nyström type exponential integra-

tor particle pusher:

$$\begin{aligned}
\boldsymbol{X}_1 &= \boldsymbol{x}_n + h\,\overline{\boldsymbol{\Psi}}_1\left(\frac{3}{4}h\boldsymbol{A}_n\right)\boldsymbol{v}_n + h\,\overline{\boldsymbol{\Upsilon}}_1\left(\frac{3}{4}h\boldsymbol{A}_n\right)\boldsymbol{f}_L(\boldsymbol{x},\boldsymbol{v}),\\
\boldsymbol{V}_1 &= \boldsymbol{v}_n + h\,\underline{\boldsymbol{\Psi}}_1\left(\frac{3}{4}h\boldsymbol{A}_n\right)\boldsymbol{v}_n + h\,\underline{\boldsymbol{\Upsilon}}_1\left(\frac{3}{4}h\boldsymbol{A}_n\right)\boldsymbol{f}_L(\boldsymbol{x},\boldsymbol{v}),\\
\boldsymbol{R}_v &= (\boldsymbol{\Omega}_1 - \boldsymbol{\Omega}_n)\,\boldsymbol{V}_1 + \frac{q}{m}\big(\boldsymbol{E}(\boldsymbol{X}_1) - \boldsymbol{E}(\boldsymbol{x}_n)\big) - \boldsymbol{H}_n(\boldsymbol{X}_1 - \boldsymbol{x}_n),
\end{aligned} \tag{EPRKN3}$$

$$\begin{aligned}
\boldsymbol{x}_{n+1} &= \boldsymbol{x}_n + h\,\overline{\boldsymbol{\Psi}}_1(h\boldsymbol{A}_n)\boldsymbol{v}_n + h\,\overline{\boldsymbol{\Upsilon}}_1\boldsymbol{f}_L(\boldsymbol{x}_n,\boldsymbol{v}_n) + 2h\,\overline{\boldsymbol{\Upsilon}}_3(h\boldsymbol{A}_n)\boldsymbol{R}_v,\\
\boldsymbol{v}_{n+1} &= \boldsymbol{v}_n + h\,\underline{\boldsymbol{\Psi}}_1(h\boldsymbol{A}_n)\boldsymbol{v}_n + h\,\underline{\boldsymbol{\Upsilon}}_1(h\boldsymbol{A}_n)\boldsymbol{f}_L(\boldsymbol{x}_n,\boldsymbol{v}_n) + 2h\,\underline{\boldsymbol{\Upsilon}}_3(h\boldsymbol{A}_n)\boldsymbol{R}_v,
\end{aligned}$$

where $\boldsymbol{\Omega}_1 = \boldsymbol{\Omega}|_{\boldsymbol{x}=\boldsymbol{X}_1}$, $\boldsymbol{\Omega}_n = \boldsymbol{\Omega}|_{\boldsymbol{x}=\boldsymbol{x}_n}$, and $\boldsymbol{H}_n = \boldsymbol{H}|_{\boldsymbol{x}=\boldsymbol{x}_n}$.

3.4.2 Computing the Block Matrix Functions

The formulas described above require computable expressions for each of the block matrix functions. To this end, we invoke the Lagrange-Sylvester Interpolation Polynomial formula. Recall from the previous chapter, the formula asserts if φ is a function analytic on a domain containing the spectrum of the $N \times N$ matrix $\boldsymbol{A}$, then there exists a unique $N-1$ degree polynomial p such that $p(\boldsymbol{A}) = \varphi(h\boldsymbol{A})$. Specifically, we choose the polynomial to be of the form

$$p(\lambda) = a_0 + a_1\lambda + a_2\lambda^2 + \ldots + a_{N-1}\lambda^{N-1},$$

where $N = 4$ for the two dimensional model and $N = 6$ for the three dimensional model. Then by the Lagrange-Sylvester formula $p(\lambda)$ is the polynomial of (at most) degree $N-1$ that interpolates $\varphi(h\lambda)$ on the eigenvalues λ_1, λ_2, …, λ_N of $\boldsymbol{A}$. The interpolation problem is equivalent to solving the linear system

$$\begin{cases}
p(\lambda_1) &= a_0 + a_1\lambda_1 + \cdots + a_{N-1}\lambda_1^{N-1} &= \varphi(h\lambda_1)\\
p(\lambda_1) &= a_0 + a_1\lambda_2 + \cdots + a_{N-1}\lambda_2^{N-1} &= \varphi(h\lambda_2)\\
\quad\vdots &\quad\vdots\qquad\vdots\qquad\quad\vdots &\quad\vdots\\
p(\lambda_N) &= a_0 + a_1\lambda_N + \cdots + a_{N-1}\lambda_N^{N-1} &= \varphi(h\lambda_N)
\end{cases} \tag{3.9}$$

for the unknown polynomial coefficients a_0, a_1, …, a_{N-1}. If the eigenvalues λ_1, λ_2, …, λ_N are all distinct, then the interpolation problem is a system of N linearly independent equations in N unknowns, which is guaranteed to have a unique solution. However, if any eigenvalue is repeated, i.e. λ_j has multiplicity $r_j > 1$, then $r_j - 1$ equations are redundant for λ_j. In this case, for each repeated eigenvalue λ_j we modify system (3.9) by replacing the $r_j - 1$ redundant equations with the following

$r_j - 1$ osculating conditions:

$$\begin{aligned} p'(\lambda_j) &= \varphi'(h\lambda_j), \\ p''(\lambda_j) &= \varphi''(h\lambda_j), \\ \vdots \quad & \quad \vdots \quad \vdots \\ p^{(r_j-1)}(\lambda_j) &= \varphi^{(r_j-1)}(h\lambda_j), \end{aligned}$$

where the superscript denotes the order of the derivative with respect to λ. This modification ensures a system of N linearly independent equations in N unknowns for which there is a unique solution. (A proof of the Lagrange-Sylvester Interpolation Polynomial formula is presented in appendix A.) Applying this formula, the matrix function $\varphi_k(h\boldsymbol{A})$ has the polynomial expression

$$\varphi_k(h\boldsymbol{A}) = p(\boldsymbol{A}) = a_0\boldsymbol{I}_{N\times N} + a_1\boldsymbol{A} + \ldots + a_{N-1}\boldsymbol{A}^{N-1}.$$

Each block matrix function can then be expressed by an (at most) $N-1$ degree matrix polynomial in terms of the blocks of $\boldsymbol{A}$. To see this, observe that the powers of $\boldsymbol{A}$ are given by the recursive formula

$$\boldsymbol{A}^j = \begin{bmatrix} \boldsymbol{O} & \boldsymbol{I} \\ \boldsymbol{H} & \boldsymbol{\Omega} \end{bmatrix}^j = \begin{bmatrix} \boldsymbol{R}^{(j-1)} & \boldsymbol{S}^{(j-1)} \\ \boldsymbol{S}^{(j-1)}\boldsymbol{H} & \boldsymbol{R}(j-1) + \boldsymbol{S}^{(j-1)}\boldsymbol{\Omega} \end{bmatrix} \quad j = 1, 2, \ldots,$$

where $\boldsymbol{R}^{(0)} = \boldsymbol{O}$ and $\boldsymbol{S}^{(0)} = \boldsymbol{I}$. Then for the two dimensional model, $\boldsymbol{A}$ is a 4×4 matrix with the matrix function $\varphi_k(h\boldsymbol{A})$ given by

$$\begin{aligned} \varphi_k(h\boldsymbol{A}) &= \begin{bmatrix} \overline{\boldsymbol{\Psi}}_k(\boldsymbol{A}) & \overline{\boldsymbol{\Upsilon}}_k(\boldsymbol{A}) \\ \underline{\boldsymbol{\Psi}}_k(\boldsymbol{A}) & \underline{\boldsymbol{\Upsilon}}_k(\boldsymbol{A}) \end{bmatrix} \\ &= a_0\boldsymbol{I} + a_1\boldsymbol{A} + a_2\boldsymbol{A}^2 + a_3\boldsymbol{A}^3 \\ &= a_0\begin{bmatrix} \boldsymbol{I} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{I} \end{bmatrix} + a_1\begin{bmatrix} \boldsymbol{O} & \boldsymbol{I} \\ \boldsymbol{H} & \boldsymbol{\Omega} \end{bmatrix} + a_2\begin{bmatrix} \boldsymbol{H} & \boldsymbol{\Omega} \\ \boldsymbol{\Omega}\boldsymbol{H} & \boldsymbol{H} + \boldsymbol{\Omega}^2 \end{bmatrix} \\ &\quad + a_3\begin{bmatrix} \boldsymbol{\Omega}\boldsymbol{H} & \boldsymbol{H} + \boldsymbol{\Omega}^2 \\ (\boldsymbol{H} + \boldsymbol{\Omega}^2)\boldsymbol{\Omega} & \boldsymbol{\Omega}\boldsymbol{H} + (\boldsymbol{H} + \boldsymbol{\Omega}^2)\boldsymbol{\Omega} \end{bmatrix}, \end{aligned}$$

where $\boldsymbol{O}, \boldsymbol{I}, \boldsymbol{H}, \boldsymbol{\Omega}$ are 2×2 matrices. This gives the following explicit expressions for the block matrix functions of the two dimensional model:

$$\begin{aligned} \overline{\boldsymbol{\Psi}}_k(h\boldsymbol{A}) &= a_0\boldsymbol{I} + a_2\boldsymbol{H} + a_3\boldsymbol{\Omega}\boldsymbol{H}, \\ \overline{\boldsymbol{\Upsilon}}_k(h\boldsymbol{A}) &= a_1\boldsymbol{I} + a_2\boldsymbol{\Omega} + a_3(\boldsymbol{H} + \boldsymbol{\Omega}^2), \\ \underline{\boldsymbol{\Psi}}_k(h\boldsymbol{A}) &= a_1\boldsymbol{H} + a_2\boldsymbol{\Omega}\boldsymbol{H} + a_3(\boldsymbol{H} + \boldsymbol{\Omega}^2)\boldsymbol{H}, \\ \underline{\boldsymbol{\Upsilon}}_k(h\boldsymbol{A}) &= a_0\boldsymbol{I} + a_1\boldsymbol{\Omega} + a_2(\boldsymbol{H} + \boldsymbol{\Omega}^2) + a_3(\boldsymbol{\Omega}\boldsymbol{H} + (\boldsymbol{H} + \boldsymbol{\Omega}^2)\boldsymbol{\Omega}). \end{aligned}$$

For the three dimensional model, $\boldsymbol{A}$ is a 6×6 matrix and the matrix function $\varphi_k(h\boldsymbol{A})$ is given by

$$\begin{aligned}\varphi_k(h\boldsymbol{A}) &= \begin{bmatrix}\overline{\boldsymbol{\Psi}}_k(\boldsymbol{A}) & \overline{\boldsymbol{\Upsilon}}_k(\boldsymbol{A})\\ \underline{\boldsymbol{\Psi}}_k(\boldsymbol{A}) & \underline{\boldsymbol{\Upsilon}}_k(\boldsymbol{A})\end{bmatrix}\\ &= a_0\boldsymbol{I}+a_1\boldsymbol{A}+a_2\boldsymbol{A}^2+a_3\boldsymbol{A}^3+a_4\boldsymbol{A}^4+a_5\boldsymbol{A}^5\\ &= a_0\begin{bmatrix}\boldsymbol{I} & \boldsymbol{O}\\ \boldsymbol{O} & \boldsymbol{I}\end{bmatrix}+a_1\begin{bmatrix}\boldsymbol{O} & \boldsymbol{I}\\ \boldsymbol{H} & \boldsymbol{\Omega}\end{bmatrix}+a_2\begin{bmatrix}\boldsymbol{H} & \boldsymbol{\Omega}\\ \boldsymbol{R}^{(2)} & \boldsymbol{S}^{(2)}\end{bmatrix}\\ &\quad+a_3\begin{bmatrix}\boldsymbol{R}^{(2)} & \boldsymbol{S}^{(2)}\\ \boldsymbol{R}^{(3)} & \boldsymbol{S}^{(3)}\end{bmatrix}+a_4\begin{bmatrix}\boldsymbol{R}^{(3)} & \boldsymbol{S}^{(3)}\\ \boldsymbol{R}^{(4)} & \boldsymbol{S}^{(4)}\end{bmatrix}+a_5\begin{bmatrix}\boldsymbol{R}^{(4)} & \boldsymbol{S}^{(4)}\\ \boldsymbol{R}^{(5)} & \boldsymbol{S}^{(5)}\end{bmatrix},\end{aligned}$$

where $\boldsymbol{O},\boldsymbol{I},\boldsymbol{H},\boldsymbol{\Omega}$ are 3×3 matrices and

$$\begin{aligned}\boldsymbol{R}^{(j)} &= \begin{cases}\boldsymbol{H}, & j=1\\ \boldsymbol{S}^{(j-1)}\boldsymbol{H}, & j=2,3,\ldots\end{cases}\\ \boldsymbol{S}^{(j)} &= \begin{cases}\boldsymbol{\Omega}, & j=1\\ \boldsymbol{R}^{(j-1)}+\boldsymbol{S}^{(j-1)}\boldsymbol{\Omega}, & j=2,3,\ldots\end{cases}\end{aligned}$$

This gives the polynomial expressions for the block matrix functions of the three dimensional model:

$$\begin{aligned}\overline{\boldsymbol{\Psi}}_k(h\boldsymbol{A}) &= a_0\boldsymbol{I}+a_2\boldsymbol{H}+a_3\boldsymbol{R}^{(2)}+a_4\boldsymbol{R}^{(3)}+a_5\boldsymbol{R}^{(4)},\\ \overline{\boldsymbol{\Upsilon}}_k(h\boldsymbol{A}) &= a_1\boldsymbol{I}+a_2\boldsymbol{\Omega}+a_3\boldsymbol{S}^{(2)}+a_4\boldsymbol{S}^{(3)}+a_5\boldsymbol{S}^{(4)},\\ \underline{\boldsymbol{\Psi}}_k(h\boldsymbol{A}) &= a_1\boldsymbol{H}+a_2\boldsymbol{R}^{(2)}+a_3\boldsymbol{R}^{(3)}+a_4\boldsymbol{R}^{(4)}+a_5\boldsymbol{R}^{(5)},\\ \underline{\boldsymbol{\Upsilon}}_k(h\boldsymbol{A}) &= a_0\boldsymbol{I}+a_1\boldsymbol{\Omega}+a_2\boldsymbol{S}^{(2)}+a_3\boldsymbol{S}^{(3)}+a_4\boldsymbol{S}^{(4)}+a_5\boldsymbol{S}^{(5)}.\end{aligned}$$

Observe that by our choice of setting the interpolation polynomial to be of the form

$$p(\lambda)=a_0+a_1\lambda+a_2\lambda^2+\ldots+a_{N-1}\lambda^{N-1},$$

the Lagrange-Sylvester formula exploits the recursive structure of the powers of the block matrices resulting in computationally efficient polynomial expressions. Also notice that by this particular form of the polynomial, solving the interpolation problem (3.9) is an ill-conditioned (Vandermonde) linear system. To overcome this issue, we employ Cramer's rule to derive analytic expressions for the coefficients a_0, a_1, a_2, …, a_{N-1} in our implementations of the Nyström exponential integrators yielding additional computational savings, see appendix B. As a final note, the analytic expressions for the coefficients a_0, a_1, …, a_{N-1} are in general subject to catastrophic cancellation for small argument values $z=h\lambda$. Thus, for small $z=h\lambda$ we compute each coefficient using a five-term Taylor polynomial approximation for any problematic analytic expression.

3.5 Numerical Experiments

In this section, we compare the computational performances of the Runge-Kutta-Nyström-type exponential integrators EPRKN2 and EPRKN3 against the standard EP2 and EPRK3 exponential integrators. The Boris and Buneman algorithms are also included to evaluate how well these exponential integrators perform against conventional particle pushers.

All numerical experiments model a particle of unit mass and unit charge in a strong magnetic field orientated in the z direction. The specific configurations for each test problem are described in the next two subsections below. For reference, we computed highly accurate solutions to each test problem using the MATLAB `ode113` integrator with error tolerances set to 10^{-12} for `RelTol` (relative error tolerance) and 10^{-12} for `AbsTol` (absolute error tolerance). The relative error of each particle particle pusher is defined by

$$\text{error } = \frac{\|\boldsymbol{x}^* - \boldsymbol{x}\|}{\|\boldsymbol{x}^*\|},$$

where $\boldsymbol{x}^*$ is the particle position of the reference solution, $\boldsymbol{x}$ is the particle position of the solution of the particle pusher, and $\|\cdot\|$ denotes the Euclidean norm. Experiments with the electric potential well problems were implemented in C++ using the Eigen C++ template library for linear algebra [18]. Experiments with the gyroradius and grad-B drift problems were implemented in MATLAB. All computations in these experiments were calculated with double precision floating point operations.

3.5.1 Two Dimensional Model Configurations

The initial conditions for all two dimensional model test problems are $\boldsymbol{x}_0 = (1, 0)$ and $\boldsymbol{v}_0 = (0, -1)$. Each test problem is integrated over the time interval [0, 100].

- **Electric Potential Well Problems:** These test problems consider particle motion in a strong uniform (constant in time and space) magnetic field $\boldsymbol{B} = 100\,\hat{\boldsymbol{z}}$ and a non-uniform (in space) electric field $\boldsymbol{E}$. The electric field $\boldsymbol{E}$ is given by an electric potential well such that the resulting anisotropic drift motion forms a closed orbit on temporal and spatial scales much larger than the gyromotion. The experiments are conducted with three specific electric fields given by a quadratic potential, a cubic potential, and a quartic potential. The electric field for each electric potential well test problem is specified in table 3.1.

- **Gyroradius Problem:** This experiment examines the gyroradius of the solutions computed by the numerical particle pushers. A known issue with the Boris pusher [33] (as well as with many other conventional particle pushers such as the Buneman pusher) is that in problems with an $\boldsymbol{E} \times \boldsymbol{B}$ drift motion it computes an artificially enlarged gyroradius when using large step sizes relative to the gyroperiod. In this context, a step size h is considered "small" when $\omega h \ll 1$ and "large" when $\omega h \gg 1$, where $\omega = qB/m$ is the gyroperiod. For strongly

magnetized problems, this can impose a severe restriction on the step size for many conventional particle pushers if accuracy requirements of the simulations demand resolution at the scale of the gyroradius. We include this experiment to study how well the exponential integrators are able to correctly resolve the gyroradius. The test problem under consideration is a linear $\boldsymbol{E} \times \boldsymbol{B}$ drift problem with electromagnetic fields

$$\boldsymbol{B} = 100\,\hat{\boldsymbol{z}} \quad \text{and} \quad \boldsymbol{E} = -\begin{bmatrix} 0 \\ 1+y \end{bmatrix},$$

which has a gyroradius of $r = 0.01$. The problem is integrated using a "small" step size $h = 0.001$ and a "large" step size $h = 0.1$.

- **Grad-B Drift Problem:** This experiment examines the so-called grad-B drift problem. The test problem has a non-uniform magnetic field with a gradient term in which the length scale of the spatial variation is much longer than the gyroradius. In other words, the variation in the magnetic field that the particle experiences is "small" over the gyro-orbit. This is formally stated by

$$\frac{r\|\nabla B\|}{B} \ll 1,$$

where r is the gyroradius, $B = \|\boldsymbol{B}\|$, and ∇B is the magnetic field gradient. Under this condition, the particle experiences an approximate drift velocity of [9, 23, 30]

$$\boldsymbol{v}_{\nabla B} = \frac{1}{2}\frac{v_\perp^2}{\omega}\frac{\boldsymbol{B} \times \nabla B}{B^2},$$

where $v_\perp$ is the particle speed in the plane perpendicular to the magnetic field and $\omega = qB/m$ is the gyrofrequency. The electromagnetic fields for this test problem are

$$\boldsymbol{B} = (100 + \delta B\, y)\hat{\boldsymbol{z}} \quad \text{and} \quad \boldsymbol{E} = \boldsymbol{0}.$$

	Quadratic Well	Cubic Well	Quartic Well
Potential $V(\boldsymbol{x})$:	$50(x^2+y^2)$	$47(x^2+y^2)$ $+x^3+y^3$	$\frac{25}{3}(x^4+y^4)$
Electric Field $\boldsymbol{E}(\boldsymbol{x})$:	$-100\begin{bmatrix} x \\ y \end{bmatrix}$	$-\begin{bmatrix} 94x+3x^2 \\ 94y+3y^2 \end{bmatrix}$	$-\frac{100}{3}\begin{bmatrix} x^3 \\ y^3 \end{bmatrix}$

Table 3.1: Electric fields for 2D potential well problems with uniform magnetic field $\boldsymbol{B} = 100\,\hat{\boldsymbol{z}}$

3.5.2 Three Dimensional Model Configurations

All three dimensional test problems are electric potential well configurations with a uniform magnetic field $\boldsymbol{B} = 100\,\hat{\boldsymbol{z}}$ and a spatially non-uniform electric field $\boldsymbol{E}$. Similar to the two dimensional model, we examine three specific electric fields given by a quadratic potential, a cubic potential, and a quartic potential. Configurations for the electric scalar potential wells and their corresponding electric fields are shown in table 3.2. The initial conditions are $\boldsymbol{x}_0 = (1, 0, 0)$ and $\boldsymbol{v}_0 = (0, -1, 1)$. Each test problem is integrated over the time interval [0, 100].

	Quadratic Well	Cubic Well	Quartic Well
Potential $V(\boldsymbol{x})$:	$50(x^2+y^2)$ $+5z^2$	$47(x^2+y^2)$ $+x^3+y^3$ $+\frac{1}{10}(47z^2+z^3)$	$\frac{25}{3}(x^4+y^4)$ $+\frac{5}{6}z^4$
Electric Field $\boldsymbol{E}(\boldsymbol{x})$:	$-\begin{bmatrix}100x\\100y\\10z\end{bmatrix}$	$-\begin{bmatrix}94x+3x^2\\94y+3y^2\\\frac{47}{5}z+\frac{3}{10}z^2\end{bmatrix}$	$-\frac{1}{3}\begin{bmatrix}100x^3\\100y^3\\10z^3\end{bmatrix}$

Table 3.2: Electric fields for 3D potential well problems with uniform magnetic field $\boldsymbol{B} = 100\,\hat{\boldsymbol{z}}$

3.5.3 Two Dimensional Model Results

Results of Electric Potential Well Problems

Figure 3.1 displays plots of the experiment results. Plots of the reference solution are in the top row. Work-precision diagrams are in the bottom row. Results for the quadratic, cubic, and quartic potential problems are in the left, center, and right columns, respectively.

Note that for the quadratic potential well problem the exponential integrators exhibit superior performance as expected, since this is a linear problem. For the cubic potential well problem, all of the exponential integrators compute solutions more accurately and efficiently than the Boris and Buneman pushers. For the quartic potential well problem, the exponential integrators are at least competititve if not better than the Boris and Buneman particle pushers. In particular, the Nyström-type exponential integrators consistently outperform both standard exponential methods and the Boris and Buneman pushers for all levels of accuracy yielding significant improvements in efficiency. Table 3.3 quantifies this gain in computational efficiency by showing the average ratios of the CPU times of the standard exponential integrators to the Nyström-type exponential integrators for each test problem.

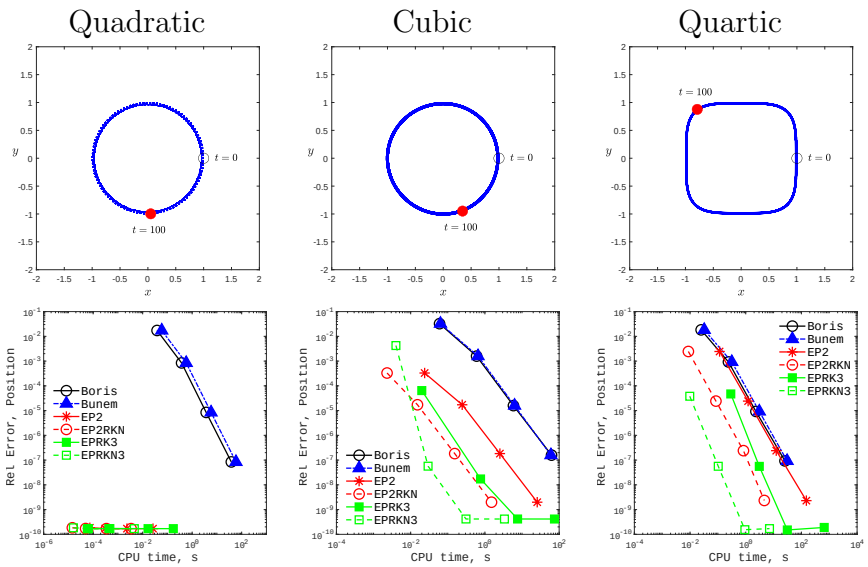


Figure 3.1: Results for 2D potential well test problems: reference solution orbits (first row) and precision diagrams (second row). Boris/Buneman step sizes are $h = 10^{-3}$, 10^{-4}, 10^{-5}, 10^{-6} for the quadratic potential problem and $h = 10^{-4}$, 10^{-5}, 10^{-6}, 10^{-7} for the cubic/quartic potential problems. Exponential integrators step sizes are $h =$ 100, 10, 1, 10^{-1} for the quadratic potential problem and $h = 10^{-2}$, 10^{-3}, 10^{-4}, 10^{-5} for the cubic/quartic potential problems.

	Quadratic Well	Cubic Well	Quartic Well
EP2/EPRKN2	7.56	16.97	29.50
EPRK3/EPRKN3	36.86	22.46	84.73

Table 3.3: Average CPU time ratios of standard exponential integrators to Nyström type exponential integrators for 2D potential well problems.

Results of Gyroradius Problem

Figure 3.2 shows the experiment results for the gyroradius problem. For the "small" step size $h = 0.001$, all particle pushers compute the gyroradius accurately. However for the "large" step size $h = 0.1$, both the Boris and Buneman algorithms compute an artificially enlarged gyroradius while all the exponential integrators compute the correct gyroradius.

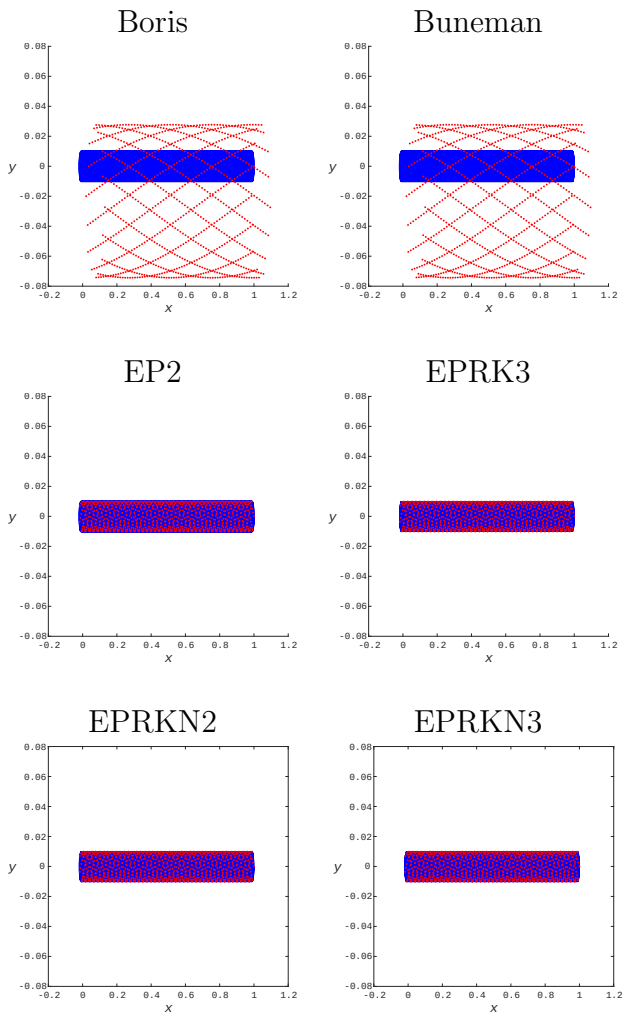


Figure 3.2: Plots of computed trajectories for the gyroradius $\boldsymbol{E} \times \boldsymbol{B}$ drift problem. Solutions for step size $h = 0.001$ are solid blue and solutions for step size $h = 0.01$ are dotted red.

Results of Grad-B Drift Problem

Figure 3.3 shows plots of the reference solution orbits and the precision diagrams for $\delta B = 0.1$, 1, 10. Again the Nyström-type exponential integrators compute much faster than the standard exponential integrators. Table 3.4 shows the average ratios of the CPU times of the standard exponential integrators to the Nyström-type exponential integrators.

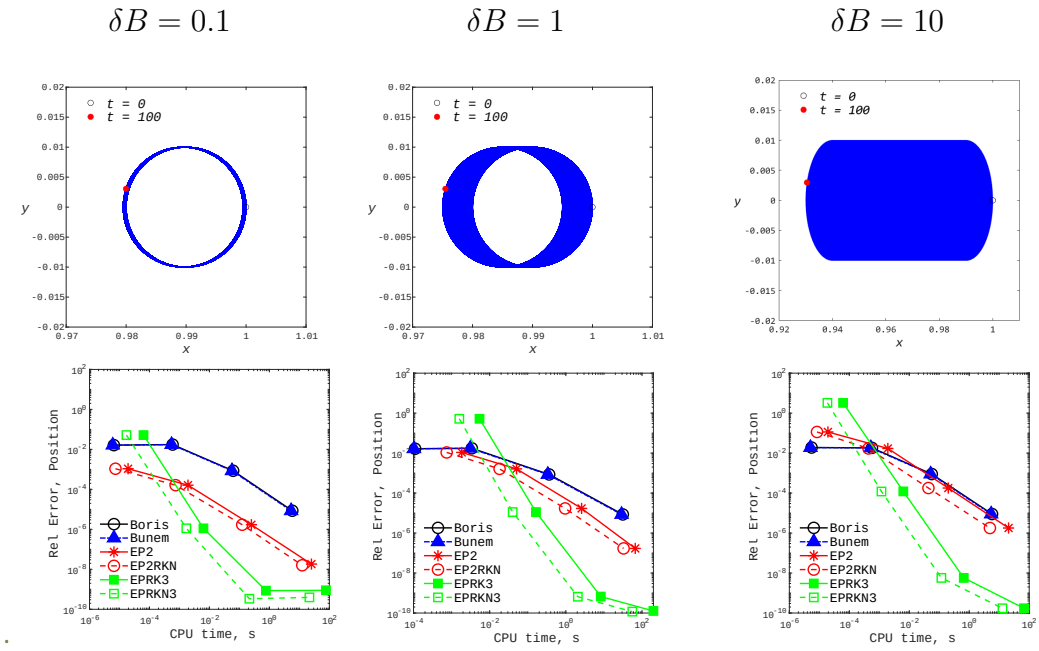


Figure 3.3: Results for grad-B drift problem: reference solution orbits (top row), and precision diagrams (bottom row). Boris/Buneman step sizes are $h = 10^{-2}$, 10^{-3}, 10^{-4}, 10^{-5}. EP2/EPRK3 step sizes are $h = 10^{-1}$, 10^{-2}, 10^{-3}, 10^{-4}.

	$\delta B = 0.1$	$\delta B = 1$	$\delta B = 10$
EP2/EPRKN2	2.03	2.04	4.19
EPRK3/EPRKN3	3.57	3.60	5.24

Table 3.4: Average CPU time ratios of standard exponential integrators to Nyström type exponential integrators for the grad-B drift problems.

3.5.4 Three Dimensional Model Results

Figure 3.4 shows plots of the reference solution orbits and the work-precision diagrams for the three dimensional electric potential well problems. As expected, all the exponential integrators exhibit superior performance for the quadratic potential well problem. For the cubic potential well problem, the exponential integrators outperform the Boris and Buneman algorithms in terms of computation speed for comparable levels of accuracy. For the quartic potential well problem, the exponential integrators are competitive with the Boris and Buneman algorithms. For all of the test problems, the Nyström type exponential integrators compute faster than the standard exponential integrators and outperform the Boris and Buneman integrators. The average CPU time ratios of the standard exponential integrators to the Nyström-type exponential integrators are shown in table 3.5.

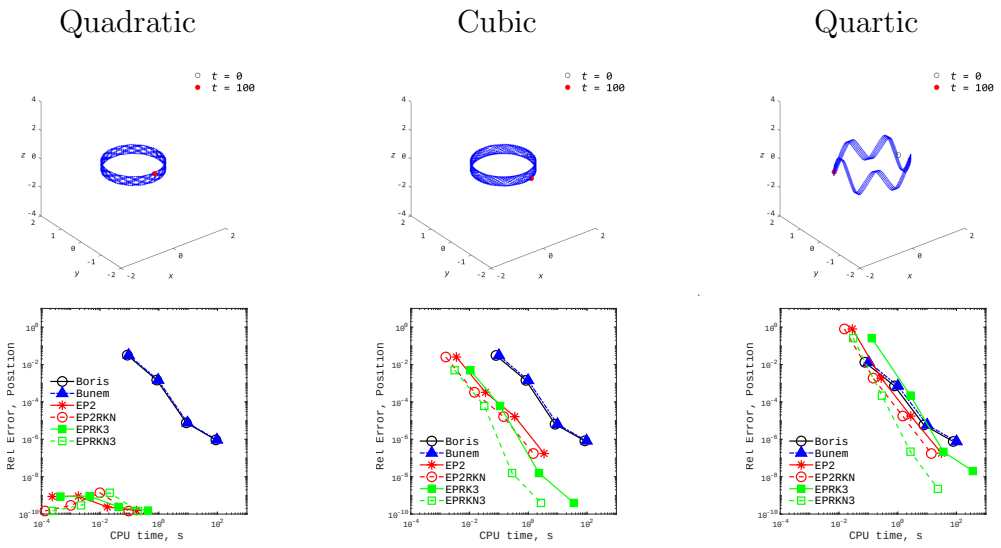


Figure 3.4: Results for 3D potential well test problems: reference solution orbits (first row) and precision diagrams (second row). Boris/Buneman step sizes are $h = 10^{-3}$, 10^{-4}, 10^{-5}, 10^{-6} for the quadratic well problem, $h = 10^{-4}$, 10^{-5}, 10^{-6}, 10^{-7} for the cubic/quartic potential problems. Exponential integrators step sizes are $h = 100$, 10, 1, 10^{-1} for the quadratic potential problem and $h = 10^{-2}$, 10^{-3}, 10^{-4}, 10^{-5} for the cubic/quartic potential problems.

	Quadratic Well	Cubic Well	Quartic Well
EP2/EPRKN2	1.85	2.32	2.18
EPRK3/EPRKN3	2.18	12.57	15.16

Table 3.5: Average CPU time ratios of standard exponential integrators to Nyström type exponential integrators for 3D potential well problems.

3.6 Conclusion

3.6.1 Summary

In this chapter we derived Nyström-type exponential integrators induced by partitioning standard exponential methods. In particular, we partitioned the second-order EP2 and the third-order EPRK3 methods corresponding to the $\boldsymbol{x}$ and $\boldsymbol{v}$ components to construct second-order EPRKN2 and the third-order EPRKN3 methods that effectively solve the particle pushing problem as a second-order differential equation. These Nyström-type exponential integrators exploit the mathematical structure of the Newtonian formulation of the particle pushing problem to improve computational efficiency. Numerical experiments demonstrate that the Nyström exponential integrators exhibit significant improvements in computation times compared to the standard exponential integrators for the same level of accuracy for both the two dimensional and three dimensional models. This work shows that Nyström exponential integrators are a promising alternative to solve strongly magnetized particle pushing problems.

3.6.2 Future Work

While the exponential integrators we constructed offer improvements in the accuracy of the solution, they are not specifically designed to preserve any geometric properties of the solution exactly. In our future work we will investigate whether exponential schemes that preserve phase space volume or energy can be constructed. Such volume- or energy-preserving methods are desired when integration over very long time intervals has to be done. Additionally, our numerical experiments showed that the computational savings offered by the new exponential methods are larger for linear or weakly nonlinear problems compared to strongly nonlinear configurations such as the quartic potential and the grad-B drift problems. A possible approach we plan to investigate to address this issue is to develop a better quadrature for the nonlinear integral terms in the variation of constants Volterra integral equation which serves as the starting point for construction of an exponential integrator. We also plan to conduct numerical experiments with more complex electromagnetic configurations for more realistic test problems. Finally, a thorough evaluation of these exponential integrators requires comparing them against the more advanced conventional particle pushers such as the modified Crank-Nicolson scheme [35] and the filtered Boris algorithm [19].

Chapter 4

Conclusion

In this dissertation, we investigated exponential integration as an alternative approach to numerically simulate charged particle dynamics in the presence of electromagnetic fields. This problem is particularly important in the context of solving the particle pushing problem for PIC simulations of strongly magnetized plasma. We have developed a numerical approach which combines ideas of exponential integration and interpolation as a method to evaluate matrix functions to constuct effective exponential methods. In chapter 2, we presented second-order and third-order numerical particle pushing schemes using the second-order EP2 and third-order EPRK3 Runge-Kutta type exponential integrator methods, respectively. The use of the Lagrange-Sylvester Interpolation Polynomial formula to compute the matrix exponential-like φ functions required in these exponential integration schemes allowed us to create numerical methods that offer significant computational savings compared to widely used numerical particle pushers. Our numerical experiments comparing the new exponential integrators against the conventional Boris and Buneman algorithm showed that exponential integrators are superior for linear and weakly nonlinear problems and are competitive for strongly nonlinear configurations. This work offered evidence that exponential methods present an efficient alternative to standard particle pushers and justified further research in this area. In fact, we were able to move further in this direction by constructing integrators with improved computational efficiency compared to standard exponential schemes not particularly adapted for the particle pushing problems.

In chapter 3, we derived Nyström-type exponential integrators that take advantage of the structure of the particle pushing problem to deliver better computational savings compared to standard exponential schemes. The exponential Nyström-type methods effectively solve the Newtonian formulation of the particle pushing problem as a second-order initial value system directly. Construction of such methods was accomplished by partitioning the standard exponential integration framework from the EPIRK class of methods into components corresponding to particle position $\boldsymbol{x}$ and particle velocity $\boldsymbol{v}$ resulting in partitioned exponential integrators. Our numerical experiments showed significant improvements in the computational efficiency of these Nyström-type exponential integrators when compared to the standard exponential

integrators, thus delivering even better computational savings against the Boris and Buneman schemes.

In summary, this dissertation demonstrates the proof of concept of using exponential integrators to solve strongly magnetized particle pushing problems. The novel developments presented in this dissertation include:

- Derivation of exponential integrators to solve strongly magnetized particle pushing problems.
- Discovering the Lagrange-Sylvestor Interpolation Polynomial formula as a computationally efficient method to compute the matrix φ functions for low dimensional problems.
- Development of Runge-Kutta-Nyström type exponential integrators that exploit the mathematical structure of the Newtonian form of the particle pushing problem for improved computational efficiency.

Results from numerical experiments illustrate that exponential integration is a promising alternative to solving strongly magnetized particle pushing problems.

This work opened a number of promising directions in development of efficient numerical methods for particle pushing problems. In particular, our future plans include construction of geometric exponential integrators that preserve the mathematical structure of the exact solution such as phase space volume or energy conservation. Methods with these properties are of a particular interst for applications where very long time computation of the particles trajectories is needed. Another important research direction we plan to pursue is development of exponential schemes that are better suited for the integration of strongly nonlinear problems. Constructing quadrature approximations that model the nonlinearity more accurately could yield both more efficiency and accuracy and lead to even more computational savings. In addition, the new integrators have to be tested on a wider range of problems with more complex electromagnetic field configurations and incorporated into widely used PIC codes. We also plan to compare the exponential methods with other advanced integrators such as the modified Crank-Nicolson scheme and the filtered Boris algorithm.

Appendix A

Proof of Lagrange-Sylvester Interpolation Polynomial Formula

The text of this appendix is a reprint of the material as it appears in T.P. Nguyen, I. Joseph and M. Tokman, Exploring exponential time integration for strongly magnetized charged particle motion, *Computer Physics Communications*, 304 (2024), 109294, doi: https://doi.org/10.1016/j.cpc.2024.109294.

Case: $\boldsymbol{A}$ has N distinct eigenvalues.

If $\boldsymbol{A}$ has N distinct eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_N$, then its characteristic polynomial satisfies

$$\begin{aligned}\det(\lambda \boldsymbol{I} - \boldsymbol{A}) &= (\lambda - \lambda_1)(\lambda - \lambda_2)\cdots(\lambda - \lambda_N) \\ &= \lambda^N + \alpha_{N-1}\lambda^{N-1} + \ldots + \alpha_1\lambda + \alpha_0 \\ &= 0.\end{aligned}$$

Solving for λ^N gives

$$\lambda^N = -\alpha_{N-2}\,\lambda^{N-1} - \ldots - \alpha_1\,\lambda^2 - \alpha_0\,\lambda. \tag{A.1}$$

In other words, λ^N can be expressed in terms of $\lambda, \lambda^2, \ldots, \lambda^{N-1}$, i.e. a polynomial of (at most) degree $N-1^{\text{th}}$.
Multiplying equation (A.1) by λ gives

$$\lambda^{N+1} = -\alpha_{N-2}\,\lambda^N - \ldots - \alpha_1\,\lambda^3 - \alpha_0\,\lambda^2.$$

Substituting equation (A.1) into the right-hand side and grouping powers of λ yields

$$\lambda^{N+1} = \alpha_{N-1}^{(1)}\,\lambda^{N-1} + \alpha_{N-2}^{(1)}\,\lambda^{N-2} + \ldots + \alpha_2^{(1)}\,\lambda^2 + \alpha_1^{(1)}\,\lambda,$$

for some coefficients $\alpha_1^{(1)}, \alpha_2^{(1)}, \ldots, \alpha_{N-1}^{(1)}$. It follows from induction that for any $k = 0, 1, 2, \ldots$,

$$\lambda^{N+k} = \alpha_1^{(k)}\,\lambda \,+\, \alpha_2^{(k)}\,\lambda^2 \,+\, \ldots \,+\, \alpha_{N-1}^{(k)}\,\lambda^{N-1}. \tag{A.2}$$

That is, λ^{N+k} can always be expressed in terms of $\lambda, \lambda^2, \ldots, \lambda^{N-1}$, i.e. a polynomial of (at most) degree $N{-}1^{\text{th}}$.

Since $f(\lambda)$ is an analytic function, it has a convergent series expansion:

$$\begin{aligned} f(\lambda) &= c_0 \,+\, c_1\,\lambda \,+\, c_2\,\lambda^2 \,+\, \ldots \\ &= c_0 \,+\, c_1\,\lambda \,+\, \ldots \,+\, c_{N-1}\,\lambda^{N-1} \,+\, c_N\,\lambda^N \,+\, \ldots \,+\, c_{N+k}\,\lambda^{N+k} \,+\, \ldots \\ &= c_0 \,+\, c_1\,\lambda \,+\, \ldots \,+\, c_{N-1}\,\lambda^{N-1} \,+\, \sum_{k=0}^{\infty} c_{N+k}\,\lambda^{N+k}. \end{aligned}$$

From equation (A.2), each term inside the summation can be expressed by a polynomial of (at most) $N{-}1^{\text{th}}$ degree. Making the substitutions and grouping powers of λ gives the polynomial

$$p(\lambda) = a_0 \,+\, a_1\,\lambda \,+\, \ldots \,+\, a_{N-1}\,\lambda^{N-1} = f(\lambda),$$

for some coefficients $a_0, a_1, \ldots, a_{N-1}$. In other words, $f(\lambda)$ can be expressed by some polynomial $p(\lambda)$ of (at most) degree $N{-}1$.

To find an explicit expression for this polynomial, observe that

$$p(\lambda_j) = f(\lambda_j) \tag{A.3}$$

must hold true for each eigenvalue λ_j. This yields a system of N linearly independent equations in N coefficients $a_0, a_1, \ldots, a_{N-1}$. Hence, $p(\lambda)$ is the unique polynomial that interpolates $f(\lambda)$ on the spectrum of $\boldsymbol{A}$.

Extending the series representation of $f(\lambda)$ to the matrix argument $\boldsymbol{A}$, we have

$$f(\boldsymbol{A}) = c_0\,I + c_1\,A + c_2\,A^2 + \ldots.$$

This implies that

$$p(A) = a_0\,\boldsymbol{I} \,+\, a_1\,\boldsymbol{A} \,+\, \ldots \,+\, a_{N-1}\,\boldsymbol{A}^{N-1} = f(\boldsymbol{A}).$$

Case: $\boldsymbol{A}$ has repeated eigenvalues.

For the case when $\boldsymbol{A}$ has repeated eigenvalues, suppose $\boldsymbol{A}$ has eigenvalues

$$\lambda_1, \lambda_2, \ldots, \lambda_m$$

with respective multiplicities

$$r_1, r_2, \ldots, r_m,$$

where

$$m \le N \quad \text{and} \quad r_1 \,+\, r_2 \,+\, \ldots \,+\, r_m = N.$$

Then the characteristic polynomial of A satisfies

$$\begin{aligned}\det(\lambda \boldsymbol{I} - \boldsymbol{A}) &= (\lambda - \lambda_1)^{r_1} (\lambda - \lambda_2)^{r_2} \cdots (\lambda - \lambda_m)^{r_m} \\ &= \lambda^{N-1} + \alpha_{N-2} \lambda^{N-2} + \ldots + \alpha_1 \lambda + \alpha_0 \\ &= 0.\end{aligned}$$

Following the same argument as in the previous case, for any $k = 0, 1, \ldots$ λ^{N+k} can be expressed by a polynomial of (at most) $N-1$ degree polynomial in λ. Hence, $f(\lambda)$ can be expressed by a polynomial of (at most) degree $N-1$:

$$p(\lambda) = a_0 + a_1 \lambda + \ldots + a_{N-1} \lambda^{N-1} = f(\lambda)$$

for some coefficients $a_0, a_1, \ldots, a_{N-1}$.

For each eigenvalue λ_j with multiplicity r_j, we make the observation that

$$\begin{array}{rcll} p(\lambda_j) & = & f(\lambda_j) & \text{interpolation condition,} \\ p'(\lambda_j) & = & f'(\lambda_j) & \text{1st osculating condition,} \\ p''(\lambda_j) & = & f''(\lambda_j) & \text{2nd osculating condition,} \\ \vdots & \vdots & \vdots & \vdots \\ p^{(r_j-1)}(\lambda_j) & = & f^{(r_j-1)}(\lambda_j) & r_j - 1^{\text{th}} \text{ osculating condition,} \end{array} \tag{A.4}$$

where the superscript denotes the order of the derivative with respect to λ. This establishes a system of N linearly independent equations in N coefficients $a_0, a_1, \ldots, a_{N-1}$. Hence, $p(\lambda)$ is the unique interpolation polynomial that satisfies $r_k - 1$ osculating conditions for each eigenvalue λ_j of multiplicity r_j. Consequently,

$$p(A) = a_0 \boldsymbol{I} + a_1 \boldsymbol{A} + \ldots + a_{N-1} \boldsymbol{A}^{N-1} = f(\boldsymbol{A}).$$

□

Appendix B

Lagrange-Sylvester Interpolation Polynomial Coefficients

This appendix describes the analytic expressions for the coefficients of the Lagrange-Sylvester interpolation polynomial for the test problems discussed in this dissertation. Recall that the Jacobian matrix of the Newtonian form of the particle pushing problem is

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{O} & \boldsymbol{I} \\ \boldsymbol{H} & \boldsymbol{\Omega} \end{bmatrix},$$

where $\boldsymbol{O}$ and $\boldsymbol{I}$ are the $d \times d$ zero and identity matrices, respectively, $\boldsymbol{H} = \partial \boldsymbol{f}_L / \partial \boldsymbol{x}$ is the Jacobian matrix of $\boldsymbol{f}_L$ with respect to $\boldsymbol{x}$, and $\boldsymbol{\Omega}$ is the $d \times d$ skew symmetrc matrix such that $\boldsymbol{\Omega}\,\boldsymbol{v} = \frac{q}{m}\boldsymbol{v} \times \boldsymbol{B}$. Here, $d = 2$ for the two dimensional model, and $d = 3$ for the three dimensional model.

B.1 Two Dimensional Model

For the two dimensional model,

$$\boldsymbol{H} = \frac{\partial \boldsymbol{f}_L}{\partial \boldsymbol{x}} = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \qquad \text{and} \qquad \boldsymbol{\Omega} = \begin{bmatrix} 0 & \omega \\ -\omega & 0 \end{bmatrix}, \quad \omega = \frac{qB}{m}.$$

The characteristic polynomial of $\boldsymbol{A}$ is

$$\det(\lambda \boldsymbol{I}_{4\times 4} - \boldsymbol{A}) = \lambda^4 + \lambda^2 P + \lambda Q + R,$$

where

$$\begin{aligned} P &= \omega^2 - H_{11} - H_{22}, \\ Q &= \omega(H_{12} - H_{21}), \\ R &= H_{11}H_{22} - H_{12}H_{21}. \end{aligned}$$

Note that all particle pushing problems under consideration in this dissertation are strongly magnetized, which implies $P \neq 0$.

B.1.1 Electric Potential Well and Gyro-radius Problems

For these test problems $\boldsymbol{H}$ is a diagonal matrix, hence $Q = 0$ and the characteristic polynomial reduces to

$$\lambda^4 + \lambda^2 P + R.$$

To determine the polynomial coefficients for the interpolation problem

$$a_0 + a_1\lambda + a_2\lambda^2 + a_3\lambda^3 = \varphi_k(h\lambda),$$

there are several cases to consider.

If either $R = 0$ or $P^2 = 4R$, then the eigenvalues of $\boldsymbol{A}$ are

$$\lambda = 0, 0, \pm i\,\mu,$$

where $\mu = \sqrt{P}$. For $\varphi_k = \varphi_1$ the polynomial coefficients are:

$$\begin{aligned}
a_0 &= 1,\\
a_1 &= \frac{h}{2},\\
a_2 &= h^2\,\frac{h\mu - \sin(h\mu)}{(h\mu)^2},\\
a_3 &= h^3\,\frac{\cos(h\mu) - 1 + (h\mu)^2/2}{(h\mu)^6}.
\end{aligned}$$

For $\varphi_k = \varphi_3$ the polynomial coefficients are:

$$\begin{aligned}
a_0 &= \frac{1}{6},\\
a_1 &= \frac{h}{24},\\
a_2 &= h^2\,\frac{\sin(h\mu) + (h\mu)^3/6 - h\mu}{(h\mu)^5},\\
a_3 &= h^3\,\frac{1 - (h\mu)^2/2 + (h\mu)^4/24 - \cos(h\mu)}{(h\mu)^6}.
\end{aligned}$$

If $R \neq 0$, then the eigenvalues of $\boldsymbol{A}$ are

$$\lambda = \pm i\,\mu, \pm i\,\nu,$$

where

$$\mu = \sqrt{\frac{P + \sqrt{P^2 - 4R}}{2}} \qquad \text{and} \qquad \nu = \sqrt{\frac{P - \sqrt{P^2 - 4R}}{2}}.$$

For $\varphi_k = \varphi_1$ the polynomial coefficients are:

$$a_0 = \frac{1}{\nu^2 - \mu^2}\left(\nu^2 \frac{\sin(h\mu)}{h\mu} - \mu^2 \frac{\sin(h\nu)}{h\nu}\right),$$

$$a_1 = \frac{1}{\nu^2 - \mu^2}\left(\nu^2 \frac{1-\cos(h\mu)}{h\mu} - \mu^2 \frac{1-\cos(h\nu)}{h\nu}\right),$$

$$a_2 = \frac{1}{\nu^2 - \mu^2}\left(\frac{\sin(h\mu)}{h\mu} - \frac{\sin(h\nu)}{h\nu}\right),$$

$$a_3 = \frac{1}{\nu^2 - \mu^2}\left(\frac{1-\cos(h\mu)}{h\mu} - \frac{1-\cos(h\nu)}{h\nu}\right).$$

For $\varphi_k = \varphi_3$ the polynomial coefficients are:

$$a_0 = \frac{1}{h^2(\nu^2 - \mu^2)}\left(\frac{\nu^2}{\mu^2}\left(1 - \frac{\sin(h\mu)}{h\mu}\right) - \frac{\mu^2}{\mu^2}\left(1 - \frac{\sin(h\nu)}{h\nu}\right)\right),$$

$$a_1 = \frac{1}{h^2(\nu^2 - \mu^2)}\left(\frac{\nu^2}{\mu^2}\left(\frac{h}{2} - \frac{1-\cos(h\mu)}{h\mu}\right) - \frac{\mu^2}{\mu^2}\left(\frac{h}{2} - \frac{1-\cos(h\nu)}{h\nu}\right)\right),$$

$$a_2 = \frac{1}{h^2(\nu^2 - \mu^2)}\left(\frac{1}{\mu^2}\left(1 - \frac{\sin(h\mu)}{h\mu}\right) - \frac{1}{\mu^2}\left(1 - \frac{\sin(h\nu)}{h\nu}\right)\right),$$

$$a_3 = \frac{1}{h^2(\nu^2 - \mu^2)}\left(\frac{1}{\mu^2}\left(\frac{h}{2} - \frac{1-\cos(h\mu)}{h\mu}\right) - \frac{1}{\mu^2}\left(\frac{h}{2} - \frac{1-\cos(h\nu)}{h\nu}\right)\right).$$

B.1.2 Grad-B Drift Problem

For the grad-B drift test problem, $R = 0$ and the characteristic polynomial reduces to

$$\lambda^4 + \lambda^2 P + \lambda Q = \lambda(\lambda^3 + \lambda P + Q).$$

Hence, the eigenvalues of $\boldsymbol{A}$ are

$$\lambda = 0, \mu, \nu, \overline{\nu},$$

where μ is the real root and the conjugate pair $\nu, \overline{\nu}$ are the complex roots of the cubic polynomial $\lambda^3 + \lambda P + Q$. The polynomial coefficients are thus:

$$a_0 = \varphi_k(0),$$

$$a_1 = \frac{|\nu|^4\varphi_k(h\mu)\mathrm{Im}(\nu) + \mu^2\mathrm{Im}(\overline{\nu}^3\varphi_k(h\nu)) + \mu^3\mathrm{Im}(\overline{\nu}^2\varphi_k(h\nu))}{\mu|\nu|^2\mathrm{Im}(\nu)(|\nu|^2 - 2\mu\,\mathrm{Re}(\nu) + \mu^2)},$$

$$a_2 = -\frac{\mu\,\mathrm{Im}(\overline{\nu}^3\varphi_k(h\nu)) + 2|\nu|^2\varphi_k(h\mu)\mathrm{Re}(\nu)\mathrm{Im}(\nu) - \mu^3\mathrm{Im}(\overline{\nu}\varphi_k(h\nu))}{\mu|\nu|^2\mathrm{Im}(\nu)(|\nu|^2 - 2\mu\,\mathrm{Re}(\nu) + \mu^2)},$$

$$a_3 = \frac{\mu\,\mathrm{Im}(\overline{\nu}^2\varphi_{\mathrm{k}}(\mathrm{h}\nu)) - \mu^2\mathrm{Im}(\overline{\nu}\varphi_{\mathrm{k}}(\mathrm{h}\nu)) + |\nu|^2\varphi_{\mathrm{k}}(\mathrm{h}\mu)\mathrm{Im}(\nu)}{\mu|\nu|^2\mathrm{Im}(\nu)(|\nu|^2 - 2\mu\,\mathrm{Re}(\nu) + \mu^2)},$$

where $\mathrm{Re}(z)$ and $\mathrm{Im}(z)$ denote the real and imaginary parts of the complex argument z, respectively.

B.2 Three Dimensional Model

For the three dimensional electric potential well test problems, the block matrices $\boldsymbol{H}$ and $\boldsymbol{\Omega}$ are given by

$$\boldsymbol{H} = \frac{\partial \boldsymbol{f}_L}{\partial \boldsymbol{x}} = \begin{bmatrix} H_{11} & 0 & 0 \\ 0 & H_{22} & 0 \\ 0 & 0 & H_{33} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\Omega} = \begin{bmatrix} 0 & \omega & 0 \\ -\omega & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \omega = \frac{qB}{m}.$$

The characteristic polynomial of $\boldsymbol{A}$ is

$$\det(\lambda \boldsymbol{I}_{6\times 6} - \boldsymbol{A}) = \lambda^6 + \lambda^4 P + \lambda^2 R + T$$

where

$$\begin{aligned} P &= \omega^2 - H_{11} - H_{22} - H_{33}, \\ R &= H_{11}H_{22} + H_{11}H_{33} + H_{22}H_{33} - \omega^2 H_{33}, \\ T &= -H_{11}H_{22}H_{33}. \end{aligned}$$

Again we assume strongly magnetized particle pushing problems implying P is always nonzero. To determine the polynomial coefficients, we next examine the various cases.

If $R = T = 0$, then the characteristic polynomial reduces to

$$\lambda^6 + \lambda^4 P = \lambda^4(\lambda^2 + P).$$

Thus, the eigenvalues of $\boldsymbol{A}$ are

$$\lambda = 0, 0, 0, \pm i\,\mu,$$

where $\mu = \sqrt{P}$. Hence, for $\varphi_k = \varphi_1$ the polynomial coefficients are:

$$\begin{aligned} a_0 &= 1, \\ a_1 &= \frac{h}{2}, \\ a_2 &= \frac{h^2}{6}, \\ a_3 &= \frac{h^3}{24}, \\ a_4 &= \frac{1}{\mu^4}\left(\frac{\sin(h\mu)}{h\mu} - 1 + \frac{(h\mu)^2}{6}\right), \\ a_5 &= \frac{1}{\mu^5}\left(\frac{1-\cos(h\mu)}{h\mu} - \frac{h\mu}{2} + \frac{(h\mu)^3}{24}\right). \end{aligned}$$

For $\varphi_k = \varphi_3$ the polynomial coefficients are:

$$\begin{aligned}
a_0 &= \frac{1}{6}, \\
a_1 &= \frac{h}{24}, \\
a_2 &= \frac{h^2}{60}, \\
a_3 &= \frac{h^3}{120}, \\
a_4 &= \frac{1}{\mu^4}\left(\frac{h\mu - \sin(h\mu)}{(h\mu)^3} - \frac{1}{6} + \frac{h\mu}{60}\right), \\
a_5 &= \frac{1}{\mu^5}\left(\frac{(h\mu)^2/2 + \cos(h\mu) - 1}{(h\mu)^3} - \frac{h\mu}{24} + \frac{(h\mu)^2}{120}\right).
\end{aligned}$$

If $R \neq 0$ and $T = 0$, then the characteristic polynomial reduces to

$$\lambda^6 + \lambda^4 P + \lambda^2 R = \lambda^2(\lambda^4 + \lambda^2 P + R).$$

In this case, the eigenvalues are

$$\lambda = 0, 0, \pm i\,\mu, \pm i\,\nu,$$

where

$$\mu = \sqrt{\frac{P - \sqrt{P^2 - 4R}}{2}} \quad \text{and} \quad \nu = \sqrt{-\frac{P + \sqrt{P^2 - 4R}}{2}}.$$

For $\varphi_k = \varphi_1$ the polynomial coefficients are:

$$\begin{aligned}
a_0 &= 1, \\
a_1 &= \frac{h}{2}, \\
a_2 &= \frac{\nu^2 S_1(h\mu) - \mu^2 S_1(h\nu)}{\mu^2 - \nu^2}, \\
a_3 &= \frac{\nu^2 C_1(h\mu) - \mu^2 C_1(h\nu)}{\mu^2 - \nu^2}, \\
a_4 &= \frac{S_1(h\mu) - S_1(h\nu)}{\mu^2 - \nu^2}, \\
a_5 &= \frac{C_1(h\mu) - C_1(h\nu)}{\mu^2 - \nu^2},
\end{aligned}$$

where

$$S_1(z) = \frac{h^2}{z^2}\left(1 - \frac{\sin(z)}{z}\right) \quad \text{and} \quad C_1(z) = \frac{h^3}{z^3}\left(\frac{z^2}{2} - \frac{1-\cos(z)}{z}\right).$$

For $\varphi_k = \varphi_3$ the polynomial coefficients are:

$$\begin{aligned}
a_0 &= \frac{1}{6}, \\
a_1 &= \frac{h}{24}, \\
a_2 &= \frac{1}{\mu^2 - \nu^2}\left(\frac{\mu^2}{\nu^2}\left(\frac{1}{6} - S_3(h\nu)\right) - \frac{\nu^2}{\mu^2}\left(\frac{1}{6} - S_3(h\mu)\right)\right), \\
a_3 &= \frac{1}{\mu^2 - \nu^2}\left(\frac{\mu^2}{\nu^3}\left(\frac{h\nu}{24} - C_3(h\nu)\right) - \frac{\nu^2}{\mu^3}\left(\frac{h\mu}{24} - C_3(h\mu)\right)\right), \\
a_4 &= \frac{S_3(h\mu) - S_3(h\nu)}{\mu^2 - \nu^2}, \\
a_5 &= \frac{1}{\mu^2 - \nu^2}\left(\left(\frac{h\nu}{24} - C_3(h\nu)\right) - \left(\frac{h\mu}{24} - C_3(h\mu)\right)\right),
\end{aligned}$$

where

$$S_3(z) = \frac{z - \sin(z)}{z^3} \quad \text{and} \quad C_3(z) = \frac{z^2/2 + \cos(z) - 1}{z^3}.$$

If $R, T \neq 0$, then the characteristic polynomial can be expressed as a cubic form

$$w^3 + w^2\, P + w\, R + T, \qquad \text{where } w = \lambda^2.$$

Let $-\mu^2, -\nu^2, -\xi^2$ be the three roots of the cubic polynomial above. Then the eigenvalues of $\boldsymbol{A}$ are

$$\lambda = \pm i\,\mu, \pm i\,\nu, \pm i\,\xi.$$

The interpolation polynomial coefficients are:

$$a_0 = \frac{1}{\Delta_1\Delta_2\Delta_3}\left(\nu^2\xi^2\Delta_3\frac{\sin(h\mu)}{h\mu} - \mu^2\xi^2\Delta_2\frac{\sin(h\nu)}{h\nu} + \mu^2\nu^2\Delta_1\frac{\sin(h\xi)}{h\xi}\right),$$

$$a_1 = \frac{1}{\Delta_1\Delta_2\Delta_3}\left(\nu^2\xi^2\Delta_3\frac{1-\cos(h\mu)}{h\mu} - \mu^2\xi^2\Delta_2\frac{1-\cos(h\nu)}{h\nu} + \mu^2\nu^2\Delta_1\frac{1-\cos(h\xi)}{h\xi}\right),$$

$$a_2 = \frac{1}{\Delta_1\Delta_2\Delta_3}\left((\nu^4-\xi^4)\Delta_3\frac{\sin(h\mu)}{h\mu} - (\mu^4-\xi^4)\Delta_2\frac{\sin(h\nu)}{h\nu} + (\mu^4-\nu^4)\Delta_1\frac{\sin(h\xi)}{h\xi}\right),$$

$$a_3 = \frac{1}{\Delta_1\Delta_2\Delta_3}\Bigg((\nu^4-\xi^4)\Delta_3\frac{1-\cos(h\mu)}{h\mu} - (\mu^4-\xi^4)\Delta_2\frac{1-\cos(h\nu)}{h\nu} + (\mu^4-\nu^4)\Delta_1\frac{1-\cos(h\xi)}{h\xi}\Bigg),$$

$$a_4 = \frac{1}{\Delta_1\Delta_2\Delta_3}\left(\Delta_3\frac{\sin(h\mu)}{h\mu} - \Delta_2\frac{\sin(h\nu)}{h\nu} + \Delta_1\frac{\sin(h\xi)}{h\xi}\right),$$

$$a_5 = \frac{1}{\Delta_1\Delta_2\Delta_3}\left(\Delta_3\frac{1-\cos(h\mu)}{h\mu} - \Delta_2\frac{1-\cos(h\nu)}{h\nu} + \Delta_1\frac{1-\cos(h\xi)}{h\xi}\right),$$

where $\Delta_1 = \mu^2 - \nu^2$, $\Delta_2 = \mu^2 - \xi^2$, and $\Delta_3 = \nu^2 - \xi^2$.

References

[1] C.K. Birdsall and A.B. Langdon. Plasma Physics via Computer Simulation. Series in Plasma Physics. CRC Press, 2018.

[2] J.P. Boris. Relativistic plasma simulation-optimization of a hybrid code. Proceeding of Fourth Conference on Numerical Simulations of Plasmas, November 1970.

[3] J.U. Brackbill and D.W. Forslund. 9 - Simulation of low-frequency, electromagnetic phenomena in plasmas. In J.U. Brackbill and B.I. Cohen, editors, Multiple Time Scales, pages 271–310. Academic Press, 1985.

[4] L. Brieda. Plasma Simulations by Example. CPC Press, 2019.

[5] A. Buchheim. Xx. an extension of a theorem of Professor Sylvester's relating to matrices. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 22(135):173–174, 1886.

[6] O. Buneman. Time-reversible difference procedures. Journal of Computational Physics, 1(4):517–535, 1967.

[7] R.L. Burden, D.J. Faires, and A.M. Burden. Numerical Analysis, 10th ed. Cengage Learning, 2016.

[8] M. Caliari, P. Kandolf, A. Ostermann, and S. Rainer. The Leja method revisited: Backward error analysis for the matrix exponential. SIAM Journal on Scientific Computing, 38(3):A1639–A1661, 2016.

[9] F.F. Chen. Introduction to Plasma Physics and Controlled Fusion. Second edition. Volume 1: Plasma Physics. Plenum Press, 12 1984.

[10] G. Chen and L. Chacón. An implicit, conservative and asymptotic-preserving electrostatic particle-in-cell algorithm for arbitrarily magnetized plasmas in uniform magnetic fields. Journal of Computational Physics, 487:112160, 2023.

[11] R.H. Cohen, A. Friedman, D.P. Grote, and J.-L. Vay. Large-timestep mover for particle simulations of arbitrarily magnetized species. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors

and Associated Equipment, 577(1):52–57, 2007. Proceedings of the 16th International Symposium on Heavy Ion Inertial Fusion.

[12] L. Einkemmer, M. Tokman, and J. Loffeld. On the performance of exponential integrators for problems in magnetohydrodynamics. Journal of Computational Physics, 330:550–565, 2016.

[13] F. Filbet and L.M. Rodrigues. Asymptotically stable particle-in-cell methods for the vlasov–poisson system with a strong external magnetic field. SIAM Journal on Numerical Analysis, 54(2):1120–1146, 2016.

[14] F. Filbet and L.M. Rodrigues. Asymptotically preserving particle-in-cell methods for inhomogeneous strongly magnetized plasmas. SIAM Journal on Numerical Analysis, 55(5):2416–2443, 2017.

[15] F.R. Gantmacher. The Theory of Matrices, Volume I. Chelsea Publishing, 1959.

[16] S. Gaudreault, G. Rainwater, and M. Tokman. KIOPS: A fast adaptive Krylov subspace solver for exponential integrators. Journal of Computational Physics, 372:236–255, 2018.

[17] T.C. Genoni, R.E. Clark, and D.R. Welch. A fast implicit algorithm for highly magnetized charged particle motion. The Open Plasma Physics Journal, 3:36–41, 2014.

[18] G. Guennebaud, B. Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.

[19] E. Hairer, C. Lubich, and B. Wang. A filtered Boris algorithm for charged particle-dynamics in a strong magnetic field. Numerische Mathematik, 144:787–809, 2020.

[20] E. Hairer, C. Lubich, and G. Wanner. Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations. Springer-Verlag, 2002.

[21] M. Hochbruck and A. Ostermann. Exponential integrators. Acta Numerica, 19:209–286, 2010.

[22] R.W. Hockney and J.W. Eastwood. Computer Simulation Using Particles. Adam Hilger, 1988.

[23] U.S. Inan and M. Gołkowski. Principles of Plasma Physics for Engineers and Scientists. Cambridge University Press, 2010.

[24] I. Joseph. Guiding center and gyrokinetic orbit theory for large electric field gradients and strong shear flows. Physics of Plasmas, 28(4):042102, 04 2021.

[25] A. Kassam and L.N. Trefethen. Fourth-order time-stepping for stiff PDEs. SIAM Journal on Scientific Computing, 26(4):1214–1233, 2005.

[26] D. Kincaid and W. Cheney. Numerical Analysis: Mathematics of Scientific Computing, 10th ed. AMS, 2002.

[27] B. Leimkuhler and S. Reich. Simulating Hamiltonian Dynamics. Cambridge University Press, 2004.

[28] C. Moler and C. Van Loan. Nineteen dubious ways to compute the exponential of a matrix. SIAM Review, 20(4):801–836, 1978.

[29] C. Moler and C. Van Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. SIAM Review, 45(1):3–49, 2003.

[30] D.R. Nicholson. Introduction to Plasma Theory. Wiley, 1983.

[31] J. Niesen and W.M. Wright. Algorithm 919: A Krylov subspace algorithm for evaluating the φ-functions appearing in exponential integrators. ACM Transactions on Mathematical Software, 38(3), apr 2012.

[32] E.J. Nyström. Über die numerische integration von differentialgleichungen. Acta Societas Scientiarum Fennicae, 50(13):1–54, 1925.

[33] S.E Parker and C.K Birdsall. Numerical error in electron orbits with large omega delta t. Journal of Computational Physics, 97(1):91–102, 1991.

[34] H. Qin, S. Zhang, J. Xiao, J. Liu, Y. Sun, and W.M. Tang. Why is Boris algorithm so good? Physics of Plasmas, 20(8):084503, 08 2013.

[35] L.F. Ricketson and L. Chacón. An energy-conserving and asymptotic-preserving charged-particle orbit implicit time integrator for arbitrary electromagnetic fields. Journal of Computational Physics, 418:109639, 2020.

[36] J.M. Sanz-Serna and M.P. Calvo. Numerical Hamiltonian Problems. Dover Publications, 2018.

[37] J. Stewart, M. Tokman, V. Dallerit, F. Bisetti, and O. Diaz-Ibarra. Variable time-stepping exponential integrators for chemical reactors with analytical jacobians. Applied and Computational Mathematics, 13(2):29–37, 2024.

[38] J.J. Sylvester. Xxxix. on the equation to the secular inequalities in the planetary theory. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 16(100):267–269, 1883.

[39] M. Tokman. Efficient integration of large stiff systems of ODEs with exponential propagation iterative (epi) methods. Journal of Computational Physics, 213(2):748–776, 2006.

[40] M. Tokman. A new class of exponential propagation iterative methods of Runge–Kutta type (EPIRK). Journal of Computational Physics, 230(24):8762–8778, 2011.

[41] H.X. Vu and J.U. Brackbill. Accurate numerical solution of charged particle motion in a magnetic field. Journal of Computational Physics, 116(2):384–387, 1995.