Spectral deferred correction methods for second-order problems

Vom Promotionsausschuss der
Technischen Universität Hamburg

zur Erlangung des akademischen Grades

Doktor der Naturwissenschaften (Dr. rer. nat.)

genehmigte Dissertation (Monographie)

von
Ikrom Akramov

aus
Bukhara, Uzbekistan

2025

1. Gutachter: Prof. Dr. Daniel Ruprecht
2. Gutachter: Prof. Dr. Giovanni Samaey

Tag der mündlichen Prüfung: 16. April 2025

doi:10.15480/882.15091
ORCiD: https://orcid.org/0000-0002-8869-0784

Summary

Spectral deferred corrections (SDC) are a class of iterative methods for the numerical solution of ordinary differential equations (ODEs). SDC can be interpreted as a Picard iteration to solve a fully implicit collocation problem, preconditioned with a low-order method. It has been widely studied for first-order problems with different preconditioners. While numerical results for SDC applied to the second-order Lorentz equations exist, no theoretical results are available for SDC applied to second-order problems.

This thesis presents a comprehensive study of SDC for second-order problems. In Chapter 2, we explain how to obtain spectral deferred correction method (SDC) for second-order problems by combining the collocation problem and preconditioner, obtained using the velocity-Verlet scheme, after applying Picard iteration. Moreover, we introduce Boris-SDC for the Lorentz equation, which combines the classical Boris-integration with SDC iteration for the collocation formulation of second-order problems. Using Boris-SDC for the Lorentz equations allows the implicit system in the SDC iteration to be solved explicitly using the Boris integrator.

In Chapter 3, we develop a theoretical framework for the second-order SDC. Using the damped harmonic oscillator as a test problem, we analyze convergence and stability domains, comparing our SDC method against a collocation approach with Picard iteration and a Runge-Kutta-Nyström-4 (RKN-4) method. We also compare the work precision of SDC against the Picard and RKN-4. Our convergence results show that each SDC iteration increases the global convergence order by one when the force depends on the velocity and by two when it does not. Moreover, we prove that the local position error order exceeds that of the local velocity error by one. These theoretical predictions are validated through numerical experiments.

Chapter 4 introduces a multi-level version of second-order SDC known as multi-level SDC (MLSDC). It performs the SDC sweeps on a hierarchy of discretization levels, with a Full Approximation Scheme (FAS) correction used to enhance coarse-level accuracy. Two strategies to reduce computational cost on the coarse level are investigated: using fewer collocation nodes and solving a macroscopic model instead of the microscopic one on the coarse level. Numerical results for the former strategy illustrate that MLSDC achieves convergence rates comparable to SDC while reducing computational overhead.

In Chapter 5, using perturbation analysis, we derive macroscopic models for classical second-order systems, such as the Duffing and harmonic oscillator problems, and present existing results for the Lorentz equation. Numerical examples show that the accuracy of these macroscopic models improves as the small parameter ε approaches zero.

Chapter 6 introduces the micro–macro MLSDC (M3LSDC) method. This approach uses the second strategy for coarsening in MLSDC where, instead of solving the full detailed (microscopic) model on the coarse grid, a simplified (macroscopic) model is used. We derive the associated FAS correction and propose suitable transfer operators. Numerical experiments reveal that the M3LSDC reaches almost the same accuracy as the MLSDC for the Duffing equation.

In Chapter 7, we finally sum up the results of the previous chapters and provide an outlook on possible future work related to this thesis.

Acknowledgments

I would like to thank several people who have supported me in one way or another throughout my journey, ultimately helping me to complete this thesis. First and foremost, I am very grateful for the excellent guidance and support from my supervisor, Daniel Ruprecht, and my co-supervisor, Sebastian Götschel. They were always welcoming whenever I had questions or sought advice, which I deeply appreciate. Moreover, I would like to thank Giovanni Samaey for serving as the second referee of my thesis and for the kind invitation to the lovely city of Leuven to discuss our project.

Furthermore, I would like to thank all of my colleagues at the Institute of Mathematics at TUHH. Their support during my PhD was invaluable. Special thanks go to Thibaut Lunet, Vamika Rathi, Judith Angel, and Julio Urizarna. We had many engaging discussions and shared great times together at conferences and during coffee breaks. My appreciation also extends to all other colleagues at the Chair of Computational Mathematics .

Finally, I want to express my heartfelt gratitude to my friends and family, who accompanied and supported me throughout this journey—or even started it, by encouraging me to study mathematics in the first place. Special thanks to my wife, Dilnora Qurbonova, who has been incredibly supportive during my PhD journey, especially in the final phase of writing this dissertation. And a final thanks to developers of ChatGPT—for helping me with grammar checks while writing my thesis!

Contents

List of Figures

Chapter 1

Introduction

Many problems in science and engineering involve modeling motion based on Newton's second law. This leads to initial value problems (IVP) of the form

$$\ddot{x} = f(t, x(t), \dot{x}(t)), \quad x(t_0) = x_0, \quad \dot{x}(t_0) = \dot{x}_0, \tag{1.1}$$

where $x : \mathbb{R} \to \mathbb{R}^d$ and

$$f : \mathbb{R} \times \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$$

are defined for $t_0 \leq t \leq t_{\text{end}}$. Since analytical solutions are often difficult to obtain, numerical time-stepping algorithms are used to approximate the solution.

A common strategy for solving such second-order ordinary differential equations (ODEs) is to reformulate them as a system of first-order ODEs by introducing the velocity variable $v(t) = \dot{x}(t)$. This yields

$$\dot{x}(t) = v(t), \tag{1.2a}$$
$$\dot{v}(t) = f(t, x(t), v(t)), \tag{1.2b}$$

which can then be solved using standard methods such as Runge-Kutta [9, Chapter II] or multistep schemes [9, Chapter III]. However, treating the position and velocity equations identically may overlook opportunities to improve the performance of the numerical method.

In many cases, standard methods do not preserve important geometric properties of the original second-order systems, such as stability and energy conservation. For example, consider the harmonic oscillator where

$$f(t, x, v) = -x.$$

The explicit Euler method applied to this problem is unconditionally unstable, leading to growing errors over time, while the implicit Euler method introduces significant numerical damping, causing the system's energy to decay. In contrast, the symplectic Euler method, which integrates the position equation explicitly and the velocity equation implicitly (with no additional implicit solver required if f is independent of v, is conditionally stable and better at conserving energy [10].

Building on these ideas, Runge-Kutta-Nyström (RKN) methods have been developed that employ different Butcher tables for the position and velocity components [11, Section

II.14]. A widely studied special case [12, 13, 14, 15] is where $f(t, x(t), v(t)) = f(t, x(t))$, that is, the right hand side depends only on the position but not the velocity. This greatly simplifies the order condition, allowing for the construction of high-order methods with favorable properties [9, Sec. II.2]. For the general case, however, constructing high-order methods remains challenging, and RKN methods often struggle to outperform standard Runge-Kutta methods designed for first-order systems [16].

An alternative approach is spectral deferred corrections methods (SDC), introduced by Dutt et al. [17]. SDC provides a straightforward framework for constructing high-order solvers for the first-order problems, and its development has been supported by a significant amount of theoretical work [18, 19, 20] as well as algorithmic improvements [21, 22, 23] and applications to complex problems [24, 25]. Moreover, a variant of SDC method is so-called multi-level SDC (MLSDC) method introduced by Speck et al. [26], inspired by the Full Approximation Scheme (FAS) coming from nonlinear multigrid method [4]. While SDC itself is robust and capable of high-order accuracy, performing all correction sweeps on a single (fine) level may be computationally expensive, especially large-scale problems or multi-dimensional partial differential equations (PDEs) [26, 27], this variation was designed to improve the efficiency of the SDC method by shifting some of the work to coarse, less expensive levels. However, the overall improvements in convergence rate and runtime efficiency can depend on the problem specifics and the chosen coarsening strategy. The paper by Kremling and Speck [28] has provided the convergence proofs and analysis of MLSDC method for first-order problems, showing that the MLSDC can, under appropriate conditions, achieve similar accuracy as SDC while reducing computational overhead.

For second-order problems, a specialized variant based on the Boris integrator [29], known as Boris-SDC, has been proposed specifically for the Lorentz equations [30, Chapter 5] that model the trajectories of charged particles in electromagnetic fields [3]. This method has been further refined [31], applied to compute fast ion trajectories in fusion reactors [32], and extended to other plasma physics problems [33]. However, no attempt has yet been made to generalize the SDC framework to second-order problems beyond the Lorentz system, and unlike in the first-order case, a comprehensive theoretical foundation remains to be established. Furthermore, the MLSDC framework has not yet been extended to the second-order problems.

In the first part of this work, we fill these gaps by providing a systematic study of the theoretical properties of second-order SDC, including proof of consistency and an assessment of stability. We study the convergence and demonstrate that SDC can compete with RKN-4 method [11, pp. 284-285] in terms of computational efficiency [1]. Following this, we introduce the MLSDC method for second-order problems, present the algorithm, and compare its residuals and numerical convergence rate against the second-order SDC.

In many applications, a system is modeled a high-dimensional system of ODEs that captures phenomena occurring at multiple time scale. Unfortunately, the computational cost of simulating such fast or small-scale systems (which we call *microscopic models*) on macroscopic time intervals is prohibitive and one often simplifies these systems by reducing to low-dimensional, coarse-grained, effective models (which we call *macroscopic models*), in which the fast degrees of freedom are eliminated. Many methods have been proposed to obtain these macroscopic models, either analytically or numerically [34]. For instance, heterogeneous multiscale methods [35, 36] analyze numerical approaches that combine the macroscopic and microscopic models even when explicit macroscopic models are not

available. Similar ideas have been extended into the parareal framework for singularly perturbed ODEs [8, 37, 38, 39, 40] and have also been applied in stochastic differential equations [41]. However, the micro-macro framework has not yet been applied within SDC or MLSDC methods. Standard SDC and MLSDC treat all scales equally, which can lead to inefficiencies when fast (*micro*) scale processes force very small time steps while the slow (*macro*) scale permits larger ones.

In the second part of the thesis, we extend the micro-macro framework to the SDC methods. A new approach is so-called the *micro-macro MLSDC (M3LSDC)* is designed to separate the fast (micro) dynamics from the slow (macro) one and integrate them using different strategies. By splitting the dynamics into micro and macro components, the method enables performing most of the computationally expensive corrections on a coarser level while handling the detailed fast dynamics on a finer level. This adaptive allocation of computational work significantly reduces the cost on the coarse level.

The remainder of this thesis is structured as follows. Chapter 2 describes the SDC method for second-order problems using velocity-Verlet integrator as base method. Chapter 3 investigates stability, using the damped harmonic oscillator as a test problem. The related issues of stability and convergence of the SDC iteration are discussed and stability domains of SDC are compared against an RKN-4 method and a collocation problem using Picard iteration. We provide proof of consistency and that each iteration increases the order by two in the case where f does not depend on v but only by one if it does. These theoretical results verified by numerical examples. In Chapter 4, the MLSDC method for second-order problems is introduced and its algorithm is presented. Its residuals and convergence rate are compared against the second-order SDC method. Chapter 5 explains how macroscopic models are derived from singularly and regularly perturbed ODEs using perturbation analysis for specific problems, and also discusses the importance of these macroscopic models with numerical examples. In Chapter 6, we introduce our new M3LSDC method and discuss its differences from the MLSDC method, again comparing residuals and convergence order with those of SDC and MLSDC through numerical examples. Finally, Chapter 7 summarizes the results of the previous chapters and provides an outlook on possible future work related to this thesis.

Chapter 2

Spectral deferred correction method for second-order problem

Spectral Deferred Correction (SDC) is a class of iterative methods for solving ordinary differential equations (ODEs) [17]. These methods can be interpreted as preconditioned fixed-point iterations that solve the collocation problem. In this chapter, we explain the fundamentals of SDC for second-order ODEs, as introduced in [1], and analyze the Boris-SDC method [3], which combines the Boris integrator with SDC and is specifically designed to solve the Lorentz equations [30, Chapter 5].

For notational simplicity, we focus on the autonomous case of equation (1.2), since any non-autonomous problem can be rewritten as an equivalent autonomous problem [42, pages 6–7]. The autonomous form of equation (1.2) is given by

$$\dot{x}(t) = v(t), \qquad x(t_0) = x_0, \tag{2.1a}$$

$$\dot{v}(t) = f(x(t), v(t)), \qquad v(t_0) = \dot{x}_0 =: v_0. \tag{2.1b}$$

To ensure the existence and uniqueness of the solution to (2.1), the function f requires being Lipschitz continuous.

Definition 2.1. A function $f\colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ in (2.1) is called *Lipschitz continuous* if there exists a constant $L > 0$ such that

$$\|f(x_1, v_1) - f(x_2, v_2)\| \leq L\left(\|x_1 - x_2\| + \|v_1 - v_2\|\right) \tag{2.2}$$

for all $x_i, v_i \in \mathbb{R}^d$ for $i = 1, 2$. Here, $\|\cdot\|$ denotes the infinity norm in $\mathbb{R}^d$.[1].

We first derive the collocation problem for the system in (2.1). Next, we develop a method to solve this collocation problem. To efficiently apply Richardson iteration [43, Chapter 1], we introduce a low-order method as a preconditioner. By combining the collocation problem with this preconditioner, we then apply Richardson iteration to derive the SDC method for second-order problems.

[1]Throughout this thesis, We use the notation $\|\cdot\|$ to denote the infinity norm of a vector $x \in \mathbb{R}^d$, defined as $\|x\|_\infty = \max_{1\leq i\leq n} |x_i|$, or a matrix $A \in \mathbb{R}^{d\times d}$, defined as $\|A\|_\infty = \max_{1\leq i\leq d} \sum_{j=1}^{d} |a_{ij}|$.

2.1 Collocation problem

In this section, we derive the collocation problem for (2.1). As usual when solving ODEs, the time range $[t_0, t_{\text{end}}]$ is divided into smaller time intervals

$$t_0 \leq t_1 < t_2 < \cdots < t_n < t_{n+1} \leq t_{\text{end}}$$

with step size $\Delta t := t_{n+1} - t_n$ and the solution is successively calculated for each of these time steps. Thus, it is sufficient to develop a method for a given time step. Thus, we will only consider a single time step throughout the dissertation, i.e., the initial value problem defined on the interval $[t_n, t_{n+1}]$.

Then, the collocation problem is obtained by integrating (2.1) over the time interval $[t_n, t_{n+1}]$

$$x(t) = x_0 + \int_{t_n}^{t} v(s)ds, \tag{2.3a}$$

$$v(t) = v_0 + \int_{t_n}^{t} f(x(s), v(s))ds, \tag{2.3b}$$

where $t \in [t_n, t_{n+1}]$ with the initial values $x_0 \approx x(t_n)$ and $v_0 \approx v(t_n)$ indicate that x_0 and v_0 are approximations of the exact values $x(t_n)$ and $v(t_n)$, respectively[2]. Furthermore, a set of M quadrature nodes $\tau_1, \ldots, \tau_M$ within the time interval are defined, such that

$$t_n \leq \tau_1 < \cdots < \tau_M \leq t_{n+1}.$$

and the weights are defined by

$$\Delta t q_{m,j} = \Delta t \int_0^{e_m} l_j(s)ds = \int_{t_n}^{\tau_m} \bar{l}_j(s)ds, \quad m, j = 1, \ldots M,$$

where $e_m := (\tau_m - t_n)/\Delta t$. $l_j(s)$ and $\bar{l}_j(s)$, $j = 1, \ldots, M$ are Lagrange interpolation polynomials corresponding to the quadrature nodes on the interval $[0, 1]$ and $[t_n, t_{n+1}]$ respectively. Evaluating (2.3) at the quadrature nodes ($t = \tau_m$, for $m = 1, \ldots, M$) gives the following system of integral equations

$$x(\tau_m) = x_0 + \int_{t_n}^{\tau_m} v(s)ds, \tag{2.4a}$$

$$v(\tau_m) = v_0 + \int_{t_n}^{\tau_m} f(x(s), v(s))ds \tag{2.4b}$$

for $m = 1, \ldots, M$. Now, we denote x_j, v_j and f_j as the approximation to $x(\tau_j), v(\tau_j)$ and $f(x(\tau_j), v(\tau_j))$ correspondingly [11, pp. 211-214] and apply polynomial approximation to the system of integral equations (2.4) [11, Theorem 7.8] which results in

$$x_m = x_0 + \Delta t \sum_{j=1}^{M} q_{m,j} v_j, \tag{2.5a}$$

[2]Since we are considering the time step $[t_n, t_{n+1}]$, we denote the initial values as $x_0 \approx x(t_n)$ and $v_0 \approx v(t_n)$. In particular, the time interval usually starts with $[t_0, t_1]$, then the initial values are given by $x_0 \approx x(t_0)$ and $v_0 \approx v(t_0)$.

$$v_m = v_0 + \Delta t \sum_{j=1}^{M} q_{m,j} f_j \tag{2.5b}$$

for $m = 1, \ldots, M$. Next, we substitute equation (2.5b) into the first equation (2.5a) and we obtain combined collocation problem

$$x_m = x_0 + \Delta t \sum_{j=1}^{M} q_{m,j} v_0 + \Delta t^2 \sum_{j=1}^{M} q q_{m,j} f_j, \tag{2.6a}$$

$$v_m = v_0 + \Delta t \sum_{j=1}^{M} q_{m,j} f_j \tag{2.6b}$$

where $qq_{m,j} = \sum_{i=1}^{M} q_{m,i} q_{i,j}$ and $m = 1, \ldots, M$. The x_m, v_m correspond to the stages of a fully implicit RKN method [11, pp. 283-300].

2.1.1 Matrix form of collocation problem

For the convergence and stability analysis in Chapter 3, it is convenient to work in the matrix form of the collocation problem. For this reason, we will write the $2M$ coupled equations (2.6) as a single system of equations like in [1, 3].

Let $\bar{Q} \in \mathbb{R}^{M \times M}$ have entries $q_{m,j}$, which means

$$\bar{Q} = \begin{pmatrix} q_{11} & q_{12} & \cdots & q_{1M} \\ q_{21} & q_{22} & \cdots & q_{2M} \\ \vdots & \vdots & \ldots & \vdots \\ q_{M1} & q_{M2} & \cdots & q_{MM} \end{pmatrix}$$

and define

$$\mathbf{X} = (x_0, x_1, \ldots, x_M)^T, \quad \mathbf{V} = (v_0, v_1, \ldots, v_M)^T \in \mathbb{R}^{d(M+1)}$$

be vectors that contain approximations at all nodes[3]. With initial conditions

$$\mathbf{X}_0 = (x_0, x_0, \ldots, x_0)^T, \quad \mathbf{V}_0 = (v_0, v_0, \ldots, v_0)^T \in \mathbb{R}^{d(M+1)}$$

and $F(\mathbf{X}, \mathbf{V}) = (f_0, f_1, \ldots, f_M)^T \in \mathbb{R}^{d(M+1)}$ denoting the vector that contains the forces at each node. Then, the equation (2.6) can be written compactly as

$$\mathbf{X} = \mathbf{X}_0 + \Delta t \mathbf{Q} \mathbf{V}_0 + \Delta t^2 \mathbf{Q}\mathbf{Q} F(\mathbf{X}, \mathbf{V}), \tag{2.7a}$$

$$\mathbf{V} = \mathbf{V}_0 + \Delta t \mathbf{Q} F(\mathbf{X}, \mathbf{V}) \tag{2.7b}$$

where

$$Q := \begin{pmatrix} 0 & \mathbf{0} \\ \mathbf{0} & \bar{Q} \end{pmatrix} \in \mathbb{R}^{(M+1)\times(M+1)}$$

with $\mathbf{0}$ being the $M-$dimensional zero-vector, and

$$\mathbf{Q} = Q \otimes \mathbf{I}_d, \quad \mathbf{Q}\mathbf{Q} = (Q \otimes \mathbf{I}_d)(Q \otimes \mathbf{I}_d) = QQ \otimes \mathbf{I}_d \tag{2.8}$$

[3]We use boldface variables to indicate that have been aggregated over multiple quadrature nodes. However, note that non-boldface variables can be vectors, too. For example, $v_1 \in \mathbb{R}^d$ is the velocity at the first quadrature nodes whereas $\mathbf{V} \in \mathbb{R}^{d(M+1)}$ are the velocities at all quadrature nodes.

where $\mathbf{I}_d$ being the identity matrix of dimension d. Finally, let

$$\mathbf{U} = (\mathbf{X}, \mathbf{V}) = (x_0, \dots, x_M, v_0, \dots, v_M)^T \in \mathbb{R}^{2d(M+1)}.$$

Then, the equation (2.7) turns into

$$\mathbf{C}_{\text{coll}}\mathbf{U}_0 = \mathbf{U} - \Delta t \mathbf{Q}_{\text{coll}}\mathbf{F}(\mathbf{U}) =: \mathbf{M}_{\text{coll}}(\mathbf{U}) \tag{2.9}$$

where $\mathbf{U}_0 = (x_0, \dots, x_0, v_0, \dots, v_0)^T$, $\mathbf{F}(\mathbf{U}) = (f_0, \dots, f_M, f_0, \dots, f_M) \in \mathbb{R}^{2d(M+1)}$ and

$$\mathbf{Q}_{\text{coll}} = \begin{pmatrix} \Delta t\mathbf{Q}\mathbf{Q} & \mathbf{O} \\ \mathbf{O} & \mathbf{Q} \end{pmatrix}, \quad \mathbf{C}_{\text{coll}} = \begin{pmatrix} \mathbf{I}_{d(M+1)} & \Delta t\mathbf{Q} \\ \mathbf{O} & \mathbf{I}_{d(M+1)} \end{pmatrix} \in \mathbb{R}^{2d(M+1)\times 2d(M+1)}$$

with $\mathbf{O}$ denote $d(M+1) \times d(M+1)-$dimensional matrix with zero entries. Using (2.9) we can write the collocation problem in operator form

$$\mathbf{M}_{\text{coll}}(\mathbf{U}) = \mathbf{C}_{\text{coll}}\mathbf{U}_0. \tag{2.10}$$

Now, to find the value at the final time step t_{n+1}, we derive the update step formula for the collocation problem [11, pp. 283-284]. The approximation at the end of the time step for the collocation problem (2.5) can be computed via

$$x(t_{n+1}) \approx x_{n+1} = x_0 + \Delta t \sum_{m=1}^{M} q_m v_m, \tag{2.11a}$$

$$v(t_{n+1}) \approx v_{n+1} = v_0 + \Delta t \sum_{m=1}^{M} q_m f_m \tag{2.11b}$$

where

$$q_j = \int_0^1 l_j(s)ds, \quad j = 1, \dots, M.$$

Insert (2.11b) into (2.11a) to obtain

$$x_{n+1} = x_0 + \Delta t v_0 + \Delta t^2 \sum_{m=1}^{M}\sum_{i=1}^{M} q_i q_{i,m} f_m, \tag{2.12a}$$

$$v_{n+1} = v_0 + \Delta t \sum_{m=1}^{M} q_m f_m. \tag{2.12b}$$

Here we use the fact that $\sum_{m=1}^{M} q_m = 1$ by the consistency of the quadrature rule [11, pp. 208-215]. Once the stages x_m, v_m for $m = 1, \dots, M$ in (2.6) are known, the approximations at the end of the time step t_{n+1} can be computed using (2.12). Moreover, equations (2.12) can be written in vector form by using the above notations

$$x_{n+1} = x_0 + \Delta t \mathbf{q}\mathbf{V}_0 + \Delta t^2 \mathbf{q}\mathbf{Q}F(\mathbf{X}, \mathbf{V}), \tag{2.13a}$$

$$v_{n+1} = v_0 + \Delta t \mathbf{q}F(\mathbf{X}, \mathbf{V}) \tag{2.13b}$$

with $q := (0, q_1, \dots, q_M) \in \mathbb{R}^{1\times(M+1)}$ and $\mathbf{q} := q \otimes \mathbf{I}_d \in \mathbb{R}^{d\times d(M+1)}$. This is the collocation problem for second-order IVPs. Now we aim to find an iterative method that converges to the collocation problem solution. Before doing that, we introduce a preconditioner to improve the convergence rate, making it more efficient in finding the solution of collocation problem.

2.1.2 Velocity-Verlet scheme

We use velocity-Verlet integration [44] as a preconditioner for an iterative method to solve the collocation problem. Applying velocity-Verlet to (2.1) over the quadrature nodes $\tau_0, \ldots, \tau_M$ gives

$$x_{m+1} = x_m + \Delta\tau_{m+1}\left(v_m + \frac{\Delta\tau_{m+1}}{2} f_m\right), \tag{2.14a}$$

$$v_{m+1} = v_m + \frac{\Delta\tau_{m+1}}{2}(f_m + f_{m+1}) \tag{2.14b}$$

where $\Delta\tau_{m+1} = \tau_{m+1} - \tau_m,\ m = 0, \ldots, M-1$. To obtain a matrix formulation of (2.14), we use recursive insertion

$$x_{m+1} = x_0 + \sum_{l=1}^{m+1} \Delta\tau_l v_{l-1} + \frac{1}{2}\sum_{l=1}^{m+1} (\Delta\tau_l)^2 f_{l-1}, \tag{2.15a}$$

$$v_{m+1} = v_0 + \frac{1}{2}\sum_{l=1}^{m+1} \Delta\tau_l (f_{l-1} + f_l). \tag{2.15b}$$

These equations can be rearranged into vector form by defining

$$Q_{\mathrm{E}} := \frac{1}{\Delta t}\begin{pmatrix} 0 & 0 & 0 & \ldots & 0 \\ \Delta\tau_1 & 0 & 0 & \ldots & 0 \\ \Delta\tau_1 & \Delta\tau_2 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \Delta\tau_1 & \Delta\tau_2 & \ldots & \Delta\tau_M & 0 \end{pmatrix}, \quad Q_{\mathrm{I}} := \frac{1}{\Delta t}\begin{pmatrix} 0 & 0 & 0 & \ldots & 0 \\ 0 & \Delta\tau_1 & 0 & \ldots & 0 \\ 0 & \Delta\tau_1 & \Delta\tau_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \Delta\tau_1 & \Delta\tau_2 & \ldots & \Delta\tau_M \end{pmatrix}$$

and

$$Q_{\mathrm{T}} := \frac{1}{2}(Q_{\mathrm{E}} + Q_{\mathrm{I}}) \in \mathbb{R}^{(M+1)\times(M+1)}. \tag{2.16}$$

Then, (2.15) becomes

$$\mathbf{X} = \mathbf{X}_0 + \Delta t \mathbf{Q}_{\mathrm{E}} \mathbf{V} + \frac{\Delta t^2}{2}(\mathbf{Q}_{\mathrm{E}} \circ \mathbf{Q}_{\mathrm{E}}) F(\mathbf{X}, \mathbf{V}), \tag{2.17a}$$

$$\mathbf{V} = \mathbf{V}_0 + \Delta t \mathbf{Q}_{\mathrm{T}} F(\mathbf{X}, \mathbf{V}) \tag{2.17b}$$

with $\circ$ denoting the Hadamard product (element-wise product of two matrices). We substitute the expression for $\mathbf{V}$ from (2.17b) into (2.17a) so that

$$\mathbf{X} = \mathbf{X}_0 + \Delta t \mathbf{Q}_{\mathrm{E}} \mathbf{V}_0 + \Delta t^2 \mathbf{Q}_{\mathrm{x}} F(\mathbf{X}, \mathbf{V}), \tag{2.18a}$$

$$\mathbf{V} = \mathbf{V}_0 + \Delta t \mathbf{Q}_{\mathrm{T}} F(\mathbf{X}, \mathbf{V}) \tag{2.18b}$$

where

$$\mathbf{Q}_{\mathrm{x}} := Q_{\mathrm{x}} \otimes \mathbf{I}_d \in \mathbb{R}^{d(M+1)\times d(M+1)} \quad \text{with} \quad Q_{\mathrm{x}} := Q_{\mathrm{E}} Q_{\mathrm{T}} + \frac{1}{2}(Q_{\mathrm{E}} \circ Q_{\mathrm{E}}). \tag{2.19}$$

To combine these equations into a compact form based on $\mathbf{U}$, we define

$$\mathbf{C}_{\mathrm{vv}} := \begin{pmatrix} \mathbf{I}_{d(M+1)} & \Delta t \mathbf{Q}_{\mathrm{E}} \\ \mathbf{O} & \mathbf{I}_{d(M+1)} \end{pmatrix}, \quad \mathbf{Q}_{\mathrm{vv}} := \begin{pmatrix} \Delta t \mathbf{Q}_{\mathrm{x}} & \mathbf{O} \\ \mathbf{O} & \mathbf{Q}_{\mathrm{T}} \end{pmatrix}.$$

Finally, the matrix representation of the velocity-Verlet scheme is

$$\mathbf{U} = \mathbf{C}_{\text{vv}}\mathbf{U}_0 + \Delta t\mathbf{Q}_{\text{vv}}\mathbf{F}(\mathbf{U})$$

or

$$\mathbf{M}_{\text{vv}}(\mathbf{U}) := \mathbf{U} - \Delta t\mathbf{Q}_{\text{vv}}\mathbf{F}(\mathbf{U}) = \mathbf{C}_{\text{vv}}\mathbf{U}_0. \tag{2.20}$$

We will use the operator $\mathbf{M}_{\text{vv}}(\cdot)$ as a preconditioner for the Picard iteration to derive the SDC method that converges to the solution of equation (2.10).

2.2 Spectral deferred correction method

Applying a Richardson iteration [43, Chapter 1] to (2.10) gives the following Picard iteration

$$\mathbf{U}^{k+1} = (\mathbf{I}_{2d(M+1)} - \mathbf{M}_{\text{coll}})(\mathbf{U}^k) + \mathbf{C}_{\text{coll}}\mathbf{U}_0 = \mathbf{C}_{\text{coll}}\mathbf{U}_0 + \Delta t\mathbf{Q}_{\text{coll}}\mathbf{F}(\mathbf{U}^k) \tag{2.21}$$

where $k = 0, \ldots, K$ is the iteration index and we use $\mathbf{U}^0$ to start the iteration.

Proposition 2.2. *Let f be a Lipschitz continuous function with Lipschitz constant L and Δt sufficiently small so that $\Delta tL\|\mathbf{Q}_{\text{coll}}\| < 1$. Then,* (2.21) *converges to the collocation problem solution for all starting values $\mathbf{U}_0$.*

Proof. Subtracting (2.21) for $k+1$ and k yields

$$\mathbf{U}^{k+1} - \mathbf{U}^k = \Delta t\mathbf{Q}_{\text{coll}}(\mathbf{F}(\mathbf{U}^k) - \mathbf{F}(\mathbf{U}^{k-1})).$$

Applying a norm and using Lipschitz continuity gives us

$$\|\mathbf{U}^{k+1} - \mathbf{U}^k\| \leq \Delta tL\|\mathbf{Q}_{\text{coll}}\|\|\mathbf{U}^k - \mathbf{U}^{k-1}\|.$$

Since $\Delta tL\|\mathbf{Q}_{\text{coll}}\| < 1$, the iteration converges [45, pp. 1-10]. □

Often, the Picard iteration converges only for impractically small time steps [46]. To improve convergence, we use $\mathbf{M}_{\text{vv}}(\cdot)$ in (2.20) as a preconditioner [43, Chapter 1], leading to

$$\mathbf{M}_{\text{vv}}(\mathbf{U}^{k+1}) = (\mathbf{M}_{\text{vv}} - \mathbf{M}_{\text{coll}})(\mathbf{U}^k) + \mathbf{C}_{\text{coll}}\mathbf{U}_0. \tag{2.22}$$

Each iteration requires solving a linear or nonlinear system of equations, depending on the right-hand side function f in (2.1). However, the structure of $\mathbf{M}_{\text{vv}}$ allows this to be done by sweeping through the quadrature nodes sequentially. Using (2.20) and (2.9), we obtain the matrix form of the SDC iteration for the second-order problems.

$$(\mathbf{I}_{2d(M+1)} - \Delta t\mathbf{Q}_{\text{vv}}\mathbf{F})(\mathbf{U}^{k+1}) = \Delta t(\mathbf{Q}_{\text{coll}} - \mathbf{Q}_{\text{vv}})\mathbf{F}(\mathbf{U}^k) + \mathbf{C}_{\text{coll}}\mathbf{U}_0 \tag{2.23}$$

for $k = 0, \ldots, K$.

Remark 2.3. Typically, the starting value $\mathbf{U}^0$ for iteration (2.23) is generated from the initial value $\mathbf{U}_0$ at the beginning of the time step, either by setting $\mathbf{U}^0 = \mathbf{U}_0$ or by performing an initial sweep of the velocity-Verlet base method to solve

$$\mathbf{M}_{\text{vv}}(\mathbf{U}^0) = \mathbf{C}_{\text{vv}}\mathbf{U}_0.$$

For analysis in Chapter 3, it will be useful to separate the equation (2.23) into its components

$$\mathbf{X}^{k+1} - \Delta t^2 \mathbf{Q}_{\mathrm{x}} F(\mathbf{X}^{k+1}, \mathbf{V}^{k+1}) = \mathbf{X}_0 + \Delta t \mathbf{Q}\mathbf{V}_0 + \Delta t^2 (\mathbf{Q}\mathbf{Q} - \mathbf{Q}_{\mathrm{x}}) F(\mathbf{X}^k, \mathbf{V}^k), \quad (2.24a)$$

$$\mathbf{V}^{k+1} - \Delta t \mathbf{Q}_{\mathrm{T}} F(\mathbf{X}^{k+1}, \mathbf{V}^{k+1}) = \mathbf{V}_0 + \Delta t (\mathbf{Q} - \mathbf{Q}_{\mathrm{T}}) F(\mathbf{X}^k, \mathbf{V}^k). \quad (2.24b)$$

Or

$$\mathbf{X}^{k+1} = \mathbf{X}_0 + \Delta t \mathbf{Q}\mathbf{V}_0 + \Delta t^2 \mathbf{Q}_{\mathrm{x}} (F(\mathbf{X}^{k+1}, \mathbf{V}^{k+1}) - F(\mathbf{X}^k, \mathbf{V}^k)) + \Delta t^2 \mathbf{Q}\mathbf{Q} F(\mathbf{X}^k, \mathbf{V}^k), \quad (2.25a)$$

$$\mathbf{V}^{k+1} = \mathbf{V}_0 + \Delta t \mathbf{Q}_{\mathrm{T}} (F(\mathbf{X}^{k+1}, \mathbf{V}^{k+1}) - F(\mathbf{X}^k, \mathbf{V}^k)) + \Delta t \mathbf{Q} F(\mathbf{X}^k, \mathbf{V}^k). \quad (2.25b)$$

Algorithm 1: Spectral Deferred Correction (SDC)

Data: Initial conditions $x_0,\ v_0$, number of iterations K, and number of quadrature nodes M.
Result: Solution after Kth SDC iterations $x_m^K,\ v_m^K$ for $m = 0, \ldots, M$.

```
# Set initial guess spread
```
for $m = 0, \ldots, M$ **do**
  $x_m^0 \leftarrow x_0$
  $v_m^0 \leftarrow v_0$
end
for $k = 0, \ldots, K-1$ **do**
  ```
  # Set initial condition
  ```
  $x_0^{k+1} \leftarrow x_0, \quad v_0^{k+1} \leftarrow v_0$
  ```
  # Compute function evaluations
  ```
  for $m = 0, \ldots, M$ **do**
    $f_m^k \leftarrow \mathrm{eval}(f(x_m^k, v_m^k))$
  end
  for $m = 0, \ldots, M-1$ **do**
    ```
    # Compute new position values
    ```
    $x_{m+1}^{k+1} \leftarrow x_m^{k+1} + \Delta\tau_{m+1} v_0 + \Delta t^2 \sum_{l=0}^{m} s_{m+1,l}^{\mathbf{x}} (f_l^{k+1} - f_l^k) + \Delta t^2 \sum_{l=0}^{M} s q_{m+1,l} f_l^k$
    ```
    # Evaluate intermediate quantity c^k
    ```
    $\mathbf{c}^k = -\frac{1}{2}\left(f_{m+1}^k + f_m^k\right) + \frac{1}{\Delta\tau_{m+1}} \sum_{l=0}^{M} s_{m+1,l} f_l^k$
    ```
    # Update the velocity using Newton solvers[47, Chapter 2]:
    ```
    $v_{m+1}^{k+1} \leftarrow \mathrm{NewtonSolver}\left(\mathbf{c}^k, f_{m+1}^{k+1}, f_m^{k+1}, v_m^{k+1}\right)$
  end
end

We can convert the (2.25) into the node-to-node formulation by using definitions (2.8), (2.16) and (2.19)

$$x_{m+1}^{k+1} = x_0 + \Delta t \sum_{l=0}^{M} q_{m+1,l} v_0 + \Delta t^2 \sum_{l=0}^{M} q_{m+1,l}^{\mathrm{x}} (f_l^{k+1} - f_l^k) + \Delta t^2 \sum_{l=0}^{M} q q_{m+1,l} f_l^k \quad (2.26a)$$

$$v_{m+1}^{k+1} = v_0 + \Delta t \sum_{l=0}^{M} q_{m+1,l}^{\mathrm{T}}(f_l^{k+1} - f_l^k) + \Delta t \sum_{l=0}^{M} q_{m+1,l} f_l^k \tag{2.26b}$$

where $m = 0, \ldots, M-1$ and $k = 0, \ldots, K$, $f_l^k := f(x_l^k, v_l^k)$ and $(q_{m,l}^{\mathrm{x}})_{m,l=0,\ldots,M}$ and $(q_{m,l}^{\mathrm{T}})_{m,l=0,\ldots,M}$ are the entries of matrices Q_{x} and Q_{T} respectively. By taking difference (2.26) for $m+1$ and m and since Q_{T} is lower diagonal and Q_{x} strictly lower diagonal matrices, we obtain

$$x_{m+1}^{k+1} = x_m^{k+1} + \Delta\tau_{m+1} v_0 + \Delta t^2 \sum_{l=0}^{m} s_{m+1,l}^{\mathrm{x}}(f_l^{k+1} - f_l^k) + \Delta t^2 \sum_{l=0}^{M} sq_{m+1,l} f_l^k, \tag{2.27a}$$

$$v_{m+1}^{k+1} = v_m^{k+1} + \frac{\Delta\tau_{m+1}}{2}(f_{m+1}^{k+1} - f_{m+1}^k) + \frac{\Delta\tau_{m+1}}{2}(f_m^{k+1} - f_m^k) + \Delta t \sum_{l=0}^{M} s_{m+1,l} f_l^k \tag{2.27b}$$

for $m = 0, \ldots, M-1$ and $k = 0, \ldots, K$. Here,

$$s_{m,j} := q_{m,j} - q_{m-1,j},\ s_{m,j}^{\mathrm{x}} := q_{m,l}^{\mathrm{x}} - q_{m-1,l}^{\mathrm{x}},\ sq_{m,l} := qq_{m,l} - qq_{m-1,l}$$

where $m, j = 1, \ldots, M$. Since $\Delta t \sum_{j=0}^{M} s_{m,j} = \Delta\tau_m$ [11, pp. 208-210], which explains the factor in front of v_0.

Remark 2.4. If f does not depend on v, (2.27) is a fully explicit SDC iteration.

Algorithm 1 illustrates a step-by-step outline of SDC method for second-order problems. The velocity update in equation (2.27b) typically requires a nonlinear solver if f depends on velocity v. One can use a nonlinear solvers [47, Chapter 2] like Newton's method to obtain the updated velocity values.

2.3 Boris-SDC

In this section, we introduce the Boris-SDC method, as published in [3], for integrating the equations of motion of electrically charged particles in magnetic fields. The Boris-SDC method combines the Boris integrator with the SDC method and is specifically designed for the Lorentz equation that describe the movement of electrically charged particles in electric and magnetic fields. The right-hand side of equation (2.1) is represented by the Lorentz force [48], leading to the following equations of motion

$$\frac{dx}{dt} = v, \tag{2.28a}$$

$$\frac{dv}{dt} = f(x, v) = \alpha[E(x,t) + v \times B(x,t)] \tag{2.28b}$$

with the magnetic field $B(x,t) \in \mathbb{R}^d$, electric field $E(x,t) \in \mathbb{R}^d$ and the charge-to-mass ratio $\alpha = q/m$. The Boris integration method [29, 49] provides a clever way to evaluate (2.28b) without having to solve an implicit system. It allows for efficient integration of the particle trajectory in the presence of electric and magnetic fields.

Boris-SDC The right-hand side of f as stated in (2.28b)

$$f(x,v) = \alpha[E(x,t) + v \times B(x,t)] \tag{2.29}$$

with a constant magnetic field $B(x,t) = B$ for simplicity (note that B-field is non-constant at the end of this section). First, consider the implicit part of the velocity-Verlet scheme (2.14b) can be rewritten as

$$\frac{v_{m+1} - v_m}{\Delta\tau} = \alpha \left[E_{m+\frac{1}{2}} + \frac{v_{m+1} + v_m}{2} \times B \right] \tag{2.30}$$

where $E_{m+\frac{1}{2}} := \frac{1}{2}\left(E(x_m, \tau_m) + E(x_{m+1}, \tau_{m+1})\right)$ is the average electric field at times τ_m and τ_{m+1}. The idea behind the Boris integrator is to separate the electric and magnetic forces such that

$$v^- := v_m + \frac{\alpha\Delta\tau}{2} E_{m+\frac{1}{2}} \quad \text{and} \quad v^+ := v_{m+1} - \frac{\alpha\Delta\tau}{2} E_{m+\frac{1}{2}}, \tag{2.31}$$

so that

$$\frac{v^+ - v^-}{\Delta\tau} = \frac{\alpha}{2}(v^+ + v^-) \times B. \tag{2.32}$$

which can be shown to correspond to a simple rotation, i.e. $|v^+| = |v^-|$ and can be solved for v^+ explicitly [49, Chapter 4] using $v^+ = v^- + (v^- + v^- \times \gamma) \times \beta$ with $\gamma = \alpha B \cdot \frac{\Delta\tau}{2}$ and $\beta = \frac{2\gamma}{(1+|\gamma|^2)}$. Therefore, using (2.31), the new velocity v_{m+1} in the equation (2.14) can be computed explicitly [49, Chapter 15].

We can use this idea to provide a very quick and easy way to solve the implicit dependency of the velocity in the SDC iteration (2.27b). Particularly, we can express the update for the $(m+1)$th component of velocity as

$$\frac{v_{m+1}^{k+1} - v_m^{k+1}}{\Delta\tau_{m+1}} = \frac{1}{2}\left(f_{m+1}^{k+1} + f_k^{k+1}\right) - \frac{1}{2}\left(f_{m+1}^k + f_m^k\right) + \frac{1}{\Delta\tau_{m+1}} \sum_{l=0}^{M} s_{m+1,l} f_l^k. \tag{2.33}$$

Define

$$\Delta\tau_{m+1}\mathbf{c}^k := -\frac{\Delta\tau_{m+1}}{2}\left(f_{m+1}^k + f_m^k\right) + \sum_{l=0}^{M} s_{m+1,l} f_l^k. \tag{2.34}$$

Note that $\mathbf{c}^k$ depends only on values at iteration k. Thus, this is known from the previous iteration. Then, (2.33) turns into

$$\frac{v_{m+1}^{k+1} - v_m^{k+1}}{\Delta\tau_{m+1}} = \frac{f_{m+1}^{k+1} + f_m^{k+1}}{2} + \mathbf{c}^k. \tag{2.35}$$

With the particular right-hand side (2.29), we can write

$$\frac{v_{m+1}^{k+1} - v_m^{k+1}}{\Delta\tau_{m+1}} = \alpha \left[E_{m+\frac{1}{2}}^{k+1} + \frac{v_{m+1}^{k+1} + v_m^{k+1}}{2} \times B \right] + \mathbf{c}^k. \tag{2.36}$$

We define

$$v^- := v_m^{k+1} + \frac{\Delta\tau_{m+1}}{2}\left(\alpha E_{m+\frac{1}{2}}^{k+1} + \mathbf{c}^k\right) \quad \text{and} \quad v^+ := v_{m+1}^{k+1} - \frac{\Delta\tau_{m+1}}{2}\left(\alpha E_{m+\frac{1}{2}}^{k+1} + \mathbf{c}^k\right) \tag{2.37}$$

which is precisely of the type of (2.32). As mentioned before, this can be solved explicitly for v^+, so that (2.37) can be used to determine v_{m+1}^{k+1}.

Now, consider non-constant magnetic field case, instead of (2.30), we obtain

$$\begin{aligned}\frac{v_{m+1}-v_m}{\Delta\tau_{m+1}} &= \alpha\left[E_{m+\frac{1}{2}}+\frac{1}{2}v_{m+1}\times B_{m+1}+\frac{1}{2}v_m\times B_m\right]\\ &= \alpha\left[E_{m+\frac{1}{2}}+\frac{v_{m+1}+v_m}{2}\times B_{m+1}\right]+\frac{\alpha}{2}v_m\times(B_m-B_{m+1}).\end{aligned}\tag{2.38}$$

Define

$$v^- := v_m+\frac{\alpha\Delta\tau_{m+1}}{2}\left(E_{m+\frac{1}{2}}+\frac{1}{2}v_m\times(B_m-B_{m+1})\right),\tag{2.39}$$

$$v^+ := v_{m+1}-\frac{\alpha\Delta\tau_{m+1}}{2}\left(E_{m+\frac{1}{2}}+\frac{1}{2}v_m\times(B_m-B_{m+1})\right).\tag{2.40}$$

This yields

$$\frac{v^+-v^-}{\Delta\tau_{m+1}}=\frac{\alpha}{2}(v^++v^-)\times B_{m+1}.\tag{2.41}$$

This equation is similar to (2.32). Therefore, we can solve it explicitly for v^+. We apply the same structure as constant magnetic field. For the non-constant magnetic field the equation (2.35) turns into

$$\frac{v_{m+1}^{k+1}-v_m^{k+1}}{\Delta\tau_{m+1}}=\alpha\left[E_{m+\frac{1}{2}}^{k+1}+\frac{v_{m+1}^{k+1}}{2}\times B_{m+1}^{k+1}+\frac{v_m^{k+1}}{2}\times B_m^{k+1}\right]+\mathbf{c}^k.\tag{2.42}$$

We can rewrite this in the following form

$$\frac{v_{m+1}^{k+1}-v_m^{k+1}}{\Delta\tau_{m+1}}=\alpha\left[E_{m+\frac{1}{2}}^{k+1}+\frac{v_{m+1}^{k+1}+v_m^{k+1}}{2}\times B_{m+1}^{k+1}\right]+\frac{\alpha}{2}v_m^{k+1}\times(B_m^{k+1}-B_{m+1}^{k+1})+\mathbf{c}^k.\tag{2.43}$$

We define

$$\tilde{\mathbf{c}}^k := \frac{\alpha}{2}v_m^{k+1}\times(B_m^{k+1}-B_{m+1}^{k+1})+\mathbf{c}^k\tag{2.44}$$

so that

$$\frac{v_{m+1}^{k+1}-v_m^{k+1}}{\Delta\tau_{m+1}}=\alpha\left[E_{m+\frac{1}{2}}^{k+1}+\frac{v_{m+1}^{k+1}+v_m^{k+1}}{2}\times B_{m+1}^{k+1}\right]+\tilde{\mathbf{c}}^k.\tag{2.45}$$

Define

$$v^- := v_m^{k+1}+\frac{\Delta\tau_{m+1}}{2}\left(\alpha E_{m+\frac{1}{2}}^{k+1}+\tilde{\mathbf{c}}^k\right)\quad\text{and}\quad v^+ := v_{m+1}^{k+1}-\frac{\Delta\tau_{m+1}}{2}\left(\alpha E_{m+\frac{1}{2}}^{k+1}+\tilde{\mathbf{c}}^k\right).$$

As a result, the equation (2.45) becomes

$$\frac{v^+-v^-}{\Delta\tau_{m+1}}=\frac{\alpha}{2}(v^++v^-)\times B_{m+1}^{k+1},\tag{2.46}$$

This equation can be solved explicitly for v^+ using the Boris integration method (2.32). Henceforth, we employ the Boris-SDC method to solve the Lorentz equation for given parameters. Algorithm 2 outlines the Boris solver, which can be used to handle the nonlinear component of velocity in Algorithm 1.

Algorithm 2: Boris solver [3]

Data: $\mathbf{c}^k, f_m^{k+1}, f_{m+1}^{k+1}, v_m^{k+1}$
Result: new velocity value v_{m+1}^{k+1}
`# Separate the parameters for electric and magnetic fields`
$E_m^{k+1},\ B_m^{k+1} \leftarrow \text{Split}(f_m^{k+1})$
$E_{m+1}^{k+1},\ B_{m+1}^{k+1} \leftarrow \text{Split}(f_{m+1}^{k+1})$
`# Compute the average of electric field`
$E_{m+\frac{1}{2}} \leftarrow 0.5 \cdot (E_{m+1}^{k+1} + E_m^{k+1})$
$\mathbf{c}^k \leftarrow \mathbf{c}^k + \frac{\Delta\tau_{m+1}}{2}\alpha(v_m^{k+1} \times (B_m^{k+1} - B_{m+1}^{k+1}))$
`# pre-velocity, separated by the electric force and the` $\mathbf{c}^k$ `term`
$v^- \leftarrow v_m^{k+1} + \frac{\Delta\tau_{m+1}}{2}\alpha E_{m+\frac{1}{2}} + \frac{\mathbf{c}^k}{2}$
`# Rotation`
$t \leftarrow \frac{\Delta\tau_{m+1}}{2}\alpha B_{m+1}^{k+1}$
$s \leftarrow \frac{2t}{1+\|t\|^2}$
$v^+ \leftarrow v^- + (v^- + v^- \times t) \times s$
`# post velocity`
$v_{m+1}^{k+1} \leftarrow v^+ + \frac{\Delta\tau_{m+1}}{2}\alpha E_{m+\frac{1}{2}} + \frac{\mathbf{c}^k}{2}$

Chapter 3

Analysis of SDC for second-order problems

In this chapter, we focus on the stability and convergence analysis of the SDC method for second-order problems, providing a more detailed explanation of the results published in [1]. This work builds on previous studies [50, 28] that analyzed the convergence of SDC method for first-order problems. By adopting the framework for second-order ODEs presented in [28], we investigate the stability and convergence properties of SDC method for second-order problems [1].

3.1 Stability

To study stability, we need to select a specific test problem. For first-order problems, the Dahlquist equation is used as a test problem to analyze stability. For second-order problems, we use the damped harmonic oscillator with unit mass

$$\dot{x}(t) = v(t), \tag{3.1a}$$
$$\dot{v}(t) = f(x(t), v(t)) := -\kappa x(t) - \mu v(t) \tag{3.1b}$$

as a test problem where κ is the spring constant and μ is the friction coefficient. Since our test problem is linear (e.g $\mathbf{F}(\mathbf{U}) = \mathbf{F}\mathbf{U}$) and assuming that $\mathbf{M}_{\text{vv}} = \mathbf{I}_{2(M+1)} - \mathbf{Q}_{\text{vv}}\mathbf{F}$ is non-singular, the iteration in (2.23) becomes

$$\mathbf{U}^{k+1} = \mathbf{K}_{\text{sdc}}\mathbf{U}^k + (\mathbf{I}_{2(M+1)} - \Delta t\mathbf{Q}_{\text{vv}}\mathbf{F})^{-1}\mathbf{C}_{\text{coll}}\mathbf{U}_0 \tag{3.2}$$

where

$$\mathbf{K}_{\text{sdc}} := (\mathbf{I}_{2(M+1)} - \Delta t\mathbf{Q}_{\text{vv}}\mathbf{F})^{-1}(\Delta t\mathbf{Q}_{\text{coll}} - \Delta t\mathbf{Q}_{\text{vv}})\mathbf{F} \tag{3.3}$$

is the iteration matrix, depending only on $\Delta t\kappa$ and $\Delta t\mu$. The iteration matrix is similar to the one for first-order problems [19].

Proposition 3.1. *The sequence* $\{\mathbf{U}^k\}$ *generated by* (3.2) *converges for any* $\mathbf{U}^0$ *and starting values* $\mathbf{U}_0$ *if and only if*

$$\rho(\mathbf{K}_{\text{sdc}}) := \max_{\lambda \in \sigma(\mathbf{K}_{\text{sdc}})} |\lambda| < 1. \tag{3.4}$$

where $\sigma(\mathbf{K}_{\text{sdc}})$ *is referred to as the spectrum of* $\mathbf{K}_{\text{sdc}}$.

Proof. First, assume that $\rho := \rho(\mathbf{K}_{\text{sdc}}) < 1$. Taking the difference of (3.2) for iterations $k+1$ and k yields

$$\mathbf{U}^{k+1} - \mathbf{U}^k = \mathbf{K}_{\text{sdc}}(\mathbf{U}^k - \mathbf{U}^{k-1}),\ k \geq 1. \tag{3.5}$$

By induction arguments, this yields

$$\mathbf{U}^{k+1} - \mathbf{U}^k = \mathbf{K}_{\text{sdc}}^k(\mathbf{U}^1 - \mathbf{U}^0),\ k \geq 1. \tag{3.6}$$

We can obtain from [51, Theorem 7] that

$$\|\mathbf{K}_{\text{sdc}}^k\| \leq C_{\mathbf{K}_{\text{sdc}}} \rho^k k^{2d(M+1)-1}$$

where $C_{\mathbf{K}_{\text{sdc}}}$ is a constant depending only on $\mathbf{K}_{\text{sdc}}$, assuming that $k \geq 2d(M+1)$. Since $0 \leq \rho < 1$ the series $\sum_{k=1}^{\infty} \rho^k k^{2d(M+1)-1}$ is a convergent series. Therefore,

$$\sum_{k=1}^{\infty} \|\mathbf{U}^k - \mathbf{U}^{k-1}\| \leq \sum_{k=1}^{2d(M+1)-1} \|\mathbf{U}^k - \mathbf{U}^{k-1}\| + C_{\mathbf{K}_{\text{sdc}}} \|\mathbf{U}^1 - \mathbf{U}^0\| \sum_{k=2d(M+1)}^{\infty} \rho^k k^{2d(M+1)-1} < +\infty$$

Thus, $\{\mathbf{U}^k\}$ is the Cauchy sequence. Hence, $\{\mathbf{U}^k\}$ is a convergent sequence.

Now suppose $\{\mathbf{U}^k\}$ is a convergent sequence for any $\mathbf{U}^0$ and $\mathbf{U}_0$. Then we show that

$$\rho(\mathbf{K}_{\text{sdc}}) < 1.$$

Note that the matrix product $(\mathbf{I}_{2(M+1)} - \Delta t \mathbf{Q}_{\text{vv}} \mathbf{F})^{-1} \mathbf{C}_{\text{coll}}$ is invertible since so are $(\mathbf{I}_{2(M+1)} - \Delta t \mathbf{Q}_{\text{vv}} \mathbf{F})^{-1}$ as well as $\mathbf{C}_{\text{coll}}$.

Assume that $\rho(\mathbf{K}_{\text{sdc}}) \geq 1$. Suppose, there exists $\lambda \in \sigma(\mathbf{K}_{\text{sdc}})$ such that $|\lambda| \geq 1$ and $\lambda \in \mathbb{R}$. Then there exists $\mathbf{e} \in \mathbb{R}^{2d(M+1)} \setminus \{0\}$ such that $\mathbf{K}_{\text{sdc}} \mathbf{e} = \lambda \mathbf{e}$. We choose $\mathbf{U}_0$ such that

$$(\mathbf{I}_{2(M+1)} - \Delta t \mathbf{Q}_{\text{vv}} \mathbf{F})^{-1} \mathbf{C}_{\text{coll}} \mathbf{U}_0 = \mathbf{e}. \tag{3.7}$$

Such $\mathbf{U}_0$ exists, actually

$$\mathbf{U}_0 = \mathbf{C}_{\text{coll}}^{-1} (\mathbf{I}_{2(M+1)} - \Delta t \mathbf{Q}_{\text{vv}} \mathbf{F}) \mathbf{e}.$$

Take $\mathbf{U}^0 = 0$, then we have

$$\mathbf{U}^1 - \mathbf{U}^0 = \mathbf{U}^1 = \mathbf{e}$$

and

$$\mathbf{U}^k - \mathbf{U}^{k-1} = \mathbf{K}_{\text{sdc}}^{k-1} (\mathbf{U}^1 - \mathbf{U}^0) = \lambda^{k-1} \mathbf{e}.$$

Therefore, $\|\mathbf{U}^k - \mathbf{U}^{k-1}\| \geq \|\lambda^{k-1} \mathbf{e}\| = |\lambda|^{k-1} \|\mathbf{e}\| \geq \|\mathbf{e}\| > 0$. Thus, $\{\mathbf{U}^k\}$ is not the Cauchy sequence and does not converge.

Suppose $|\lambda| \geq 1$ and $\lambda \in \mathbb{C} \setminus \mathbb{R}$. Then we can write $\lambda = |\lambda| \exp(i\pi\theta),\ \theta \in (0,2) \setminus \{1\}$. Then, there exists $\mathbf{e} \in \mathbb{C}^{2d(M+1)} \setminus \{0\}$ such that

$$\mathbf{K}_{\text{sdc}}(\mathbf{e}) = \lambda \mathbf{e}.$$

Again there exists a vector $\mathbf{U}_0'$ such that

$$(\mathbf{I}_{2(M+1)} - \Delta t \mathbf{Q}_{\text{vv}} \mathbf{F})^{-1} \mathbf{C}_{\text{coll}} \mathbf{U}_0' = \mathbf{e}.$$

Take $\mathbf{U}_0 = \mathbf{U}_0' + \bar{\mathbf{U}}_0'$, where $\bar{\mathbf{U}}_0'$ denotes the complex conjugate of the vector $\mathbf{U}_0'$. Then we have $\mathbf{K}_{\text{sdc}}(\bar{\mathbf{e}}) = \bar{\lambda}\bar{\mathbf{e}}$, $(\mathbf{I}_{2(M+1)} - \Delta t\mathbf{Q}_{\text{vv}}\mathbf{F})^{-1}\mathbf{C}_{\text{coll}}\bar{\mathbf{U}}_0' = \bar{\mathbf{e}}$ because $\mathbf{K}_{\text{sdc}}$, $(\mathbf{I}_{2(M+1)} - \Delta t\mathbf{Q}_{\text{vv}}\mathbf{F})^{-1}\mathbf{C}_{\text{coll}}$ are real matrices. Assume $\mathbf{U}^0 = 0$. Then we have $\mathbf{U}^1 - \mathbf{U}^0 = \bar{\mathbf{e}} + \mathbf{e}$. Thus, we have

$$\mathbf{U}^{k+1} - \mathbf{U}^k = \mathbf{K}_{\text{sdc}}^k(\mathbf{U}^1 - \mathbf{U}^0) = \mathbf{K}_{\text{sdc}}^k(\bar{\mathbf{e}} + \mathbf{e}) = \bar{\lambda}^k\bar{\mathbf{e}} + \lambda^k\mathbf{e}. \tag{3.8}$$

We assume that

$$\mathbf{e} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_{2d(M+1)} \end{pmatrix}$$

with $c_1 \in \mathbb{R} \setminus \{0\}$. Without loss of generality, we may also assume that c_1 is a real number provided $c_1 \neq 0$, otherwise if $c_1 \in \mathbb{C} \setminus \mathbb{R}$ we can replace $\mathbf{e}$ with $\frac{1}{c_1}\mathbf{e}$. Then we have

$$\mathbf{U}^{k+1} - \mathbf{U}^k = \begin{pmatrix} c_1(\lambda^k + \bar{\lambda}^k) \\ \lambda^k c_2 + \bar{\lambda}^k\bar{c}_2 \\ \vdots \\ \lambda^k c_{2d(M+1)} + \bar{\lambda}^k\bar{c}_{2d(M+1)} \end{pmatrix},$$

and

$$c_1(\lambda^k + \bar{\lambda}^k) = c_1|\lambda|^k(\exp(\pi i\theta k) + \exp(-\pi i\theta k)) = 2c_1|\lambda|^k\cos(\pi k\theta).$$

Since $|\lambda| \geq 1$, the sequence $|\lambda|^k\cos(\pi\theta k)$ does not converge to zero for any $\theta \in (0,2) \setminus \{1\}$. Therefore $\|\mathbf{U}^{k+1} - \mathbf{U}^k\|$ does not converge to zero.

Consequently again $\{\mathbf{U}^k\}$ is not the Cauchy sequence and hence it does not converge. So we can conclude that $\rho(\mathbf{K}_{\text{sdc}}) < 1$. Proposition is proved. □

Remark 3.2. If we identify a pair of positive parameters $\Delta t(\kappa, \mu)$ with a point in the positive quadrant $\mathbb{R}_+^2$, we can define the *convergence domain* of SDC as

$$\Omega_{\text{conv}} := \{(\Delta t\kappa, \Delta t\mu) \in \mathbb{R}_+^2 : \rho(\mathbf{K}_{\text{sdc}}) < 1\}. \tag{3.9}$$

For a set of parameters within Ω_{conv}, SDC will converge to the solution of the collocation problem (2.6) as $k \to \infty$.

To assess stability as $n \to \infty$, we derive the stability function of SDC. We first introduce the following proposition to obtain the stability function.

Proposition 3.3. *If the right-hand side function f in the equation* (2.1) *is linear and $\mathbf{U}^0 = \mathbf{U}_0$, then SDC iteration can be written as*

$$\mathbf{U}^{k+1} = \mathbf{P}_{\text{sdc}}^{k+1}\mathbf{U}_0 \tag{3.10}$$

where

$$\mathbf{P}_{sdc}^k := \mathbf{K}_{\text{sdc}}^k + (\mathbf{I}_{2(M+1)} - \mathbf{K}_{\text{sdc}}^{k+1})(\mathbf{I}_{2(M+1)} - \mathbf{K}_{\text{sdc}})^{-1}\mathbf{M}_{\text{vv}}^{-1}\mathbf{C}_{\text{coll}}. \tag{3.11}$$

Proof. Using recursive insertion to (3.2) and taking $\mathbf{U}^0 = \mathbf{U}_0$, we obtain that

$$\mathbf{U}^{k+1} = \mathbf{K}_{\text{sdc}}^{k+1}\mathbf{U}_0 + \sum_{j=0}^{k}\mathbf{K}_{\text{sdc}}^j\mathbf{M}_{\text{vv}}^{-1}\mathbf{C}_{\text{coll}}\mathbf{U}_0. \tag{3.12}$$

In the summation term, we have a finite sum with the argument $\mathbf{K}_{\text{sdc}}$. Thus, using the following geometric series formula

$$\sum_{j=0}^{k} \mathbf{K}_{\text{sdc}}^{j} = (\mathbf{I}_{2(M+1)} - \mathbf{K}_{\text{sdc}}^{k+1})(\mathbf{I}_{2(M+1)} - \mathbf{K}_{\text{sdc}})^{-1},$$

we obtain

$$\mathbf{U}^{k+1} = \mathbf{K}_{\text{sdc}}^{k+1}\mathbf{U}_0 + (\mathbf{I}_{2(M+1)} - \mathbf{K}_{\text{sdc}}^{k+1})(\mathbf{I}_{2(M+1)} - \mathbf{K}_{\text{sdc}})^{-1}\mathbf{M}_{\text{vv}}^{-1}\mathbf{C}_{\text{coll}}\mathbf{U}_0. \tag{3.13}$$

Define

$$\mathbf{P}_{\text{sdc}}^{k} := \mathbf{K}_{\text{sdc}}^{k} + (\mathbf{I}_{2(M+1)} - \mathbf{K}_{\text{sdc}}^{k+1})(\mathbf{I}_{2(M+1)} - \mathbf{K}_{\text{sdc}})^{-1}\mathbf{M}_{\text{vv}}^{-1}\mathbf{C}_{\text{coll}}. \tag{3.14}$$

This yields the equation (3.10) which completes the proof. □

The final quadrature update step (2.13) can also be written in the following matrix form

$$\begin{pmatrix} x_{n+1} \\ v_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ v_0 \end{pmatrix} + \begin{pmatrix} \Delta t^2\mathbf{q}\mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \Delta t\mathbf{q} \end{pmatrix} \mathbf{F}\mathbf{U}^{k+1} \tag{3.15}$$

where $\mathbf{0}$ is an $(M+1)$-dimensional vector with all elements equal to zero. Insert $\mathbf{U}^{k+1}$ from (3.10) into (3.15) yields

$$\begin{pmatrix} x_{n+1} \\ v_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ v_0 \end{pmatrix} + \begin{pmatrix} \Delta t^2\mathbf{q}\mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \Delta t\mathbf{q} \end{pmatrix} \mathbf{F}\mathbf{P}_{\text{sdc}}^{k+1}\mathbf{U}_0. \tag{3.16}$$

Alternatively,

$$\begin{pmatrix} x_{n+1} \\ v_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ v_0 \end{pmatrix} + \begin{pmatrix} \Delta t^2\mathbf{q}\mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \Delta t\mathbf{q} \end{pmatrix} \mathbf{F}\mathbf{P}_{\text{sdc}}^{k+1}\bar{\mathbf{1}} \begin{pmatrix} x_0 \\ v_0 \end{pmatrix} \tag{3.17}$$

where

$$\bar{\mathbf{1}} = \begin{pmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{pmatrix} \in \mathbb{R}^{2(M+1)\times 2} \quad \text{with} \quad \mathbf{1} = (1, 1, \ldots, 1)^T \in \mathbb{R}^{M+1}.$$

SDC sweep from t_n to t_{n+1} for the damped harmonic oscillator becomes

$$\begin{pmatrix} x_{n+1} \\ v_{n+1} \end{pmatrix} = \left(\begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} \Delta t^2\mathbf{q}\mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \Delta t\mathbf{q} \end{pmatrix} \mathbf{F}\mathbf{P}_{\text{sdc}}^{k+1}\bar{\mathbf{1}} \right) \begin{pmatrix} x_0 \\ v_0 \end{pmatrix} \tag{3.18}$$

and the stability function of the SDC sweep is

$$R(\Delta t\kappa, \Delta t\mu) = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} \Delta t^2\mathbf{q}\mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \Delta t\mathbf{q} \end{pmatrix} \mathbf{F}\mathbf{P}_{\text{sdc}}^{k+1}\bar{\mathbf{1}} \tag{3.19}$$

Stability, in the sense that x_n and v_n remain bounded as $n \to \infty$, is ensured if $(\Delta t\kappa, \Delta t\mu)$ lies within the stability domain

$$\Omega_{\text{stab}} := \left\{ (\Delta t\kappa, \Delta t\mu) \in \mathbb{R}_+^2 : \rho(R(\Delta t\kappa, \Delta t\mu)) < 1 \right\}. \tag{3.20}$$

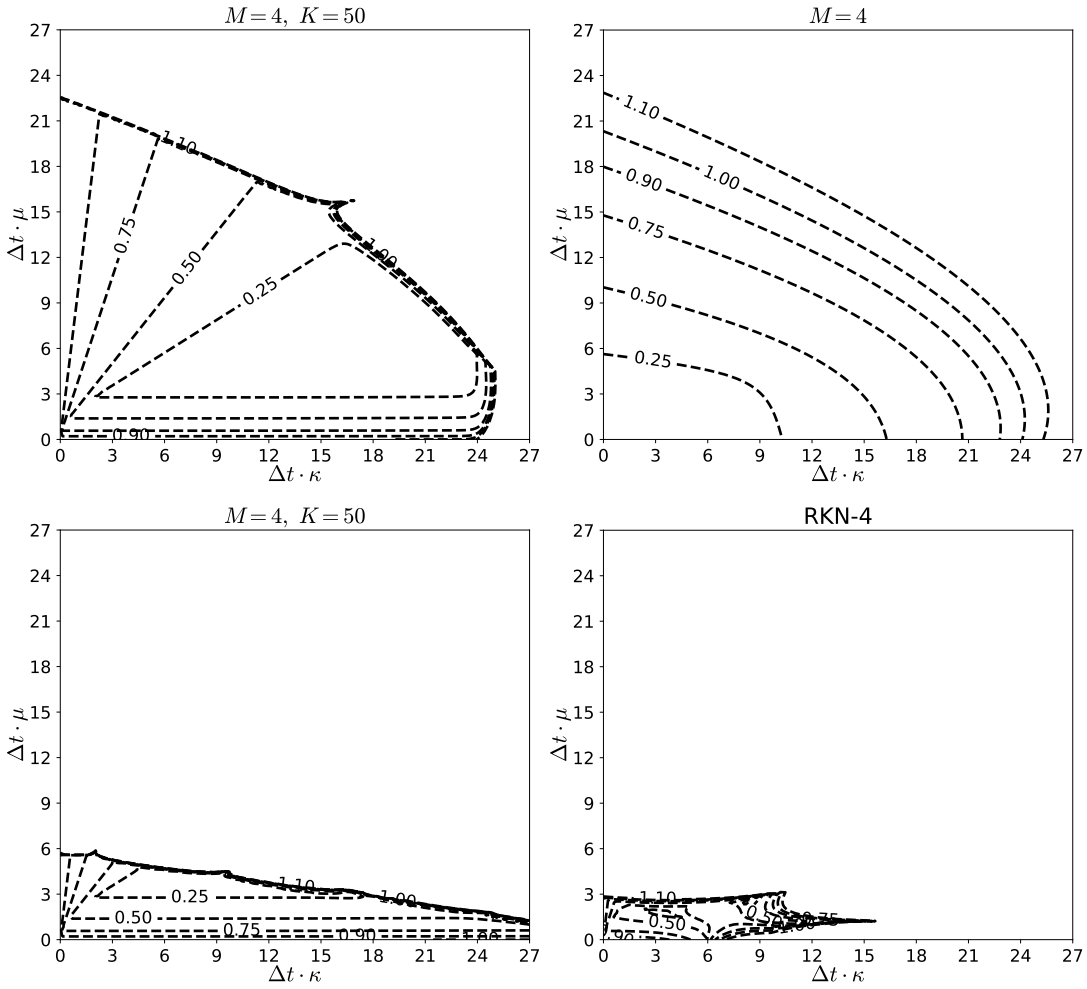

Figure 3.1: Stability domain for $K = 50$ iterations (upper left) and convergence domain (upper right) of SDC with $M = 4$ Gauss-Legendre quadrature nodes. Stability domain of the Picard iteration with $K = 50$ iterations and $M = 4$ nodes (lower right) and stability domain of RKN-4 (lower right). [1, Figure 1]

Figure 3.1 shows the stability domain after $K = 50$ iterations (upper left) and the convergence domain (upper right) for $M = 4$ Gauss-Legendre nodes. Note that how the boundaries of the convergence and stability domain coincide. Figure 3.1 also compares the stability domain of the Picard iteration (3.10) (lower left) and RKN-4 (lower right) with SDC iteration (2.23) (upper left) for $K = 50$. For the undamped system with $\mu = 0$, the Picard iteration converges more that $\Delta t\kappa = 27$ while SDC only converges until $\Delta t\kappa = 24$, although neither method will provide accuracy for a low resolution. However, once damping is added to the system SDC converges for a much larger range of parameters. In particular, SDC converges for the stiff case with very strong damping while Picard does not. However, in all cases the stability domain of SDC is much larger than that of Picard and RKN-4.

Figure 3.2 presents the stability domains of the SDC iteration for $K = 50$ and $M = 3$ using various quadrature rules. For the undamped system with $\mu = 0$, Gauss-Legendre collocation nodes are the most suitable choice. However, as damping increases, Gauss-

Lobatto nodes cover a larger stability domain compared to the other collocation nodes. In contrast, Gauss-Radau right nodes exhibit the smallest stability domain. Thus, we observe that the optimal choice of quadrature rule depends on the type of problem.

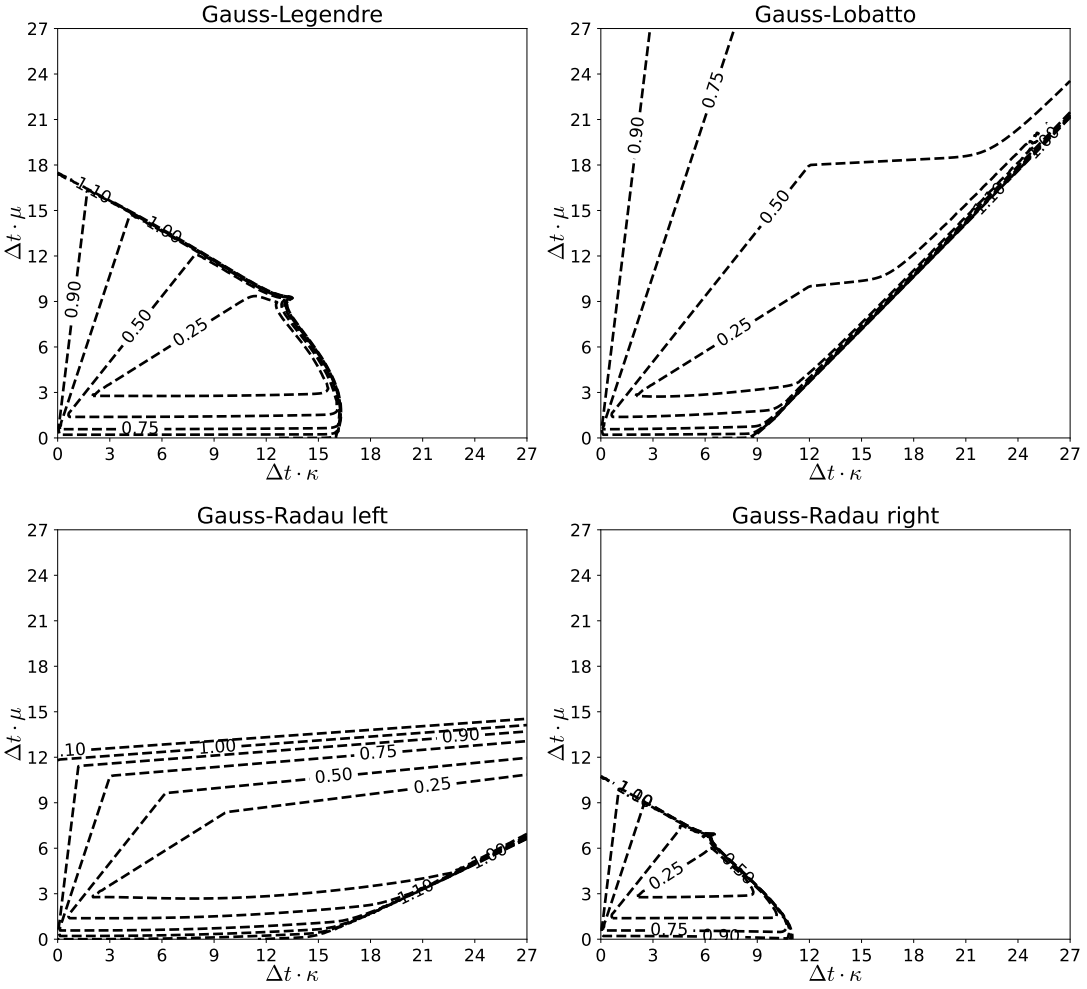

Figure 3.2: Stability domains for $K = 50$ iterations of SDC with $M = 3$ nodes using different types of quadrature rule: Gauss-Legendre (top left), Gauss-Lobatto (top right), Gauss-Radau left (bottom left) and Gauss-Radau right (bottom right).

Figure 3.3 illustrates how the stability domain of SDC varies with the number of iterations K. For $K = 1$, the stability domain is noticeably smaller than the convergence domain. Interestingly, for $K = 2$, the entire displayed range becomes stable—perhaps due to the findings reported in the recent paper [52]. Preliminary parameter searches suggest that Lobatto nodes, in particular, often yield methods that remain stable under very strong damping (up to $\Delta t\mu \approx 1000$), though a robust heuristic for this behavior has not yet been identified. With $K = 3$ iterations, parts of the parameter range become unstable again, however, the stability domain remains significantly larger than for $K = 1$. Increasing to $K = 4$ iterations expands the stability domain in the direction of stronger damping but slightly reduces it with respect to a larger spring constant. Nevertheless, in all cases, the stability domain of SDC is much larger than that of Picard or RKN-4.

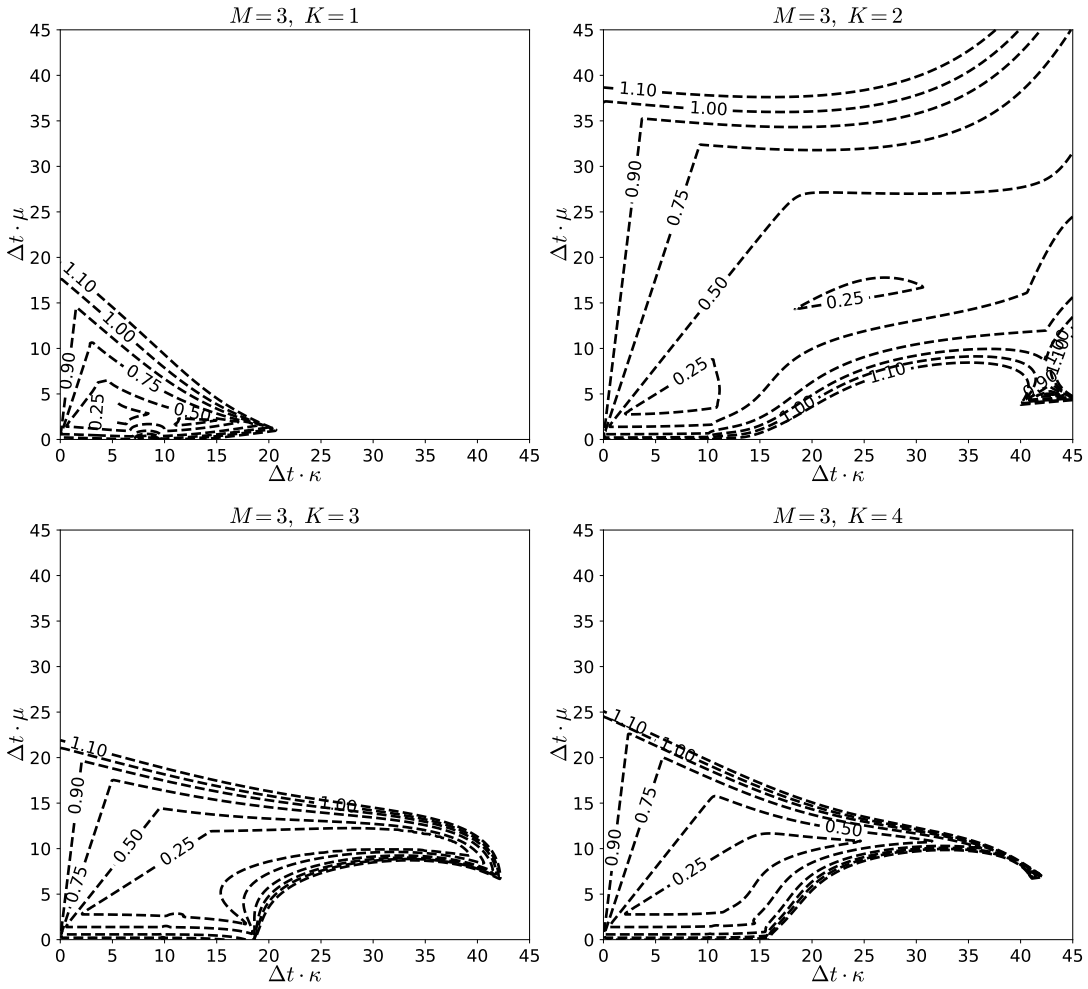

Figure 3.3: Stability domains of SDC with $M = 3$ Gauss-Legendre nodes and $K = 1, 2, 3, 4$ iterations. [1, Figure 2]

Stability for the purely oscillatory case. Table 3.1 shows the maximum stable values of $\Delta t\kappa$ for SDC and Picard iterations along the x-axis, representing the purely oscillatory case with no damping ($\mu = 0$). Choosing an even number of iterations K appears to be unfavorable for purely oscillatory systems, as both methods either become unstable or exhibit very restrictive stability limits. In contrast, if K is odd, both methods are stable for very large steps with SDC allowing even larger stable time steps than Picard iterations. At the moment, we do not have explanation the cause of the different stability limits between odd and even K.

3.2 Consistency and convergence

In this section, we demonstrate the convergence rate of the SDC method for second-order ODEs. Before presenting the main results from the published paper [1], we introduce several technical results that are crucial for proving these outcomes.

SDC (Picard)					
K	$M=2$	$M=3$	$M=4$	$M=5$	$M=6$
1	6.0 (4.7)	7.2 (4.7)	7.8 (4.7)	8.4 (4.7)	8.6 (4.7)
2	0.0 (12.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
3	0.0 (0.0)	9.6 (7.1)	26.5 (4.0)	35.3 (4.0)	55.1 (4.0)
4	11.6 (7.0)	0.2 (0.1)	0.4 (0.2)	0.4 (0.2)	0.6 (0.2)

Table 3.1: Stability limit for $\Delta t\kappa$ for $\mu = 0$ (purely oscillatory case with no damping) rounded to the first digit for SDC and Picard iteration (in brackets). [1, Table 1.]

Proposition 3.4. *For the matrices introduced in* (2.8), (2.16) *and* (2.19), *we have the following bounds*

$$\|Q\| \leq 1, \quad \|QQ\| \leq 1, \tag{3.21a}$$

$$\|Q_{\mathrm{T}}\| \leq 1, \quad \|Q_{\mathrm{x}}\| \leq \frac{3}{2}. \tag{3.21b}$$

Proof. The norm of the Q matrix is bounded by 1 which was proven in [53]. Therefore, we have $\|QQ\| \leq \|Q\|\|Q\| \leq 1$. Also, we already know that

$$\|Q_{\mathrm{I}}\| \leq 1, \ \|Q_{\mathrm{E}}\| \leq 1$$

from [54, Lemma 3.1]. Furthermore, it holds that

$$\|Q_{\mathrm{E}} \circ Q_{\mathrm{E}}\| \leq \Delta\tau_1^2 + \cdots + \Delta\tau_M^2 \leq \Delta\tau_1 + \cdots + \Delta\tau_M \leq 1. \tag{3.22}$$

Thus,

$$\|Q_{\mathrm{T}}\| \leq \frac{1}{2}(\|Q_{\mathrm{E}}\| + \|Q_{\mathrm{I}}\|) \leq 1 \tag{3.23}$$

and

$$\|Q_{\mathrm{x}}\| \leq \|Q_{\mathrm{E}}\|\|Q_{\mathrm{T}}\| + \frac{1}{2}\|Q_{\mathrm{E}}\| \leq \frac{3}{2}$$

which completes the proof. □

Proposition 3.5. *Let f be a Lipschitz continuous function with Lipschitz constant L and $(\mathbf{X}, \mathbf{V})$ be the solution to the collocation problem defined in* (2.7). *Let $(\mathbf{X}^k, \mathbf{V}^k)$ be approximate solutions provided by the SDC iteration* (2.25). *Assume that Δt satisfies*

$$\Delta t \leq (1-\delta)/L \text{ and } \Delta t^2 < \frac{1}{3} \tag{3.24}$$

for some positive constant $0 < \delta < 1$. Then, the following hold

$$\|\mathbf{X} - \mathbf{X}^k\| \leq C_1 L \Delta t^2 (\|\mathbf{X} - \mathbf{X}^{k-1}\| + \|\mathbf{V} - \mathbf{V}^{k-1}\| + \|\mathbf{V} - \mathbf{V}^k\|), \tag{3.25a}$$

$$\|\mathbf{V} - \mathbf{V}^k\| \leq C_2 L \Delta t (\|\mathbf{V} - \mathbf{V}^{k-1}\| + \|\mathbf{X} - \mathbf{X}^{k-1}\| + \|\mathbf{X} - \mathbf{X}^k\|) \tag{3.25b}$$

with constants C_1, C_2 independent of Δt. If f does not depend on v, we have

$$\|\mathbf{X} - \mathbf{X}^k\| \leq C_1 L \Delta t^2 \|\mathbf{X} - \mathbf{X}^{k-1}\|, \tag{3.26a}$$

$$\|\mathbf{V} - \mathbf{V}^k\| \leq 2L\Delta t (\|\mathbf{X} - \mathbf{X}^{k-1}\| + \|\mathbf{X} - \mathbf{X}^k\|). \tag{3.26b}$$

Proof. To prove (3.25a), we subtract (2.25a) from (2.7a) to obtain

$$\mathbf{X} - \mathbf{X}^k = \Delta t^2 \mathbf{Q}\mathbf{Q}(F(\mathbf{X}, \mathbf{V}) - F(\mathbf{X}^{k-1}, \mathbf{V}^{k-1})) + \Delta t^2 \mathbf{Q}_{\mathrm{x}}(F(\mathbf{X}^{k-1}, \mathbf{V}^{k-1}) - F(\mathbf{X}^k, \mathbf{V}^k)).$$

After taking a norm of the above expression

$$\|\mathbf{X} - \mathbf{X}^k\| \leq \Delta t^2 \|\mathbf{Q}\mathbf{Q}\| \|F(\mathbf{X}, \mathbf{V}) - F(\mathbf{X}^k, \mathbf{V}^k)\| + \Delta t^2 \|\mathbf{Q}_{\mathrm{x}}\| \|F(\mathbf{X}^{k-1}, \mathbf{V}^{k-1}) - F(\mathbf{X}^k, \mathbf{V}^k)\|.$$

Then, we apply the triangle inequality and use the Lipschitz continuity of f to obtain

$$\begin{aligned}\|\mathbf{X} - \mathbf{X}^k\| \leq\ & L\Delta t^2 \|\mathbf{Q}\mathbf{Q}\| (\|\mathbf{X} - \mathbf{X}^{k-1}\| + \|\mathbf{V} - \mathbf{V}^{k-1}\|) \\ & + L\Delta t^2 \|\mathbf{Q}_{\mathrm{x}}\| (\|\mathbf{X} - \mathbf{X}^{k-1}\| + \|\mathbf{V} - \mathbf{V}^{k-1}\|) \\ & + L\Delta t^2 \|\mathbf{Q}_{\mathrm{x}}\| (\|\mathbf{X} - \mathbf{X}^k\| + \|\mathbf{V} - \mathbf{V}^k\|).\end{aligned}$$

By Proposition 3.4, we have $\|\mathbf{Q}_{\mathrm{x}}\| \leq \frac{3}{2}$ and $\|\mathbf{Q}\mathbf{Q}\| < 1$. This yields

$$\|\mathbf{X} - \mathbf{X}^k\| \leq \left(1 + \frac{3}{2}\right) L\Delta t^2 (\|\mathbf{X} - \mathbf{X}^{k-1}\| + \|\mathbf{V} - \mathbf{V}^{k-1}\|) + \frac{3}{2} L\Delta t^2 (\|\mathbf{X} - \mathbf{X}^k\| + \|\mathbf{V} - \mathbf{V}^k\|).$$

This simplifies to

$$\|\mathbf{X} - \mathbf{X}^k\| \leq \frac{5L\Delta t^2}{2 - 3L\Delta t^2} (\|\mathbf{X} - \mathbf{X}^{k-1}\| + \|\mathbf{V} - \mathbf{V}^{k-1}\|) + \frac{3L\Delta t^2}{2 - 3L\Delta t^2} \|\mathbf{V} - \mathbf{V}^k\|.$$

By assumption, we know that $\Delta t^2 < \frac{1}{3}$, thus $1 - \frac{3}{2} L\Delta t^2 \geq 1 - L\Delta t \geq \delta > 0$, this yields

$$\frac{5}{2 - 3L\Delta t^2} \leq \frac{5}{2\delta} =: C_1 \text{ and also } \frac{5}{2\delta} \geq \frac{3}{2\delta}.$$

Hence,

$$\|\mathbf{X} - \mathbf{X}^k\| \leq C_1 L\Delta t^2 (\|\mathbf{X} - \mathbf{X}^{k-1}\| + \|\mathbf{V} - \mathbf{V}^{k-1}\| + \|\mathbf{V} - \mathbf{V}^k\|). \tag{3.27}$$

We can prove (3.26b) in a similar manner. Subtract (2.7b) from (2.25b) to obtain

$$\mathbf{V} - \mathbf{V}^k = \Delta t \mathbf{Q}(F(\mathbf{X}, \mathbf{V}) - F(\mathbf{X}^{k-1}, \mathbf{V}^{k-1})) + \Delta t \mathbf{Q}_{\mathrm{T}}(F(\mathbf{X}^{k-1}, \mathbf{V}^{k-1}) - F(\mathbf{X}^k, \mathbf{V}^k))$$

Then,

$$\|\mathbf{V} - \mathbf{V}^k\| \leq \Delta t \|\mathbf{Q}\| \|F(\mathbf{X}, \mathbf{V}) - F(\mathbf{X}^{k-1}, \mathbf{V}^{k-1})\| + \Delta t \|\mathbf{Q}_{\mathrm{T}}\| \|F(\mathbf{X}^{k-1}, \mathbf{V}^{k-1}) - F(\mathbf{X}^k, \mathbf{V}^k)\|$$

Since f is Lipschitz continuous, we obtain

$$\begin{aligned}\|\mathbf{V} - \mathbf{V}^k\| \leq\ & L\Delta t \|\mathbf{Q}\| (\|\mathbf{X} - \mathbf{X}^{k-1}\| + \|\mathbf{V} - \mathbf{V}^{k-1}\|) \\ & + L\Delta t \|\mathbf{Q}_{\mathrm{T}}\| (\|\mathbf{X} - \mathbf{X}^{k-1}\| + \|\mathbf{V} - \mathbf{V}^{k-1}\| + \|\mathbf{X} - \mathbf{X}^k\| + \|\mathbf{V} - \mathbf{V}^k\|).\end{aligned}$$

We know that $\|\mathbf{Q}\| \leq 1$ and $\|\mathbf{Q}_{\mathrm{T}}\| \leq 1$ by Proposition 3.4, so it follows that

$$\|\mathbf{V} - \mathbf{V}^k\| \leq 2L\Delta t (\|\mathbf{X} - \mathbf{X}^{k-1}\| + \|\mathbf{V} - \mathbf{V}^{k-1}\|) + L\Delta t (\|\mathbf{X} - \mathbf{X}^k\| + \|\mathbf{V} - \mathbf{V}^k\|).$$

By assumption $1 - L\Delta t \geq \delta > 0$, we obtain

$$\|\mathbf{V} - \mathbf{V}^k\| \leq \frac{2L\Delta t}{1 - L\Delta t} (\|\mathbf{X} - \mathbf{X}^{k-1}\| + \|\mathbf{V} - \mathbf{V}^{k-1}\|) + \frac{L\Delta t}{1 - L\Delta t} \|\mathbf{X} - \mathbf{X}^k\|.$$

We have

$$\frac{2}{1-L\Delta t} \leq \frac{2}{\delta} =: C_2.$$

Therefore,

$$\|\mathbf{V}-\mathbf{V}^k\| \leq C_2 L\Delta t(\|\mathbf{X}-\mathbf{X}^{k-1}\| + \|\mathbf{V}-\mathbf{V}^{k-1}\| + \|\mathbf{X}-\mathbf{X}^k\|).$$

Now, let f be independent of v, i.e., $F(\mathbf{X},\mathbf{V}) = F(\mathbf{X})$. Subtracting (2.7) from (2.25) yields

$$\begin{aligned}\mathbf{X}-\mathbf{X}^k &= \Delta t^2\mathbf{Q}\mathbf{Q}(F(\mathbf{X})-F(\mathbf{X}^{k-1})) + \Delta t^2\mathbf{Q}_{\mathrm{x}}(F(\mathbf{X}^{k-1})-F(\mathbf{X}^k)),\\ \mathbf{V}-\mathbf{V}^k &= \Delta t\mathbf{Q}(F(\mathbf{X})-F(\mathbf{X}^{k-1})) + \Delta t\mathbf{Q}_{\mathrm{T}}(F(\mathbf{X}^{k-1})-F(\mathbf{X}^k)).\end{aligned}$$

Taking a norm and since f is Lipschitz continuous as well as triangle inequality, it holds

$$\begin{aligned}\|\mathbf{X}-\mathbf{X}^k\| &\leq L\Delta t^2\|\mathbf{Q}\mathbf{Q}\|\|\mathbf{X}-\mathbf{X}^{k-1}\| + L\Delta t^2\|\mathbf{Q}_{\mathrm{x}}\|(\|\mathbf{X}-\mathbf{X}^{k-1})\| + \|\mathbf{X}-\mathbf{X}^k\|),\\ \|\mathbf{V}-\mathbf{V}^k\| &\leq L\Delta t\|\mathbf{Q}\|\|\mathbf{X}-\mathbf{X}^{k-1}\| + L\Delta t\|\mathbf{Q}_{\mathrm{T}}\|(\|\mathbf{X}-\mathbf{X}^{k-1}\| + \|\mathbf{X}-\mathbf{X}^k\|).\end{aligned}$$

Applying Proposition 3.4 and simplifying the expression, we directly obtain

$$\begin{aligned}\|\mathbf{X}-\mathbf{X}^k\| &\leq C_1 L\Delta t^2\|\mathbf{X}-\mathbf{X}^{k-1}\|,\\ \|\mathbf{V}-\mathbf{V}^k\| &\leq 2L\Delta t(\|\mathbf{X}-\mathbf{X}^{k-1}\| + \|\mathbf{X}-\mathbf{X}^k\|).\end{aligned}$$

This completes the proof of the proposition. □

The following theorem provides us the error bound for SDC at the quadrature nodes.

Theorem 3.6. *Consider the initial value problem* (2.1) *and let f be Lipschitz continuous with Lipschitz constant L. If the step size Δt is sufficiently small, we have*

$$\|\mathbf{X}-\mathbf{X}^k\| \leq \tilde{C}_1 L^k \Delta t^{k+k_0+1}, \tag{3.28a}$$

$$\|\mathbf{V}-\mathbf{V}^k\| \leq \tilde{C}_2 L^k \Delta t^{k+k_0} \tag{3.28b}$$

with constants $\tilde{C}_1$, $\tilde{C}_2$ independent of Δt, and k_0 the order of the procedure used to generate the starting value $\mathbf{U}^0$ for the SDC iteration; see Remark 2.3. If f is independent of v, then

$$\|\mathbf{X}-\mathbf{X}^k\| \leq \hat{C}_1 L^k \Delta t^{2k+k_0}, \tag{3.29a}$$

$$\|\mathbf{V}-\mathbf{V}^k\| \leq \hat{C}_2 L^k \Delta t^{2k+k_0-1} \tag{3.29b}$$

with constants $\hat{C}_1$, $\hat{C}_2$ independent of Δt.

Proof. First, consider the case where the right-hand side function f is independent of v. By substituting (3.26a) into (3.26b), we obtain

$$\begin{aligned}\|\mathbf{V}-\mathbf{V}^k\| &\leq 2L\Delta t(\|\mathbf{X}-\mathbf{X}^{k-1}\| + C_1 L\Delta t^2\|\mathbf{X}-\mathbf{X}^{k-1}\|)\\ &= 2L\Delta t\|\mathbf{X}-\mathbf{X}^{k-1}\| + 2C_1L^2\Delta t^3\|\mathbf{X}-\mathbf{X}^{k-1}\|.\end{aligned}$$

This yields,

$$\|\mathbf{X}-\mathbf{X}^k\| \leq C_1 L\Delta t^2\|\mathbf{X}-\mathbf{X}^{k-1}\|,$$

$$\|\mathbf{V}-\mathbf{V}^k\| \le 2L\Delta t\|\mathbf{X}-\mathbf{X}^{k-1}\| + 2C_1L^2\Delta t^3\|\mathbf{X}-\mathbf{X}^{k-1}\|.$$

By recursive substitution, we obtain

$$\|\mathbf{X}-\mathbf{X}^k\| \le C_1^kL^k\Delta t^{2k}\|\mathbf{X}-\mathbf{X}^0\|, \tag{3.30a}$$
$$\|\mathbf{V}-\mathbf{V}^k\| \le 2C_1^{k-1}L^k\Delta t^{2k-1}\|\mathbf{X}-\mathbf{X}^0\| + 2C_1^kL^{k+1}\Delta t^{2k+1}\|\mathbf{X}-\mathbf{X}^0\|. \tag{3.30b}$$

For starting values $\mathbf{X}^0$ and $\mathbf{V}^0$ with order of k_0 for the SDC iteration, so we have

$$\|\mathbf{X}-\mathbf{X}^0\| \le C_0\Delta t^{k_0}, \tag{3.31a}$$
$$\|\mathbf{V}-\mathbf{V}^0\| \le C_0\Delta t^{k_0}, \tag{3.31b}$$

where the constant C_0 is independent of Δt. Applying (3.31) to (3.30a), we find that

$$\|\mathbf{X}-\mathbf{X}^k\| \le C_1^kL^k\Delta t^{2k}\|\mathbf{X}-\mathbf{X}^0\| \le C_1^kC_0L^k\Delta t^{2k+k_0}.$$

Alternatively, using (3.31) in (3.30b), we get

$$\begin{aligned}\|\mathbf{V}-\mathbf{V}^k\| &\le 2C_1^{k-1}L^k\Delta t^{2k-1}\|\mathbf{X}-\mathbf{X}^0\| + 2C_1^kL^{k+1}\Delta t^{2k+1}\|\mathbf{X}-\mathbf{X}^0\| \\ &\le 2C_1^{k-1}C_0L^k\Delta t^{2k+k_0-1} + 2C_1^kC_0L^{k+1}\Delta t^{2k+k_0+1} \\ &= 2C_0C_1^{k-1}(1+C_1L\Delta t^2)L^k\Delta t^{2k+k_0-1}.\end{aligned}$$

Because $\Delta t^2 \le \frac{1}{3}$ by the assumption of Proposition 3.5, the following estimate is valid

$$2C_0C_1^{k-1}(1+C_1L\Delta t^2) \le 2C_0C_1^{k-1}\left(1+\frac{C_1L}{3}\right) =: \hat{C}_2.$$

Thus,

$$\|\mathbf{X}-\mathbf{X}^k\| \le \hat{C}_1L^k\Delta t^{2k+k_0},$$
$$\|\mathbf{V}-\mathbf{V}^k\| \le \hat{C}_2L^k\Delta t^{2k+k_0-1},$$

where $\hat{C}_1 := C_1^kC_0$.

For the general case where f depends on v, we use estimate (3.25). First, we insert $\|\mathbf{X}-\mathbf{X}^k\|$ from (3.25a) on the right-hand side of the inequality (3.25b) to obtain

$$\begin{aligned}\|\mathbf{X}-\mathbf{X}^k\| &\le C_1L\Delta t^2(\|\mathbf{X}-\mathbf{X}^{k-1}\|+\|\mathbf{V}-\mathbf{V}^{k-1}\|+C_2L\Delta t(\|\mathbf{V}-\mathbf{V}^{k-1}\|+\|\mathbf{X}-\mathbf{X}^{k-1}\|+\|\mathbf{X}-\mathbf{X}^k\|)) \\ &= (C_1L\Delta t^2 + C_1C_2L^2\Delta t^3)(\|\mathbf{X}-\mathbf{X}^{k-1}\| + \|\mathbf{V}-\mathbf{V}^{k-1}\|) + C_1C_2L^2\Delta t^3\|\mathbf{X}-\mathbf{X}^k\|.\end{aligned}$$

If Δt is sufficiently small such that $1 - C_1C_2L^2\Delta t^3 \ge \tilde{\delta} > 0$, then

$$\|\mathbf{X}-\mathbf{X}^k\| \le \frac{C_1L\Delta t^2 + C_1C_2L^2\Delta t^3}{1-C_1C_2L^2\Delta t^3}(\|\mathbf{X}-\mathbf{X}^{k-1}\| + \|\mathbf{V}-\mathbf{V}^{k-1}\|).$$

Similarly, substitute the expression for $\|\mathbf{V}-\mathbf{V}^k\|$ from (3.25b) into (3.25a) to obtain

$$\begin{aligned}\|\mathbf{V}-\mathbf{V}^k\| &\le C_2L\Delta t(\|\mathbf{V}-\mathbf{V}^{k-1}\|+\|\mathbf{X}-\mathbf{X}^{k-1}\|+C_1L\Delta t^2(\|\mathbf{X}-\mathbf{X}^{k-1}\|+\|\mathbf{V}-\mathbf{V}^{k-1}\|+\|\mathbf{V}-\mathbf{V}^k\|)) \\ &= (C_2L\Delta t + C_1C_2L^2\Delta t^3)(\|\mathbf{X}-\mathbf{X}^{k-1}\| + \|\mathbf{V}-\mathbf{V}^{k-1}\|) + C_1C_2L^2\Delta t^3\|\mathbf{V}-\mathbf{V}^k\|.\end{aligned}$$

Therefore,

$$\|\mathbf{V}-\mathbf{V}^k\| \leq \frac{C_2 L\Delta t + C_1 C_2 L^2 \Delta t^3}{1 - C_1 C_2 L^2 \Delta t^3}(\|\mathbf{X}-\mathbf{X}^{k-1}\| + \|\mathbf{V}-\mathbf{V}^{k-1}\|).$$

Define

$$m_1 := \frac{L\Delta t^2(C_1 + C_1 C_2 L\Delta t)}{1 - C_1 C_2 L^2 \Delta t^3},\ m_2 := \frac{L\Delta t(C_2 + C_1 C_2 L\Delta t^2)}{1 - C_1 C_2 L^2 \Delta t^3}$$

and we get the following system of inequalities

$$\begin{aligned}\|\mathbf{X}-\mathbf{X}^k\| &\leq m_1(\|\mathbf{X}-\mathbf{X}^{k-1}\| + \|\mathbf{V}-\mathbf{V}^{k-1}\|),\\ \|\mathbf{V}-\mathbf{V}^k\| &\leq m_2(\|\mathbf{X}-\mathbf{X}^{k-1}\| + \|\mathbf{V}-\mathbf{V}^{k-1}\|).\end{aligned}$$

These can be written in the following matrix form

$$\begin{pmatrix}\|\mathbf{X}-\mathbf{X}^k\|\\ \|\mathbf{V}-\mathbf{V}^k\|\end{pmatrix} \leq \begin{pmatrix}m_1 & m_1\\ m_2 & m_2\end{pmatrix}\begin{pmatrix}\|\mathbf{X}-\mathbf{X}^{k-1}\|\\ \|\mathbf{V}-\mathbf{V}^{k-1}\|\end{pmatrix}.$$

Recursive insertion yields

$$\begin{pmatrix}\|\mathbf{X}-\mathbf{X}^k\|\\ \|\mathbf{V}-\mathbf{V}^k\|\end{pmatrix} \leq \begin{pmatrix}m_1 & m_1\\ m_2 & m_2\end{pmatrix}^k\begin{pmatrix}\|\mathbf{X}-\mathbf{X}^0\|\\ \|\mathbf{V}-\mathbf{V}^0\|\end{pmatrix} =: M^k\begin{pmatrix}\|\mathbf{X}-\mathbf{X}^0\|\\ \|\mathbf{V}-\mathbf{V}^0\|\end{pmatrix}. \tag{3.32}$$

It is easy to show that by induction that $M^k = (m_1+m_2)^{k-1}M$ so that (3.32) becomes

$$\begin{pmatrix}\|\mathbf{X}-\mathbf{X}^k\|\\ \|\mathbf{V}-\mathbf{V}^k\|\end{pmatrix} \leq \begin{pmatrix}m_1(m_1+m_2)^{k-1} & m_1(m_1+m_2)^{k-1}\\ m_2(m_1+m_2)^{k-1} & m_2(m_1+m_2)^{k-1}\end{pmatrix}\begin{pmatrix}\|\mathbf{X}-\mathbf{X}^0\|\\ \|\mathbf{V}-\mathbf{V}^0\|\end{pmatrix},$$

i.e.,

$$\begin{aligned}\|\mathbf{X}-\mathbf{X}^k\| &\leq m_1(m_1+m_2)^{k-1}(\|\mathbf{X}-\mathbf{X}^0\| + \|\mathbf{V}-\mathbf{V}^0\|),\\ \|\mathbf{V}-\mathbf{V}^k\| &\leq m_2(m_1+m_2)^{k-1}(\|\mathbf{X}-\mathbf{X}^0\| + \|\mathbf{V}-\mathbf{V}^0\|).\end{aligned}$$

For $\Delta t < \frac{1}{\sqrt{3}}$ and $1 - C_1 C_2 L^2 \Delta t^3 \geq \tilde{\delta} > 0$, the following estimates hold

$$\begin{aligned}\frac{C_2 + C_1 C_2 L\Delta t^2}{1 - C1 C_2 L^2 \Delta t^3} &\leq \frac{3C_2 + C_1 C_2 L}{3\delta} =: C_3,\\ \frac{C_1 + C_1 C_2 L\Delta t}{1 - C_1 C_2 L^2 \Delta t^3} &\leq \frac{\sqrt{3}C_1 + C_1 C_2 L}{\sqrt{3}\tilde{\delta}} =: C_4\end{aligned}$$

and so we get

$$\begin{aligned}\|\mathbf{X}-\mathbf{X}^k\| &\leq C_4 L\Delta t^2(C_3 L\Delta t + C_4 L\Delta t^2)^{k-1}(\|\mathbf{X}-\mathbf{X}^0\| + \|\mathbf{V}-\mathbf{V}^0\|),\\ \|\mathbf{V}-\mathbf{V}^k\| &\leq C_3 L\Delta t(C_3 L\Delta t + C_4 L\Delta t^2)^{k-1}(\|\mathbf{X}-\mathbf{X}^0\| + \|\mathbf{V}-\mathbf{V}^0\|).\end{aligned}$$

Using the results from (3.31) yields

$$\begin{aligned}\|\mathbf{X}-\mathbf{X}^k\| &\leq L^k\Delta t^{k+1} C_4(C_3 + C_4\Delta t)^{k-1}(\|\mathbf{X}-\mathbf{X}^0\| + \|\mathbf{V}-\mathbf{V}^0\|)\\ &\leq L^k\Delta t^{k+1} C_4(C_3 + C_4\Delta t)^{k-1}(C_0\Delta t^{k_0} + C_0\Delta t^{k_0})\end{aligned}$$

$$\leq 2C_4C_0(C_3 + C_4\Delta t)^{k-1}L^k\Delta t^{k+k_0+1}.$$

Analogously, these computations can be done for the velocity variable $\mathbf{V}$ which yields

$$\|\mathbf{V} - \mathbf{V}^k\| \leq L^k\Delta t^k C_3(C_3 + C_4\Delta t)^{k-1}(\|\mathbf{X} - \mathbf{X}^0\| + \|\mathbf{V} - \mathbf{V}^0\|)$$
$$\leq 2C_3C_0(C_3 + C_4\Delta t)^{k-1}L^k\Delta t^{k+k_0}.$$

With $\Delta t < \frac{1}{\sqrt{3}}$, we have that

$$2C_4C_0(C_3 + C_4\Delta t)^{k-1} \leq 2C_4C_0\left(C_3 + \frac{1}{\sqrt{3}}C_4\right)^{k-1} := \tilde{C}_1,$$
$$2C_3C_0(C_3 + C_4\Delta t)^{k-1} \leq 2C_3C_0\left(C_3 + \frac{1}{\sqrt{3}}C_4\right)^{k-1} := \tilde{C}_2.$$

Therefore,

$$\|\mathbf{X} - \mathbf{X}^k\| \leq \tilde{C}_1L^k\Delta t^{k+k_0+1},$$
$$\|\mathbf{V} - \mathbf{V}^k\| \leq \tilde{C}_2L^k\Delta t^{k+k_0}$$

which completes the proof. □

Convergence rate. We state and prove our main theoretical results on the convergence rate of SDC for second-order problems in this section. The strategy we employ to prove the theorem below follows approaches used for the first-order case [55, 28].

Theorem 3.7. *Consider the initial value problem* (2.1) *with a Lipschitz continuous function f with Lipschitz constant L. Let p denote the order of the quadrature rule, and assume that $f \circ (x, v) \in \mathbf{C}^p([t_n, t_{n+1}])$ and that there exists a positive constant G such that $\|\frac{d^p}{dt^p}(f \circ (x, v))\| \leq G$. Let $(x(t_{n+1}), v(t_{n+1}))$ be the exact solutions to* (2.1) *and (x^k_{n+1}, v^k_{n+1}) be the approximate solutions to* (2.12) *provided by SDC after k iterations* (2.25)*. If the step size Δt is sufficiently small, then*

$$|x(t_{n+1}) - x^k_{n+1}| = \mathcal{O}(G\Delta t^{p+1}) + \mathcal{O}(L^{k+1}\Delta t^{k+k_0+2}),$$
$$|v(t_{n+1}) - v^k_{n+1}| = \mathcal{O}(G\Delta t^{p+1}) + \mathcal{O}(L^{k+1}\Delta t^{k+k_0+1}),$$

where k_0 denotes the approximation order of the base method used to generate $\mathbf{U}^0$; see Remark 2.3. Moreover, if f is independent of v, we have

$$|x(t_{n+1}) - x^k_{n+1}| = \mathcal{O}(G\Delta t^{p+1}) + \mathcal{O}(L^{k+1}\Delta t^{2k+k_0+2}), \tag{3.34a}$$
$$|v(t_{n+1}) - v^k_{n+1}| = \mathcal{O}(G\Delta t^{p+1}) + \mathcal{O}(L^{k+1}\Delta t^{2k+k_0+1}). \tag{3.34b}$$

Proof. We substitute (2.7) into (2.13) to find the update steps (x_{n+1}, v_{n+1}) of the collocation method. Additionally, we determine the SDC method update formula (x^k_{n+1}, v^k_{n+1}) from (2.25) by plugging it into (2.13), subtract and use the Cauchy-Schwarz inequality [56, pp. 171-177] and Lipschitz continuity to obtain

$$|v_{n+1} - v^k_{n+1}| = \Delta t|\mathbf{q}(F(\mathbf{X}, \mathbf{V}) - F(\mathbf{X}^k, \mathbf{V}^k))| \leq \Delta t\|\mathbf{q}\|\|F(\mathbf{X}, \mathbf{V}) - F(\mathbf{X}^k, \mathbf{V}^k)\|$$

$$\leq \Delta t L \|\mathbf{q}\|(\|\mathbf{X} - \mathbf{X}^k\| + \|\mathbf{V} - \mathbf{V}^k\|). \quad (3.35)$$

From [54], we know that $\|q\| \leq 1$ and apply Theorem 3.6, which implies that

$$|v_{n+1} - v_{n+1}^k| \leq \Delta t L \|\mathbf{q}\|(\|\mathbf{X} - \mathbf{X}^k\| + \|\mathbf{V} - \mathbf{V}^k\|) \leq \Delta t L^{k+1} \left(\tilde{C}_1 \Delta t^{k+k_0+1} + \tilde{C}_2 \Delta t^{k+k_0}\right)$$
$$\leq L^{k+1} \left(\tilde{C}_1 + \tilde{C}_2 \Delta t\right) \Delta t^{k+k_0+1}.$$

The entries of the qQ satisfy [11, pp.208-210]

$$\sum_{i=1}^{M} q_i q_{i,j} = q_j(1 - \tau_j), \ j = 0, \ldots, M.$$

Since $\tau_1 \leq 1$ for all $j = 1, \ldots, M$ on the unit interval, this yields

$$\|\mathbf{qQ}\| = \max_{j=1,\ldots,M} |q_j(1 - \tau_j)| \leq 1$$

By plugging (2.7a) and (2.25a) into (2.13a) to find the update steps, subtracting these update steps and then applying the Cauchy-Schwarz inequality along with Lipschitz continuity, we obtain

$$|x_{n+1} - x_{n+1}^k| = \Delta t^2 |\mathbf{qQ}(F(\mathbf{X}, \mathbf{V}) - F(\mathbf{X}^k, \mathbf{V}^k))|$$
$$\leq L \Delta t^2 \|\mathbf{qQ}\|(\|\mathbf{X} - \mathbf{X}^k\| + \|\mathbf{V} - \mathbf{V}^k\|) \quad (3.36)$$

Using Theorem 3.6 yields

$$|x_{n+1} - x_{n+1}^k| \leq \Delta t^2 L \|\mathbf{qQ}\|(\|\mathbf{X} - \mathbf{X}^k\| + \|\mathbf{V} - \mathbf{V}^k\|)$$
$$\leq \Delta t^2 L^{k+1} \left(\tilde{C}_1 \Delta t^{k+k_0+1} + \tilde{C}_2 \Delta t^{k+k_0}\right)$$
$$\leq L^{k+1} (\tilde{C}_2 + \tilde{C}_1 \Delta t) \Delta t^{k+k_0+2}.$$

We know that $\Delta t \leq 1$, so we have

$$\tilde{C}_2 + \tilde{C}_1 \Delta t \leq \tilde{C}_2 + \tilde{C}_1 =: C_L$$

and therefore

$$|x_{n+1} - x_{n+1}^k| \leq C_L L^{k+1} \Delta t^{k+k_0+2}, \quad (3.37a)$$
$$|v_{n+1} - v_{n+1}^k| \leq C_L L^{k+1} \Delta t^{k+k_0+1}. \quad (3.37b)$$

Gauss quadrature nodes satisfy the following orthogonality condition [11, Theorem 7.9]

$$\int_0^1 s^{j-1} \prod_{i=1}^{M} (s - \tau_i) ds = 0, \ j = 1, \ldots, \xi.$$

Thus, the following estimates are valid

$$|x(t_{n+1}) - x_{n+1}| \leq C_G G \Delta t^{p+1}, \quad (3.38a)$$

$$|v(t_{n+1}) - v_{n+1}| \leq C_G G \Delta t^{p+1}, \tag{3.38b}$$

where $p = M + \xi$ and C_G is a constant. We have $\xi = M$ and $p = 2M$ for Legendre nodes, $\xi = M - 1$ and $p = 2M - 1$ for Radau nodes, and $\xi = M - 2$ and $p = 2M - 2$ for Lobatto nodes. Subtracting the analytical solution from the SDC solution at time t_{n+1} and using the triangle inequality along with (3.37) and (3.38) gives the bound

$$\begin{aligned} |x(t_{n+1}) - x_{n+1}^k| &\leq |x(t_{n+1}) - x_{n+1}| + |x_{n+1} - x_{n+1}^k| \\ &\leq C_G G \Delta t^{p+1} + C_L L^{k+1} \Delta t^{k+k_0+2} \end{aligned}$$

for the position error and the bound

$$\begin{aligned} |v(t_{n+1}) - v_{n+1}^k| &\leq |v(t_{n+1}) - v_{n+1}| + |v_{n+1} - v_{n+1}^k| \\ &\leq C_G G \Delta t^{p+1} + C_L L^{k+1} \Delta t^{k+k_0+1} \end{aligned}$$

for the velocity error. In summary, the local error of second-order SDC satisfies

$$\begin{aligned} |x(t_{n+1}) - x_{n+1}^k| &= \mathcal{O}(G \Delta t^{p+1}) + \mathcal{O}(L^{k+1} \Delta t^{k+k_0+2}), \\ |v(t_{n+1}) - v_{n+1}^k| &= \mathcal{O}(G \Delta t^{p+1}) + \mathcal{O}(L^{k+1} \Delta t^{k+k_0+1}). \end{aligned}$$

When f is independent of v, (3.36) and (3.35) become

$$|x_{n+1} - x_{n+1}^k| = \Delta t^2 |\mathbf{qQ}(F(\mathbf{X}) - F(\mathbf{X}^k))| \leq L \Delta t^2 \|\mathbf{X} - \mathbf{X}^k\|$$

and

$$|v_{n+1} - v_{n+1}^k| = \Delta t |\mathbf{q}(F(\mathbf{X}) - F(\mathbf{X}^k))| \leq L \Delta t \|\mathbf{X} - \mathbf{X}^k\|.$$

Using the triangle inequality, (3.38) and Theorem 3.6 yield

$$\begin{aligned} |x(t_{n+1}) - x_{n+1}^k| &\leq |x(t_{n+1}) - x_{n+1}| + |x_{n+1} - x_{n+1}^k| \\ &\leq C_G G \Delta t^{p+1} + C_L L^{k+1} \Delta t^{2k+k_0+2} \end{aligned}$$

and

$$\begin{aligned} |v(t_{n+1}) - v_{n+1}^k| &\leq |v(t_{n+1}) - v_{n+1}| + |v_{n+1} - v_{n+1}^k| \\ &\leq C_G G \Delta t^{p+1} + C_L L^{k+1} \Delta t^{2k+k_0+1}. \end{aligned}$$

Thus, we obtain

$$\begin{aligned} |x(t_{n+1}) - x_{n+1}^k| &= \mathcal{O}(G \Delta t^{p+1}) + \mathcal{O}(L^{k+1} \Delta t^{2k+k_0+2}), \\ |v(t_{n+1}) - v_{n+1}^k| &= \mathcal{O}(G \Delta t^{p+1}) + \mathcal{O}(L^{k+1} \Delta t^{2k+k_0+1}) \end{aligned}$$

which completes the proof. □

A direct consequence of Theorem 3.7 and [57, Definition 2.1] is the following.

Theorem 3.8. *Let the right-hand side function f in (2.1) satisfy the assumptions of Theorem 3.7. Then, the global convergence rate of SDC is $p^* := \min\{p,\ k + k_0\}$. If f does not depend on v, we have $p^* := \min\{p,\ 2k + k_0\}$.*

α	1.0
t_{end}	2.0
$\vec{x}(t=0)$	$(10, 0, 0)^T$
$\vec{v}(t=0)$	$(100, 0, 100)^T$
ω_E	4.9
ω_B	25.0
N_{steps}	variable

Table 3.2: Parameters chosen for the case of a single particle in the Penning trap [3, Table 1].

3.3 Numerical examples

As a numerical example, we consider particles in a standard Penning trap [58]. Figure 3.4 shows a cylindrical version of a Penning trap device for the storage of charged particles. Here, a magnetic field (B, green arrow) runs along the trap's axis, forcing particles into circular or helical trajectories, preventing radial escape. Meanwhile, an electric field (E, red arrows), created by electrodes, confines the particles axially by forming an electrostatic potential well (for more details see [59]).

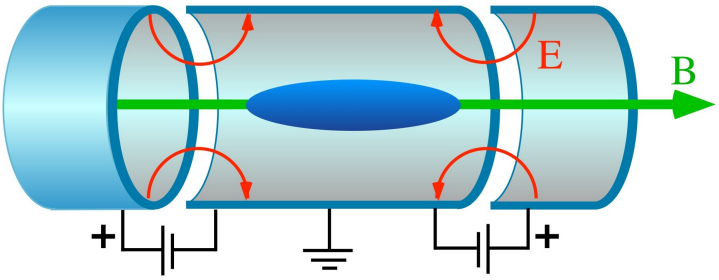

Figure 3.4: A cylindrical version of a Penning trap, with open endcraps to permit axial access. B indicates the magnetic field, and E indicates the electric field used for storage of the particles in the trap center [2].

Table 3.2 shows the list of physical parameters used in this section like in [3, 60] and choose a constant magnetic field $B = \frac{\omega_B}{\alpha} \cdot \hat{\mathbf{e}}_z$ along the $z-$ axis with frequency ω_B and the particle's charge-to-mass ratio $\alpha = \frac{q}{m}$ so that

$$v \times B = \frac{\omega_B}{\alpha} \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} v. \tag{3.41}$$

The electric field with frequency ω_E is given by

$$E(x) = -\varepsilon \frac{\omega_E^2}{\alpha} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -2 \end{pmatrix} x. \tag{3.42}$$

We provide analytical solution for the single particle Penning trap, as presented in [3]. Movement in z direction is a harmonic oscillator, decoupled from the other coordinates,

$$z(t) = z(0)\cos(\tilde{\omega}t) + \frac{v_z(0)}{\tilde{\omega}}\sin(\tilde{\omega}t), \quad \tilde{\omega} := \sqrt{-2\varepsilon}\cdot\omega_E. \tag{3.43}$$

Here, the role of ε stated before becomes obvious: For $\varepsilon = +1$, the frequency $\tilde{\omega}$ is purely imaginary and the z-trajectory diverges, for $\varepsilon = -1$ the frequency $\tilde{\omega} \in \mathbb{R}$ and the trajectory $z(t)$ corresponds to a harmonic oscillation. With the definition $w(t) := x(t) + \mathrm{i}y(t)$, the particle movement in the $x - y$ plane is given by

$$w(t) = (\mathcal{L}_+ + \mathrm{i}\mathcal{I}_+)\mathrm{e}^{-i\Omega_+ t} + (\mathcal{L}_- + \mathrm{i}\mathcal{I}_-)\mathrm{e}^{-i\Omega_- t}, \quad \Omega_\pm := \frac{1}{2}\left(\omega_B \pm \sqrt{\omega_B^2 + 4\varepsilon\omega_E^2}\right)$$

$$\mathcal{L}_- := \frac{\Omega_+ x(0) + v_y(0)}{\Omega_+ - \Omega_-},\ \mathcal{L}_+ := x(0) - \mathcal{L}_-,\ \mathcal{I}_- := \frac{\Omega_+ y(0) - v_x(0)}{\Omega_+ - \Omega_-},\ \mathcal{I}_- := y(0) - \mathcal{I}_-.$$

Note that for $\omega_B^2 < -4\varepsilon\omega_E^2$ (which can only happen for $\varepsilon = -1$), we have $\Omega_\pm \notin \mathbb{R}$ for the revolution frequency. In this case, the physical setup is unstable and the particle escape from the trap due to a too weak magnetic or too strong electric field.

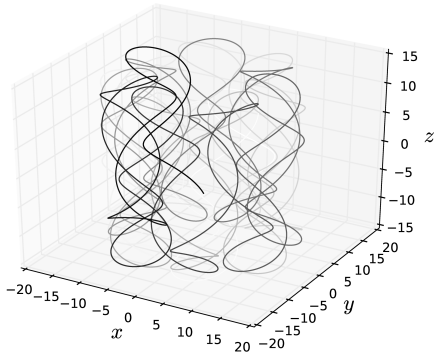

Figure 3.5: Trajectory of the particle for given parameters in Table 3.2. Time is changing by the line shade from light ($t = 0$) to dark ($t = 16$) [3, Fig. 1].

Figure 3.5 shows the analytical trajectory of a single particle in the Penning trap problem with initial guess $\mathbf{U}^0 = \mathbf{U}_0$ for the parameters given in Table 3.2 but the final time $t_{\text{end}} = 16.0$.

We validate our convergence analysis for the Penning trap problem [58]. Notably, because of the zero row in the matrix (3.41), the force along the third component is independent of the velocity v, whereas the forces along the first and second components depend on it. By examining the error in the first and third components separately, we confirm below the distinct convergence orders predicted by our theory for these cases. Note that we always initialize $\mathbf{U}^0$ with random values which means $k_0 = 0$. Even a simple copy of $\mathbf{U}_0$ was found to yield convergence orders exceeding theoretical guarantees in some cases.

3.3.1 Local error

Figure 3.6 illustrates the local position (left) and velocity (right) of SDC along the first axis as a function of the time step Δt, scaled by the frequency of the magnetic field. The

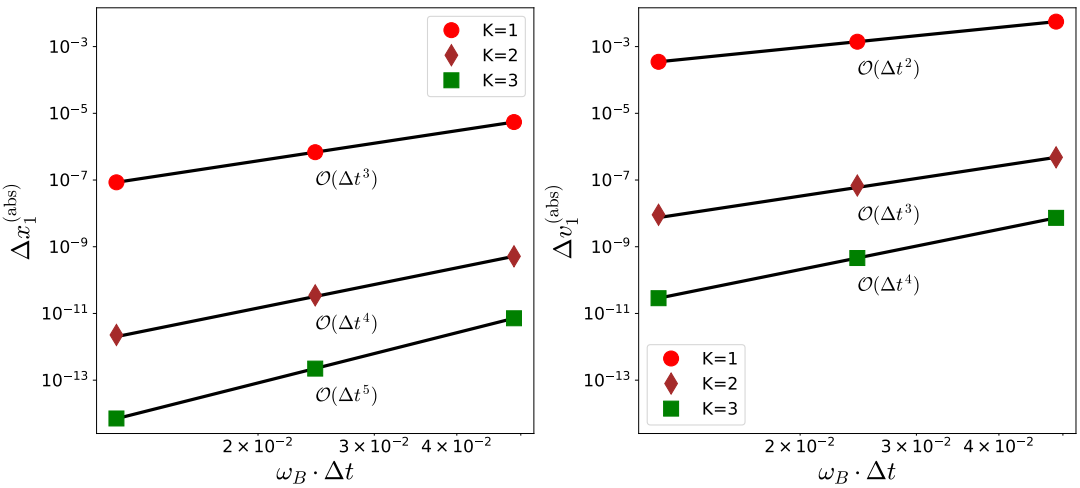

Figure 3.6: Absolute local error $\Delta x_1^{(\mathrm{abs})}$ and $\Delta v_1^{(\mathrm{abs})}$ in the first component of the particle's position (left) and velocity (right) using $K = 1, 2, 3$ SDC iterations and $M = 5$ [1, Fig. 3.].

local error is computed as the absolute difference $\Delta x_i^{(\mathrm{abs})} := |x_i^{(\mathrm{approx})} - x_i^{(\mathrm{analyt})}|$ between numerical and analytical solutions after a single time step. The index $i = 1, 2, 3$ denotes the two horizontal and one vertical axes. Consistent with our theoretical predictions, the order of the local error increases by one with each iteration, and the order of the local error in position is always one order higher than that in velocity.

Figure 3.7 shows the local error for position and velocity in the third component, where the force is independent of v, for SDC using $M = 5$ Gauss-Legendre quadrature nodes. As predicted by Theorem 3.7, the one-order difference between position and velocity persists, but the order of the local error increases by two orders with each iteration.

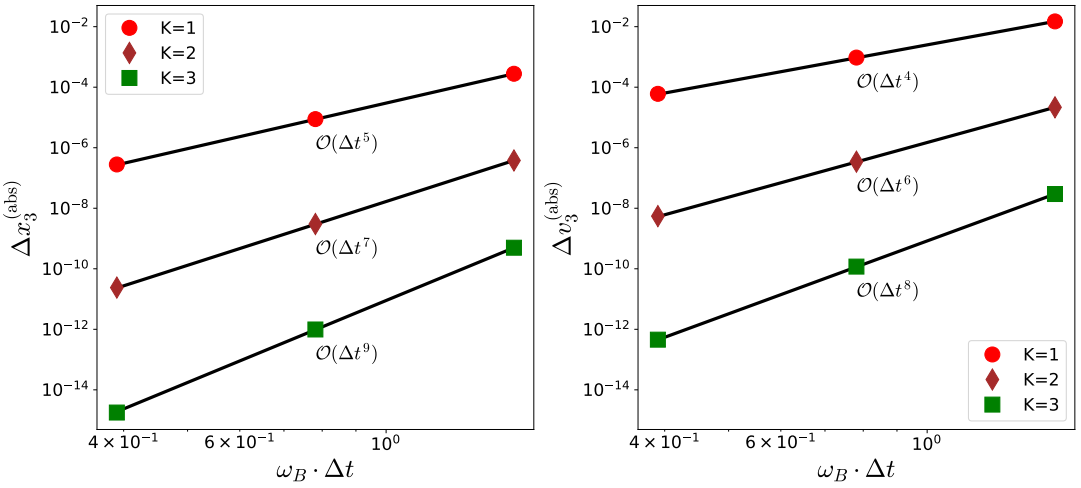

Figure 3.7: Absolute local error $\Delta x_3^{(\mathrm{abs})}$ and $\Delta v_3^{(\mathrm{abs})}$ in the third component of the particle's position (left) and velocity (right) using one, two, and three SDC iterations and 5 quadrature nodes. In line with Theorem 3.7, the order increases by two per iteration [1, Fig. 4.].

3.3.2 Global error

Figure 3.8 displays the relative global error in the v_1-component (left) and v_3-component (right) of the velocity for $M = 3$ Gauss-Legendre nodes with fixed final time $t_{\text{end}} = 2$. Since the velocity depends on the position due to the inhomogeneous magnetic and electric fields, the global error is of the same order as the lower local order of the velocity. Consistent with Theorem 3.8, we observe that in the v_1-direction, each iteration increases the global convergence order by one. In contrast, in the v_3-direction, where the force is independent of the velocity, each iteration increases the global order by two.

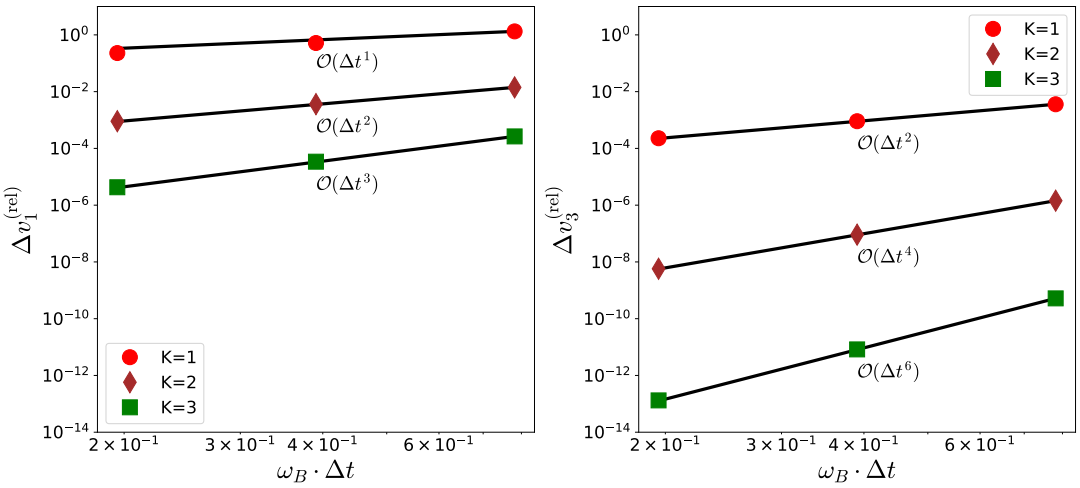

Figure 3.8: Relative global error $\Delta v_i^{(\text{rel})}$, $i = 1, 3$ in the first component of the particle's horizontal (left) and vertical (right) velocity in the Penning trap versus time step size for 3 Gauss-Legendre collocation nodes using one, two, and three SDC iterations [1, Figure 5].

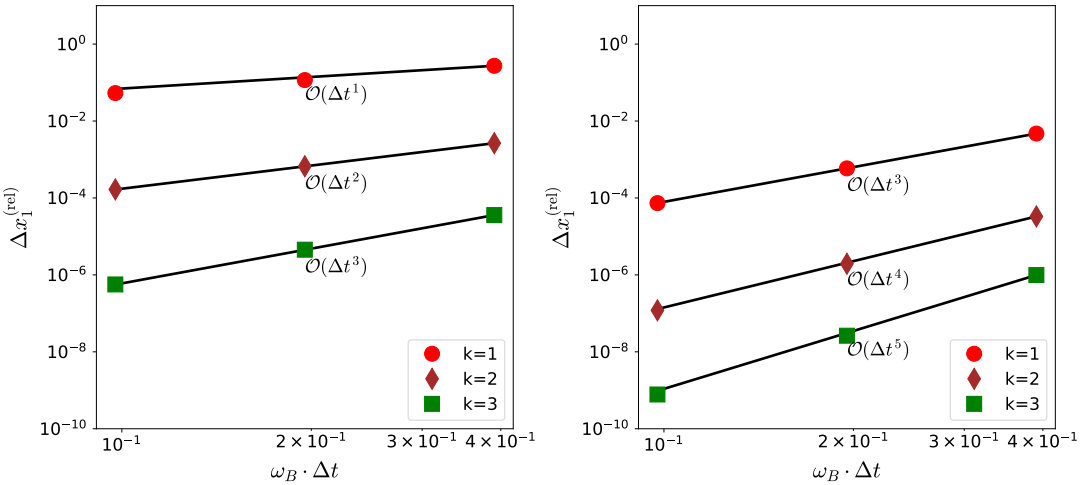

Figure 3.9: Relative global error $\Delta x_1^{(\text{rel})}$ in the first component of the particle's position in the Penning trap versus time step size for 4 Gauss-Legendre (left) and Gauss-Lobatto (right) collocation nodes using one, two, and three SDC iterations.

Moreover, the global convergence order depends on the choice of quadrature. Figure 3.9 shows the global error in the first component particle's position for $M = 4$ quadrature nodes, using Gauss-Legendre nodes (left) and Gauss-Lobatto nodes (right) with $K = 1, 2, 3$ iterations. For Gauss-Legendre nodes, the results align with our theoretical predictions, yielding a convergence order of 1 after a single iteration, as expected. However, with Gauss-Lobatto nodes, the first iteration achieves a convergence order of 3 and then each iteration increases the global order by one. We do not know why the order is 3 in the first iteration, but we assume that this is because the Gauss-Lobatto quadrature nodes include the endpoints of the time steps.

Table 3.3 shows measured convergence rates, rounded to two decimal places, for $M = 2, 3, 4$ nodes and $K = 1, 2, 3$ and $K = 10$ iterations. The theoretically predicted convergence rates from Theorem 3.8 are shown in brackets. The left table presents the error in the x_1-component, while the right table shows the error in the x_3-component. In line with the theoretical predictions, the order increases by one per iteration in the former and by two per iteration in the latter. For $K = 10$ iterations, the order is determined by the underlying quadrature rule and is, therefore, the same in both the first and third components.

Horizontal axis			
K	$M = 2$	$M = 3$	$M = 4$
1	1.28(1)	1.30(1)	1.61(1)
2	1.99(2)	1.99(2)	2.14(2)
3	2.99(3)	2.99(3)	2.98(3)
10	3.99(4)	5.99(6)	7.77(8)

Vertical axis			
K	$M = 2$	$M = 3$	$M = 4$
1	1.99(2)	2.00(2)	1.99(2)
2	4.00(4)	3.99(4)	3.98(4)
3	3.99(4)	5.96(6)	5.97(6)
10	3.99(4)	5.99(6)	7.91(8)

Table 3.3: Measured convergence rate rounded to two digits followed by convergence rate predicted by Theorem 3.7 in brackets for different number of Gauss-Legendre quadrature nodes [1, Table 2.].

3.3.3 Computational efficiency

SDC requires more function evaluation per time step than the Picard iteration or an RKN-4 method. However, for the same Δt, it will produce a smaller error. This allows SDC to achieve computational efficiency comparable to Picard iteration.

Figure 3.10 shows the relative error in the first (left) and third component (right) of the position for the Penning trap benchmark for SDC, Picard iteration and RKN-4 against the total number of f evaluations required. Note that the different iteration numbers for SDC in the left plot ($K = 2,\ 4, 6$) and the right plot ($K = 1,\ 2,\ 3$) are chosen to achieve the same global convergence rates in both cases. Moreover, we use copy initial guess, i.e. $\mathbf{U}^0 = \mathbf{U}_0$. In all cases, SDC is more efficient than Picard using the same number of iterations. The advantage of SDC is more pronounced for the case where the force does depend on velocity. Furthermore, with sufficiently many iterations, the increasing order of SDC allows it to eventually outperform RKN-4. For the error in the third component, $K = 3$ iterations are enough for SDC to become more efficient than RKN-4 while in the first component it requires $K = 4$ iterations.

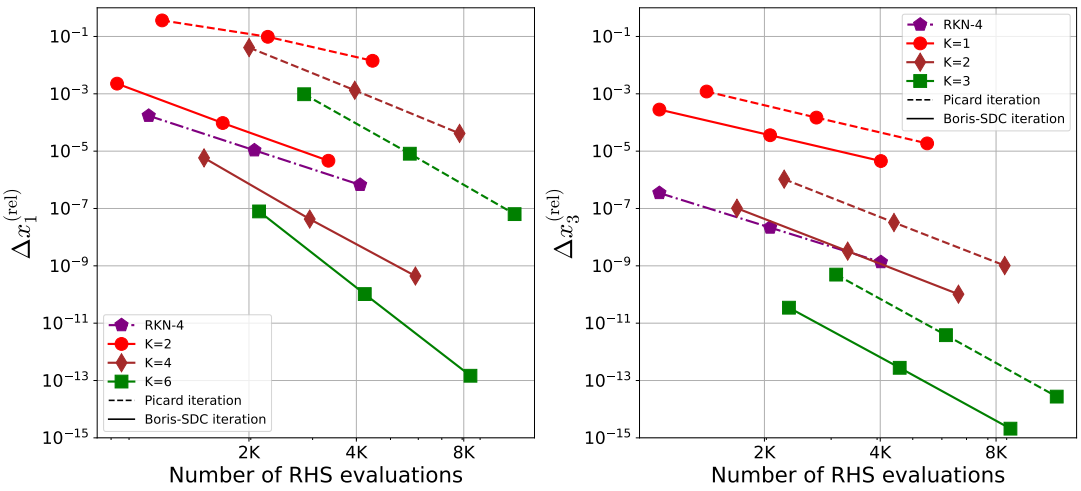

Figure 3.10: Relative position error in the x_1-direction on the left and x_3- direction on the right for SDC (solid lines), Picard (dashed lines) with different iteration numbers K with $M = 5$ quadrature nodes and RKN-4 against the total number of function evaluations. [1, Figure 7].

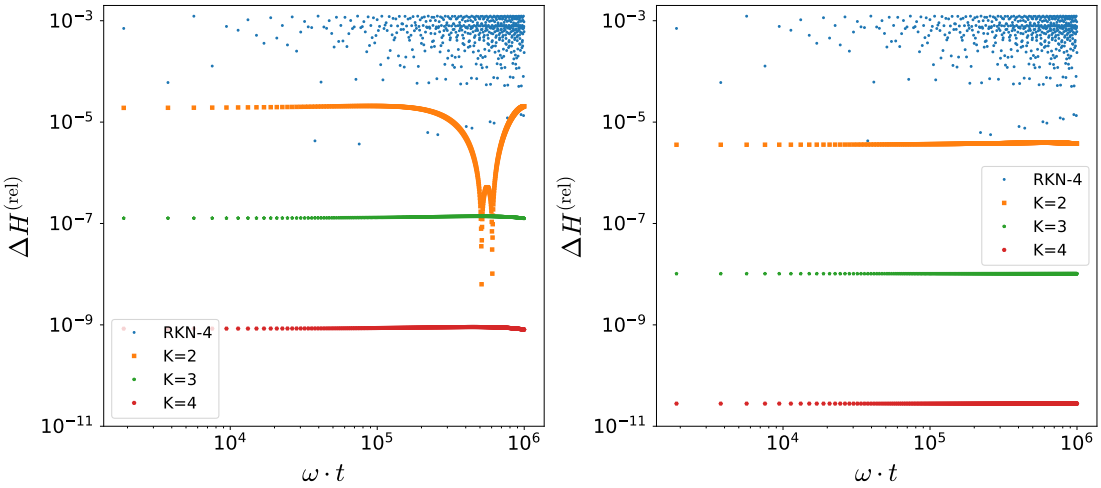

Figure 3.11: Relative error in the discrete Hamiltonian for the undamped harmonic oscillator over 1.5 million time steps for $M = 3$ (left) and $M = 5$ (right) [1, Figure 6].

3.3.4 Conservation properties

Many second-order problems are Hamiltonian systems, where the conservation properties of the time integration are crucial. We consider the undamped harmonic oscillator (3.1) with $\mu = 0.0$ and $\kappa = 1.0$ and a resulting oscillation frequency of $\omega = 1.0$. The continuous Hamiltonian $H(t) = \frac{1}{2}(x(t)^2 + v(t)^2)$ is constant so that $H(t) = H(0)$. Figure 3.11 shows the relative error $(|H_n - H_0|)/H_0$ in the discrete Hamiltonian $H_n = \frac{1}{2}(x_n^2 + v_n^2)$ for a time step of $\Delta t = \frac{2\pi}{10}$, up to $t_{\mathrm{end}} = 1 \times 10^6$, for a total of $1,591,551$ steps for RKN-4 and SDC with $M = 3$ and $M = 5$ Gauss-Legendre quadrature nodes, and $K = 2,\ 3,\ 4$ iterations. Because the collocation method is symplectic, we expect a bounded long-term error for

large K. However, even with $K = 2$, the second-order SDC shows no noticeable drift. Furthermore, the relative error in the Hamiltonian from SDC is smaller than from RKN-4 and decreases by approximately two orders of magnitude with each iteration. This is consistent with previous findings for the Lorentz equations, which also showed low energy errors and reduced numerical drift for SDC, even in very long simulations [3, 31, 61].

Chapter 4

Multi-level SDC for second-order problems

While SDC is a stable, high-order iterative scheme for ODEs, it can require many iterations, especially for stiff problems or when very high accuracy is needed. MLSDC addresses this limitation by incorporating a multilevel strategy, similar to multigrid methods, which performs SDC iterations on a hierarchy of levels and incorporates a FAS correction term [26]. This variation was designed to improve the efficiency of the method by shifting some of the computational work to coarser, less expensive levels [28]. MLSDC was originally introduced for first-order IVPs in [26], and its convergence analysis can be found in [28]; however, it has not yet been extended to second-order problems.

In this chapter, we introduce the multi-level SDC (MLSDC) method for second-order problems. We begin with an overview of linear and nonlinear multigrid techniques [4], followed by a brief explanation of the MLSDC for second order problem inspired by [26]. Finally, we compare the residual and convergence behavior of SDC and MLSDC in the numerical examples section.

The collocation problem can be formulated as a system of algebraic equations (2.10). For simplicity, we represent this system in the following form

$$\mathbf{A}(\mathbf{u}) = \mathbf{b} \tag{4.1}$$

where $\mathbf{A}$, $\mathbf{u}$ and $\mathbf{b}$ correspond to the parameters $\mathbf{M}_{\text{coll}}$, $\mathbf{U}$ and $\mathbf{C}_{\text{coll}}\mathbf{U}_0$ in the collocation problem (2.10). Here, $\mathbf{A}(\cdot)$ can represent either a linear or nonlinear operator. We use $\mathbf{u}$ to denote the exact solution of this system and $\mathbf{v}$ to denote an approximation to the exact solution, potentially generated by iterative solvers such as SDC or Picard iteration.

In this thesis, we focus on a two-level multigrid technique; for extensions to more than two levels, see [4, 26]. The main idea of multigrid is to accelerate the convergence of a basic iterative method (known as *relaxation*) by applying a global correction to the fine grid solution approximation, which is accomplished by solving a coarse problem. The coarse problem, while cheaper to solve, is similar to the fine grid problem. This recursive process is repeated until the desired tolerance on the fine level is reached [62].

4.1 Linear multigrid method

In this section, we introduce the linear multigrid method as described in [4, Chapter 3], assuming that (4.1) represents a system of linear equations

$$\mathbf{A}\mathbf{u} = \mathbf{b}. \tag{4.2}$$

Also, assume that (4.2) has a unique solution, and let $\mathbf{v}$ be an approximation of $\mathbf{u}$. Then, the error is given by

$$\mathbf{e} = \mathbf{u} - \mathbf{v} \tag{4.3}$$

Furthermore, the residual provides a computable measure of how well $\mathbf{v}$ approximates $\mathbf{u}$, given by

$$\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{v} \tag{4.4}$$

The residual is simply the amount by which the approximation $\mathbf{v}$ fails to satisfy the original problem (4.2).

Remark 4.1. By the uniqueness of the solution, $\mathbf{r} = 0$ if and only if $\mathbf{e} = 0$.

From (4.4) and (4.3), we can derive an extremely important relationship between the error and the residual

$$\mathbf{A}\mathbf{e} = \mathbf{r} \tag{4.5}$$

this relationship is called the *residual equation*, and it plays a vital role in multigrid methods.

Remark 4.2. Relaxation on the original equation (4.2) with an arbitrary initial guess $\mathbf{v}_0$ is equivalent to relaxing on the residual equation (4.5) with the specific initial guess $\mathbf{e} = 0$.

We have not yet explained how to use the residual equation to full advantage. Figure 4.1 illustrates the basic idea of a two-level multigrid method with a single iteration. Suppose that an approximation $\mathbf{v}$ is computed by some iterative method on the fine level. Then, the residual can be easily found using (4.4). To increase accuracy, we relax the residual equation for $\mathbf{e}$ on the coarse level with initial guess $\mathbf{e}_0 = 0$ and then update the approximation using the error definition (4.3). We then relax the fine level problem using an iterative method with the new initial condition $\mathbf{v}$. This process can be repeated until we reach desired tolerance on the fine level. It is known as V-cycle.

To reduce the computational cost at the coarse level, one of the option is decreasing the number of grid points on that level. However, we need an operator that transfers values from the fine grid to the coarse grid. This transfer operator is called the *restriction* operator and is denoted by R. For example, assume that the coarse grid $\mathbf{v}^c \in \mathbb{R}^6$ is a subset of the fine grid $\mathbf{v}^f \in \mathbb{R}^{12}$. The simplest restriction is the *injection*. In this approach, each coarse grid node (red) directly takes the value of the corresponding fine grid node (blue), as illustrated in Figure 4.2.

After obtaining the coarse-grid residual values using the restriction operator, we relax the error equation and then return to the fine level. This process, commonly referred to as *interpolation* or *prolongation*, can be performed using various methods. The interpolation operator, denoted by $\mathcal{P}$, converts coarse-grid values into fine-grid values according to the relation $\mathcal{P}\mathbf{v}_c = \mathbf{v}_f$. For instance, consider the case where the coarse grid has twice the grid spacing of the next fine grid. Figure 4.3 illustrates a simple interpolation operator: at

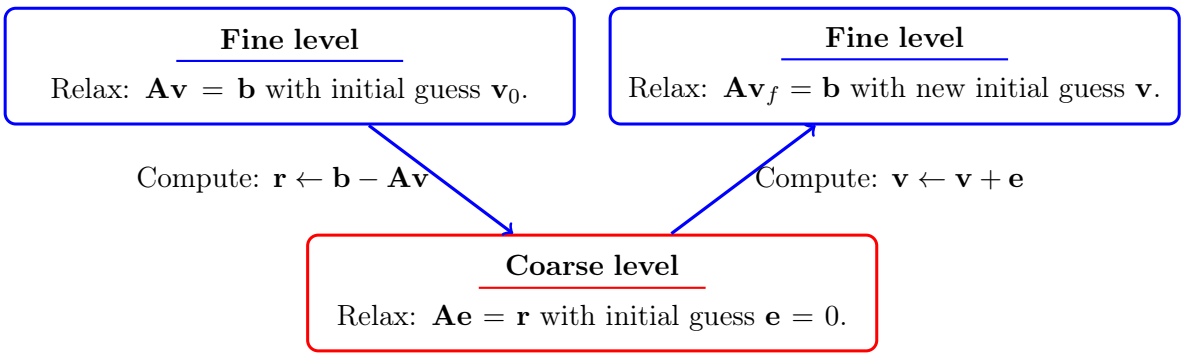

Figure 4.1: A hierarchical structure of interactions between fine and coarse levels representations with a single iteration.

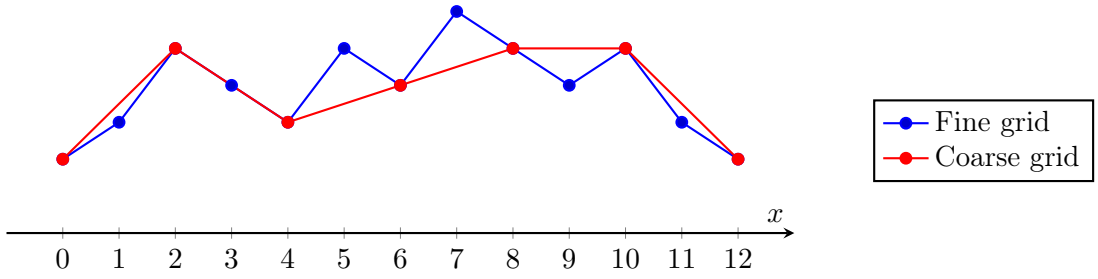

Figure 4.2: Graph illustrating the relationship between fine and coarse grid. The set of coarse grid (red) is subset of fine grid (blue). [4, Figure 3.4]. Source: Generated using [5] and manually changed.

even-numbered (red) fine-grid points, values are directly transferred from the coarse grid, while at odd-numbered (black) fine-grid points, the values $\mathbf{v}_f$ are computed as the averages of the adjacent coarse-grid values.

We now present a step-by-step outline of a two-level multigrid algorithm in Algorithm 3. First, we relax the fine level problem and compute the residual of the current approximation. This residual is then transferred to the coarse level using the restriction operator. Next, we relax the coarse grid error problem with the initial condition $\mathbf{e}_0 = 0$. The resulting approximation is interpolated back to the fine grid, where it corrects the fine-grid approximation. Finally, an additional relaxation is performed on the fine level using the updated initial condition. This process is repeated until we reach desired tolerance on the fine level.

We have written $\mathbf{A}_c\mathbf{e}_c = \mathbf{b}_c$ without precisely defining the coarse-grid operator $\mathbf{A}_c$. Let's consider the residual equation on the fine level

$$\mathbf{A}_f\mathbf{e}_f = \mathbf{r}_f. \tag{4.6}$$

By the definition of interpolation, we have $\mathbf{e}_f = \mathcal{P}\mathbf{e}_c$. Plugged this into (4.6) yields

$$\mathbf{A}_f\mathcal{P}\mathbf{e}_c = \mathbf{r}_f \tag{4.7}$$

Now, we apply the restriction operator R to both sides of (4.7). This gives us the coarse

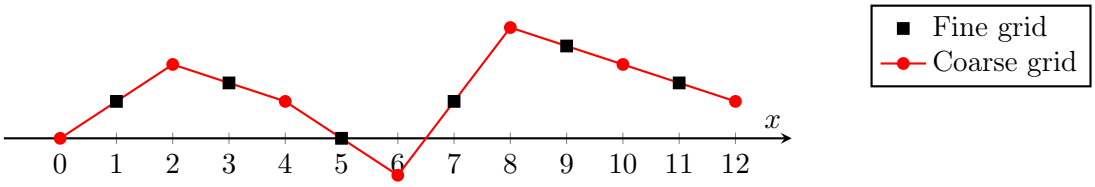

Figure 4.3: Illustration of transferring data from the coarse grid to the fine grid. The red points represent the coarse grid nodes, while the black points show the averages of the adjacent coarse grid values. Together, these red and black points form the set of fine grid nodes. [4, Figure 3.2]. Source: [5] used to generate this figure with minor manual modification.

Algorithm 3: Two-grid correction scheme [4, pp. 37-40]

1. Relax: $\mathbf{A}_f\mathbf{v}_f = \mathbf{b}_f$ with initial guess $\mathbf{v}_0$ by using some iterative solver.
2. Compute: the fine-grid residual $\mathbf{r}_f = \mathbf{b}_f - \mathbf{A}_f\mathbf{v}_f$ and restrict it to the coarse-grid by $\mathbf{r}_c = R\mathbf{r}_f$.
3. Relax: $\mathbf{A}_c\mathbf{e}_c = \mathbf{r}_c$ with initial guess $\mathbf{e}_0 = 0$.
4. Interpolate: the coarse-grid error to the fine-grid by $\mathbf{e}_f = \mathcal{P}\mathbf{e}_c$ and correct the fine-grid approximation by $\mathbf{v}_f \leftarrow \mathbf{v}_f + \mathbf{e}_f$.
5. Relax: $\mathbf{A}_f\mathbf{v}_f = \mathbf{b}_f$ with new initial guess $\mathbf{v}_f$.

level residual equation, which becomes

$$R\mathbf{A}_f\mathcal{P}\mathbf{e}_c = R\mathbf{r}_f$$

This observation provides a definition for the coarse-grid operator which is known as *Galerkin condition*

$$\mathbf{A}_c = R\mathbf{A}_f\mathcal{P}. \tag{4.8}$$

The Galerkin operator can be derived directly from the fine level matrices without referencing the coarse grid explicitly. It is particularly convenient when the fine and coarse operators are linear. However, if they are nonlinear, the Galerkin approach usually fails. An alternative is *rediscretization*, where the original discretization scheme is reapplied on the coarse grid problem. This often results in a coarse operator that better represents the underlying continuous problem [62]. We will discuss this approach in detail for the MLSDC method in Paragraph 4.3.3.

4.2 Nonlinear multigrid method

In the previous section, we have discussed multigrid methods for the linear problems. Here, we examine their extension to the nonlinear problems, following [4, Chapter 6].

Now, consider that (4.1) represents a system of nonlinear algebraic equations, meaning that $\mathbf{A}$ is a nonlinear operator. As in the linear case, let $\mathbf{v}$ be an approximation to the exact solution $\mathbf{u}$, and we define the error and residual for this problem, analogous to (4.3) and (4.4). By subtracting the original equation (4.1), from the definition of the residual, we find that

$$\mathbf{A}(\mathbf{u}) - \mathbf{A}(\mathbf{v}) = \mathbf{r}. \tag{4.9}$$

However, because $\mathbf{A}(\cdot)$ is a nonlinear operator, even though $\mathbf{u}-\mathbf{v} = \mathbf{e}$, we cannot conclude that

$$\mathbf{A}(\mathbf{u}) - \mathbf{A}(\mathbf{v}) = \mathbf{A}(\mathbf{e}). \tag{4.10}$$

This means we no longer have a simple linear residual equation, requiring modifications to handle nonlinear problems effectively.

To understand how residual equation (4.9) can serve as a basis for multigrid methods, let $\mathbf{v}$ be the current approximation and replace the exact solution $\mathbf{u}$ by $\mathbf{v} + \mathbf{e}$. Then, residual equation (4.9) becomes

$$\mathbf{A}(\mathbf{v} + \mathbf{e}) - \mathbf{A}(\mathbf{v}) = \mathbf{r}. \tag{4.11}$$

Suppose we have found an approximate solution, $\mathbf{v}_f$, to the original fine-grid problem

$$\mathbf{A}_f(\mathbf{u}_f) = \mathbf{b}_f \tag{4.12}$$

Following the approach used in the linear problem, we now aim to use the residual equation on the coarse level. Based on the previous argument (4.11), the residual equation on the coarse level is given by

$$\mathbf{A}_c(\mathbf{v}_c + \mathbf{e}_c) - \mathbf{A}_c(\mathbf{v}_c) = \mathbf{r}_c \tag{4.13}$$

where $\mathbf{A}_c$ denotes the coarse-grid operator, $\mathbf{r}_c$ is a coarse-grid residual, $\mathbf{e}_c$ is a coarse-grid error and $\mathbf{v}_c$ is a coarse-grid approximation. $\mathbf{v}_c$ can be obtained by restricting fine grid approximate solution $R\mathbf{v}_f$. Once $\mathbf{e}_c$ is computed using iterative methods, the fine-grid approximation can be update by $\mathbf{v}_f \leftarrow \mathbf{v}_f + \mathcal{P}\mathbf{e}_c$.

The coarse-grid residual $\mathbf{r}_c$ in (4.13) can be computed by restricting the fine-grid residual onto the coarse grid, resulting in

$$\mathbf{r}_c = R\mathbf{r}_f = R(\mathbf{b}_f - \mathbf{A}_f(\mathbf{v}_f)). \tag{4.14}$$

For the current approximation $\mathbf{v}_c$, we can use the same restriction operator $\mathbf{v}_c = R\mathbf{v}_f$. Substituting these into the residual equation (4.13) gives

$$\mathbf{A}_c(R\mathbf{v}_f + \mathbf{e}_c) = \mathbf{A}_c(R\mathbf{v}_f) + R(\mathbf{b}_f - \mathbf{A}_f(\mathbf{v}_f)). \tag{4.15}$$

The right side of this equation (4.15) is known. Our goal is to find or approximate the solution to the system (4.15), which we denote by $\mathbf{u}_c := R\mathbf{v}_f + \mathbf{e}_c$. The coarse-grid approximation, $\mathbf{e}_c = \mathbf{u}_c - R\mathbf{v}_f$, can then be interpolated back to the fine grid and used to correct the fine-grid approximation $\mathbf{v}_f$. The correction step takes the form

$$\mathbf{v}_f \leftarrow \mathbf{v}_f + \mathcal{P}\mathbf{e}_c \quad \text{or} \quad \mathbf{v}_f \leftarrow \mathbf{v}_f + \mathcal{P}(\mathbf{u}_c - R\mathbf{v}_f).$$

The scheme we have just outlined is a commonly used nonlinear version of multigrid, known as the *full approximation scheme* (FAS) because the coarse-grid problem is solved

for the full approximation, $\mathbf{u}_c = R\mathbf{v}_f + \mathbf{e}_c$, rather than solely for the error $\mathbf{e}_c$. A two-grid version of this scheme is described in Algorithm 4.

Algorithm 4: Full Approximation Scheme (FAS) [4, pp. 98-99]

1. Relax $\mathbf{A}_f(\mathbf{v}_f) = \mathbf{b}_f$ with initial condition $\mathbf{v}_0$
2. Restrict the fine grid approximation and its residual: $\mathbf{r}_c = R(\mathbf{b}_f - \mathbf{A}_f(\mathbf{v}_f))$ and $\mathbf{v}_c = R\mathbf{v}_f$
3. Relax the coarse grid problem: $\mathbf{A}_c(\mathbf{u}_c) = \mathbf{A}_c(\mathbf{v}_c) + \mathbf{r}_c$
4. Compute the coarse grid approximation: $\mathbf{e}_c = \mathbf{u}_c - \mathbf{v}_c$
5. Interpolate the error approximation to the fine grid and correct the current fine grid: $\mathbf{v}_f \leftarrow \mathbf{v}_f + \mathcal{P}\mathbf{e}_c$
6. Relax $\mathbf{A}_f\mathbf{v}_f = \mathbf{b}_f$ with new initial guess $\mathbf{v}_f$.

Remark 4.3. If $\mathbf{A}(\cdot)$ is a linear operator, then the FAS scheme reduces directly to the linear two-grid correction scheme.

Remark 4.4. An exact solution to the fine-grid problem (4.12) is a fixed point of the FAS iteration.

Now, suppose that our restriction operator is linear, the equation (4.15) becomes

$$\mathbf{A}_c(\mathbf{u}_c) = \mathbf{A}_c(R\mathbf{v}_f) + R\mathbf{b}_f - R\mathbf{A}_f(\mathbf{v}_f)).$$

Then, the FAS coarse-grid equation can be written as

$$\mathbf{A}_c(\mathbf{u}_c) = \mathbf{b}_c + \boldsymbol{\tau}_f^c \tag{4.16}$$

where $\mathbf{b}_c = R\mathbf{b}_f$ and the *tau correction* $\boldsymbol{\tau}_f^c$ is defined by

$$\boldsymbol{\tau}_f^c = \mathbf{A}_c(R\mathbf{v}_f) - R\mathbf{A}_f(\mathbf{v}_f). \tag{4.17}$$

It is important to note that the solution of the coarse-grid FAS equation, $\mathbf{u}_c$, is not the same as the solution of the original coarse-grid equation, $\mathbf{A}_c(\mathbf{u}_c) = \mathbf{b}_c$ because $\boldsymbol{\tau}_f^c \neq 0$ in general. As the FAS processing advances, $\mathbf{u}_c$ converges to the restriction of the fine solution $R\mathbf{u}_f$ [4].

Algorithm 5 presents the FAS algorithm using tau correction (4.17). When comparing it to Algorithm 4, we can see that the main differences occur in the second and third stages. Instead of computing residual, we calculate the tau correction and then solve the FAS-corrected equation on the coarse level. If our restriction operator is nonlinear, then we have to apply Algorithm 4 to our problem.

Algorithm 5: FAS with tau correction

1. Relax $\mathbf{A}_f(\mathbf{v}_f) = \mathbf{b}_f$ with initial condition $\mathbf{v}_0$
2. Restrict the fine grid approximation $\mathbf{v}_c = R\mathbf{v}_f$ and find tau correction $\boldsymbol{\tau}_f^c = \mathbf{A}_c(\mathbf{v}_c) - R\mathbf{A}_f(\mathbf{v}_f)$
3. Relax the coarse grid problem $\mathbf{A}_c(\mathbf{u}_c) = \mathbf{b}_c + \boldsymbol{\tau}_f^c$ with initial guess $\mathbf{v}_c$
4. Compute the coarse-grid approximation error: $\mathbf{e}_c = \mathbf{u}_c - \mathbf{v}_c$
5. Interpolate the error approximation to the fine grid and correct the current fine grid approximation: $\mathbf{v}_f \leftarrow \mathbf{v}_f + \mathcal{P}\mathbf{e}_c$
6. Relax $\mathbf{A}_f\mathbf{v}_f = \mathbf{b}_f$ with new initial guess $\mathbf{v}_f$.

4.3 Multi-level Spectral deferred correction method

We now introduce multi-level SDC (MLSDC) for second-order problems, building on the work in [26]. MLSDC is a method inspired by a multigrid approach to solve the collocation problem described in (2.10). It is an extension of SDC in which the SDC sweeps are computed on a hierarchy of levels and the individual solutions are coupled using an FAS correction term stemming from nonlinear multigrid methods. In this section, we present the details of a two-level MLSDC iteration and explore two different coarsening strategies. The method can be extended to more than two levels, with additional details available in [26].

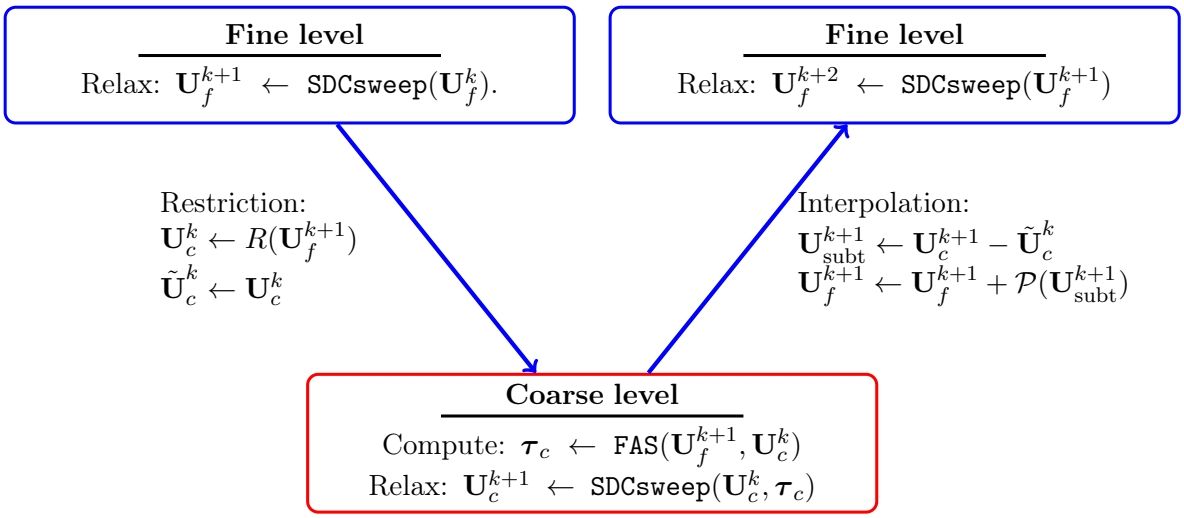

Figure 4.4: A single V-cycle iteration of MLSDC with one sweep per level is used.

The diagram in Figure 4.4 illustrates the key steps of a single MLSDC iteration. It uses a V-cycle with one SDC sweep per level. First, a collocation problem is relaxed on the fine level by applying SDC sweep to it. The fine level solution is then restricted to the coarse level. Next, we compute the coarse-level tau correction term, denoted as $\boldsymbol{\tau}_c \leftarrow$

FAS($\mathbf{U}_f^{k+1}, \mathbf{U}_c^k$). The coarse level approximate solution is then obtained using another SDC sweep by solving FAS-corrected equation, resulting in improved solution $\mathbf{U}_c^{k+1}$. Once the coarse-level computations are complete, the difference $\mathbf{U}_c^{k+1} - \tilde{\mathbf{U}}_c^k$ is computed and interpolated back to the fine grid. The interpolated values are added to the old fine level solution $\mathbf{U}_f^{k+1}$. In the final stage, an additional SDC sweep is performed on the fine level to compute the next iteration of the solution, $\mathbf{U}_f^{k+2}$. This ensures that the solution is refined to the desired level of accuracy.

4.3.1 FAS correction for MLSDC

MLSDC employs the FAS scheme introduced in Section 4.2 to solve the collocation problem (2.9). Thus, we first derive the tau correction for the collocation problem as described in [26], and then present the MLSDC algorithm.

Define the following operator for the fine level in the collocation operator form given by (2.9)

$$\mathbf{M}_{\text{coll},f}(\mathbf{U}_f) := \mathbf{U}_f - \Delta t\, \mathbf{Q}_{\text{coll},f}\, \mathbf{F}_f(\mathbf{U}_f), \tag{4.18}$$

where the subscript f indicates the fine-level elements.

Assume that the coarse-level operator is also provided

$$\mathbf{M}_{\text{coll},c}(\mathbf{U}_c) := \mathbf{U}_c - \Delta t\, \mathbf{Q}_{\text{coll},c}\, \mathbf{F}_c(\mathbf{U}_c), \tag{4.19}$$

where the subscript c denotes coarse-level variables. Since we are solving the same problem at both the fine and coarse levels in MLSDC, the right-hand side function evaluation, $\mathbf{F}(\cdot)$, remains unchanged. Furthermore, we assume that suitable transfer operators are available: a linear restriction operator, denoted by R, and an interpolation operator, $\mathcal{P}$. We will discuss these transfer operators in Section 4.3.3.

With this setup, we have sufficient information to apply the tau correction formula (4.17)

$$\boldsymbol{\tau}_f^c = \mathbf{M}_{\text{coll},c}(R\mathbf{U}_f) - R\mathbf{M}_{\text{coll},f}(\mathbf{U}_f)$$

Now, using the operators (4.18) and (4.19) to the above equation yields

$$\begin{aligned}\boldsymbol{\tau}_f^c = R\mathbf{U}_f - \Delta t\mathbf{Q}_{\text{coll},c}\mathbf{F}_c(R\mathbf{U}_f) - R\mathbf{U}_f + \Delta t R\mathbf{Q}_{\text{coll},f}\mathbf{F}_f(\mathbf{U}_f)\\ = \Delta t(R\mathbf{Q}_{\text{coll},f}\mathbf{F}_f(\mathbf{U}_f) - \mathbf{Q}_{\text{coll},c}\mathbf{F}_c(R\mathbf{U}_f))\end{aligned} \tag{4.20}$$

Thus, the tau correction for MLSDC iteration is given by

$$\boldsymbol{\tau}_f^c = \Delta t(R\mathbf{Q}_{\text{coll},f}\mathbf{F}_f(\mathbf{U}_f) - \mathbf{Q}_{\text{coll},c}\mathbf{F}_c(R\mathbf{U}_f)) \tag{4.21}$$

Then, FAS-corrected coarse problem becomes

$$\mathbf{U}_c - \Delta t\mathbf{Q}_{\text{coll},c}\mathbf{F}_c(\mathbf{U}_c) - \boldsymbol{\tau}_f^c = \mathbf{C}_{\text{coll},c}\mathbf{U}_{0,c} \tag{4.22}$$

So, we can relax (4.22) on the coarse level using SDC iteration and then interpolate its solution back to the fine level.

Remark 4.5. If the fine residual is zero (i.e., $\mathbf{U}_f \equiv \mathbf{C}_{\text{coll},f}\mathbf{U}_{0,f} + \Delta t\mathbf{Q}_{\text{coll},f}\mathbf{F}_f(\mathbf{U}_f)$), the FAS-corrected equation turns into

$$\begin{aligned}\mathbf{U}_c - \Delta t\mathbf{Q}_{\text{coll},c}\mathbf{F}_c(\mathbf{U}_c) = R\mathbf{C}_{\text{coll},f}\mathbf{U}_{0,f} + \Delta t\left(R\mathbf{Q}_{\text{coll},f}\mathbf{F}_f(\mathbf{U}_f) - \mathbf{Q}_{\text{coll},c}\mathbf{F}_c(R\mathbf{U}_f)\right)\\ = R\mathbf{U}_f - \Delta t\mathbf{Q}_{\text{coll},c}\mathbf{F}_c(R\mathbf{U}_f) \quad (4.23)\end{aligned}$$

so that the coarse solution is the restriction of the fine solution.

Algorithm 6: A Single MLSDC iteration [26]

1. Perform fine SDC sweep using the values $\mathbf{U}_f^k$. This will yield provisional updated values $\mathbf{U}_f^{k+1}$.
2. Cycle from fine to coarse:
   - Restrict the fine values $\mathbf{U}_f^{k+1}$ to the coarse values $\mathbf{U}_c^k$.
   - Compute the tau correction $\boldsymbol{\tau}_f^c$ using $\mathbf{U}_f^{k+1}$ and $\mathbf{U}_c^k$.
   - Perform SDC sweep with the values on the coarse level with $\mathbf{U}_c^k$ and the tau correction $\boldsymbol{\tau}_f^c$. This yields the new values $\mathbf{U}_c^{k+1}$.
3. Cycle from coarse to fine:
   - Interpolate coarse grid correction $\mathbf{U}_c^{k+1} - \mathbf{U}_c^k$ and add to the old $\mathbf{U}_f^{k+1}$. This yield updated values $\mathbf{U}_f^{k+1}$.
4. Perform SDC sweeps on the fine level with initial condition $\mathbf{U}_f^{k+1}$.

Algorithm 6 shows the step-by-step procedure for a single iteration of the MLSDC algorithm. The initial condition $\mathbf{U}_0$ and its function evaluation are spread to each of the collocation nodes on the fine level, so that the first provisional solution is $\mathbf{U}_f^0 = \mathbf{U}_0$.

Remark 4.6. We can apply the SDC iteration on the coarse level multiple times, for the sake of comparison with the standard SDC iteration, we apply only one SDC sweep per level.

4.3.2 Coarsening strategies

In MLSDC, the multilevel hierarchy is applied exclusively along the time dimension, while the spatial grid remains unchanged. In contrast, multigrid methods use spatial coarsening, which involves reducing the resolution of the spatial mesh (e.g., in 2D or 3D) to eliminate low-frequency errors in the physical domain. MLSDC accelerates time integration by using a hierarchy of time discretizations, that is, using different numbers of collocation points without modifying the spatial discretization. The primary goal of MLSDC is to reduce the overall computational cost by performing SDC sweeps on coarser time levels. In this section, we describe two specific coarsening strategies in the time domain.

1. **Fewer collocation nodes on the coarse level**: This approach decreases the number of quadrature nodes on the coarse level which directly translates into significant computational savings when evaluating f. This method requires special interpolation and restriction operators to transfer the solution between levels.

2. **Asymptotic models as the coarse problem**: This strategy involves solving asymptotic models on the coarse level instead of the original problem. These models can usually be solved analytically, making them less expensive to evaluate than the original problem. The derivation of these asymptotic models is explained in Chapter 5. Furthermore, we apply these models to the MLSDC method, as discussed in Chapter 6. For an application of this strategy in the time parareal method, see [8, 63].

We will discuss these strategies for the rest of the dissertation

4.3.3 Transfer operators

In this section, we want to focus on Strategy 1, which aims to create an operator that transfers information when we modify the number of quadrature nodes between the levels. One of the suitable approaches is to use the barycentric Lagrange interpolation method, as published in [64]. We will now briefly explain barycentric Lagrange interpolation.

Barycentric Lagrange Interpolation. Let $\prod_M$ denote the vector space of all polynomials of degree at most M. The classical problem addressed here is that of finding the polynomial $p \in \prod_M$ that interpolates a function f at the points τ_j, i.e.,

$$p(\tau_j) = f_j, \quad j = 0, \ldots, M.$$

The problem is well-posed; i.e., it has a unique solution that depends continuously on the data. Moreover, the solution can be written in Lagrange form [65, Chapter 3.]

$$p(\tau) = \sum_{j=0}^{M} f_j \ell_j(\tau), \quad \ell_j(\tau) = \frac{\prod_{k=0,k\neq j}^{M}(\tau - \tau_k)}{\prod_{k=0,k\neq j}^{M}(\tau_j - \tau_k)}. \tag{4.24}$$

The Lagrange polynomial ℓ_j corresponding to the node τ_j has the property

$$\ell_j(\tau_k) = \begin{cases} 1, & j = k, \\ 0, \text{otherwise}, \end{cases} \qquad j, k = 0, \ldots, M. \tag{4.25}$$

Assume that the numerator of ℓ_j in (4.24) can be written as the equality

$$\ell(\tau) = (\tau - \tau_0)(\tau - \tau_1)\ldots(\tau - \tau_M) \tag{4.26}$$

divided by $\tau - \tau_j$. If we define the barycentric weights by

$$w_j = \frac{1}{\prod_{k\neq j}(\tau_j - \tau_k)}, \quad j = 0, \ldots, M, \tag{4.27}$$

that is, $w_j = 1/\ell'(\tau_j)$ [66, p. 243], we can thus write ℓ_j as

$$\ell_j(\tau) = \ell(x)\frac{w_j}{\tau - \tau_j}.$$

Note that all the terms of the sum in (4.24) contain the factor $\ell(\tau)$, which does not depend on j. This factor may therefore be brought in the front of the sum to yield

$$p(\tau) = \ell(\tau)\sum_{j=0}^{M}\frac{w_j}{\tau - \tau_j} f_j. \tag{4.28}$$

This equation (4.28) can be modified to an even more elegant formula, the one that is often used in practice.

Suppose that we interpolate, besides the data f_j, the constant function 1, whose interpolant is of course itself. Inserting into (4.28), we obtain

$$1 = \sum_{j=0}^{M} \ell_j(\tau) = \ell(\tau) \sum_{j=0}^{M} \frac{w_j}{\tau - \tau_j}.$$

Dividing (4.28) by this expression and canceling the common factor $\ell(\tau)$, we get the barycentric formula for p

$$p(\tau) = \frac{\sum_{j=0}^{M} \frac{w_j}{\tau-\tau_j} f_j}{\sum_{j=0}^{M} \frac{w_j}{\tau-\tau_j}} \tag{4.29}$$

where w_j is still defined by (4.27).

The weights w_j appear in the denominator exactly as in the numerator, except without the data factors f_j. This means that any common factor in all the weights w_j may be canceled without affecting the value of p_j. Unlike traditional Lagrange interpolation, which requires recalculating the entire Lagrange basis polynomial for each new evaluation, barycentric interpolation uses the weighted form that allows precomputation of the weights. This significantly reduces the computational effort when evaluating the interpolating polynomial at new points.

Restriction and Interpolation operators. Now, we use the barycentric formula (4.29) to obtain restriction and interpolation operators. Let's first find the restriction operator R. First, define $\tau_i^{(f)},\ i = 0, \ldots, M_f$ and $\tau_i^{(c)},\ i = 0, \ldots, M_c$ a quadrature nodes on the fine and coarse level respectively such that $M_f \geq M_c$. We want to find restriction operator R that transfers the fine values $\mathbf{u}_j^{(f)},\ j = 0, \ldots, M_f$ to the coarse one $\mathbf{u}_j^{(c)},\ j = 0, \ldots, M_c$. We can find quadrature weights using (4.27) on the fine level. This yields

$$w_j^{(f)} = \frac{1}{\prod_{k \neq j} \left(\tau_j^{(f)} - \tau_k^{(f)}\right)}, \quad j = 0, \ldots M_f.$$

Now, we use (4.29) to get coarse values

$$\begin{aligned}
\mathbf{u}_0^{(c)} &= \frac{\sum_{j=0}^{M_f} \frac{w_j^{(f)}}{\tau_0^{(c)} - \tau_j^{(f)}} \mathbf{u}_j^{(f)}}{\sum_{j=0}^{M_f} \frac{w_j^{(f)}}{\tau_0^{(c)} - \tau_j^{(f)}}} \\
\mathbf{u}_1^{(c)} &= \frac{\sum_{j=0}^{M_f} \frac{w_j^{(f)}}{\tau_1^{(c)} - \tau_j^{(f)}} \mathbf{u}_j^{(f)}}{\sum_{j=0}^{M_f} \frac{w_j^{(f)}}{\tau_1^{(c)} - \tau_j^{(f)}}} \\
&\vdots \\
\mathbf{u}_{M_c}^{(c)} &= \frac{\sum_{j=0}^{M_f} \frac{w_j^{(f)}}{\tau_{M_c}^{(c)} - \tau_j^{(f)}} \mathbf{u}_j^{(f)}}{\sum_{j=0}^{M_f} \frac{w_j^{(f)}}{\tau_{M_c}^{(c)} - \tau_j^{(f)}}}
\end{aligned} \tag{4.30}$$

We can rewrite this in the matrix form

$$\begin{pmatrix} \mathbf{u}_0^{(c)} \\ \mathbf{u}_1^{(c)} \\ \vdots \\ \mathbf{u}_{M_c}^{(c)} \end{pmatrix} = \begin{pmatrix} \frac{\frac{w_0^{(f)}}{\tau_0^{(c)}-\tau_0^{(f)}}}{\sum_{j=0}^{M_f}\frac{w_j^{(f)}}{\tau_0^{(c)}-\tau_j^{(f)}}} & \frac{\frac{w_1^{(f)}}{\tau_0^{(c)}-\tau_1^{(f)}}}{\sum_{j=0}^{M_f}\frac{w_j^{(f)}}{\tau_0^{(c)}-\tau_j^{(f)}}} & \cdots & \frac{\frac{w_{M_f}^{(f)}}{\tau_0^{(c)}-\tau_{M_f}^{(f)}}}{\sum_{j=0}^{M_f}\frac{w_j^{(f)}}{\tau_0^{(c)}-\tau_j^{(f)}}} \\ \frac{\frac{w_0^{(f)}}{\tau_1^{(c)}-\tau_0^{(f)}}}{\sum_{j=0}^{M_f}\frac{w_j^{(f)}}{\tau_1^{(c)}-\tau_j^{(f)}}} & \frac{\frac{w_1^{(f)}}{\tau_1^{(c)}-\tau_1^{(f)}}}{\sum_{j=0}^{M_f}\frac{w_j^{(f)}}{\tau_1^{(c)}-\tau_j^{(f)}}} & \cdots & \frac{\frac{w_{M_f}^{(f)}}{\tau_1^{(c)}-\tau_{M_f}^{(f)}}}{\sum_{j=0}^{M_f}\frac{w_j^{(f)}}{\tau_1^{(c)}-\tau_j^{(f)}}} \\ \vdots & \ddots & \cdots & \vdots \\ \frac{\frac{w_0^{(f)}}{\tau_{M_c}^{(c)}-\tau_0^{(f)}}}{\sum_{j=0}^{M_f}\frac{w_j^{(f)}}{\tau_{M_c}^{(c)}-\tau_j^{(f)}}} & \frac{\frac{w_1^{(f)}}{\tau_{M_c}^{(c)}-\tau_1^{(f)}}}{\sum_{j=0}^{M_f}\frac{w_j^{(f)}}{\tau_{M_c}^{(c)}-\tau_j^{(f)}}} & \cdots & \frac{\frac{w_{M_f}^{(f)}}{\tau_{M_c}^{(c)}-\tau_{M_f}^{(f)}}}{\sum_{j=0}^{M_f}\frac{w_j^{(f)}}{\tau_{M_c}^{(c)}-\tau_j^{(f)}}} \end{pmatrix} \begin{pmatrix} \mathbf{u}_0^{(f)} \\ \mathbf{u}_1^{(f)} \\ \vdots \\ \mathbf{u}_{M_f}^{(f)} \end{pmatrix} \tag{4.31}$$

So, our linear transfer operator is given by the following matrix form

$$R := \begin{pmatrix} \frac{\frac{w_0^{(f)}}{\tau_0^{(c)}-\tau_0^{(f)}}}{\sum_{j=0}^{M_f}\frac{w_j^{(f)}}{\tau_0^{(c)}-\tau_j^{(f)}}} & \frac{\frac{w_1^{(f)}}{\tau_0^{(c)}-\tau_1^{(f)}}}{\sum_{j=0}^{M_f}\frac{w_j^{(f)}}{\tau_0^{(c)}-\tau_j^{(f)}}} & \cdots & \frac{\frac{w_{M_f}^{(f)}}{\tau_0^{(c)}-\tau_{M_f}^{(f)}}}{\sum_{j=0}^{M_f}\frac{w_j^{(f)}}{\tau_0^{(c)}-\tau_j^{(f)}}} \\ \frac{\frac{w_0^{(f)}}{\tau_1^{(c)}-\tau_0^{(f)}}}{\sum_{j=0}^{M_f}\frac{w_j^{(f)}}{\tau_1^{(c)}-\tau_j^{(f)}}} & \frac{\frac{w_1^{(f)}}{\tau_1^{(c)}-\tau_1^{(f)}}}{\sum_{j=0}^{M_f}\frac{w_j^{(f)}}{\tau_1^{(c)}-\tau_j^{(f)}}} & \cdots & \frac{\frac{w_{M_f}^{(f)}}{\tau_1^{(c)}-\tau_{M_f}^{(f)}}}{\sum_{j=0}^{M_f}\frac{w_j^{(f)}}{\tau_1^{(c)}-\tau_j^{(f)}}} \\ \vdots & \ddots & \cdots & \vdots \\ \frac{\frac{w_0^{(f)}}{\tau_{M_c}^{(c)}-\tau_0^{(f)}}}{\sum_{j=0}^{M_f}\frac{w_j^{(f)}}{\tau_{M_c}^{(c)}-\tau_j^{(f)}}} & \frac{\frac{w_1^{(f)}}{\tau_{M_c}^{(c)}-\tau_1^{(f)}}}{\sum_{j=0}^{M_f}\frac{w_j^{(f)}}{\tau_{M_c}^{(c)}-\tau_j^{(f)}}} & \cdots & \frac{\frac{w_{M_f}^{(f)}}{\tau_{M_c}^{(c)}-\tau_{M_f}^{(f)}}}{\sum_{j=0}^{M_f}\frac{w_j^{(f)}}{\tau_{M_c}^{(c)}-\tau_j^{(f)}}} \end{pmatrix} \in \mathbb{R}^{M_c \times M_f} \tag{4.32}$$

To find the interpolation operator from coarse to fine, we can define weights on the coarse level using (4.27)

$$w_j^{(c)} = \frac{1}{\prod_{k \neq j}\left(\tau_j^{(c)} - \tau_k^{(c)}\right)}, \quad j = 0, \dots M_c.$$

And using the equation (4.29) to obtain

$$\begin{aligned}
\mathbf{u}_0^{(f)} &= \frac{\sum_{j=0}^{M_c} \frac{w_j^{(c)}}{\tau_0^{(f)}-\tau_j^{(c)}} \mathbf{u}_j^{(c)}}{\sum_{j=0}^{M_c} \frac{w_j^{(c)}}{\tau_0^{(f)}-\tau_j^{(c)}}} \\
\mathbf{u}_1^{(f)} &= \frac{\sum_{j=0}^{M_c} \frac{w_j^{(c)}}{\tau_1^{(f)}-\tau_j^{(c)}} \mathbf{u}_j^{(c)}}{\sum_{j=0}^{M_c} \frac{w_j^{(c)}}{\tau_1^{(f)}-\tau_j^{(c)}}} \\
&\vdots \\
\mathbf{u}_{M_c}^{(f)} &= \frac{\sum_{j=0}^{M_c} \frac{w_j^{(c)}}{\tau_{M_f}^{(f)}-\tau_j^{(c)}} \mathbf{u}_j^{(c)}}{\sum_{j=0}^{M_c} \frac{w_j^{(c)}}{\tau_{M_f}^{(f)}-\tau_j^{(c)}}}
\end{aligned} \tag{4.33}$$

Similar to the restriction operator, we can obtain an interpolation matrix

$$\mathcal{P} := \begin{pmatrix}
\frac{\frac{w_0^{(c)}}{\tau_0^{(f)}-\tau_0^{(c)}}}{\sum_{j=0}^{M_c} \frac{w_j^{(c)}}{\tau_0^{(f)}-\tau_j^{(c)}}} & \frac{\frac{w_1^{(c)}}{\tau_0^{(f)}-\tau_1^{(c)}}}{\sum_{j=0}^{M_c} \frac{w_j^{(c)}}{\tau_0^{(f)}-\tau_j^{(c)}}} & \cdots & \frac{\frac{w_{M_c}^{(c)}}{\tau_0^{(f)}-\tau_{M_c}^{(c)}}}{\sum_{j=0}^{M_c} \frac{w_j^{(c)}}{\tau_0^{(f)}-\tau_j^{(c)}}} \\
\frac{\frac{w_0^{(c)}}{\tau_1^{(f)}-\tau_0^{(c)}}}{\sum_{j=0}^{M_c} \frac{w_j^{(c)}}{\tau_1^{(f)}-\tau_j^{(c)}}} & \frac{\frac{w_1^{(c)}}{\tau_1^{(f)}-\tau_1^{(c)}}}{\sum_{j=0}^{M_c} \frac{w_j^{(c)}}{\tau_1^{(f)}-\tau_j^{(c)}}} & \cdots & \frac{\frac{w_{M_c}^{(c)}}{\tau_1^{(f)}-\tau_{M_c}^{(c)}}}{\sum_{j=0}^{M_c} \frac{w_j^{(c)}}{\tau_1^{(f)}-\tau_j^{(c)}}} \\
\vdots & \ddots & \cdots & \vdots \\
\frac{\frac{w_0^{(c)}}{\tau_{M_f}^{(f)}-\tau_0^{(c)}}}{\sum_{j=0}^{M_c} \frac{w_j^{(c)}}{\tau_{M_f}^{(f)}-\tau_j^{(c)}}} & \frac{\frac{w_1^{(c)}}{\tau_{M_f}^{(f)}-\tau_1^{(c)}}}{\sum_{j=0}^{M_c} \frac{w_j^{(c)}}{\tau_{M_f}^{(f)}-\tau_j^{(c)}}} & \cdots & \frac{\frac{w_{M_c}^{(c)}}{\tau_{M_f}^{(f)}-\tau_{M_c}^{(c)}}}{\sum_{j=0}^{M_c} \frac{w_j^{(c)}}{\tau_{M_f}^{(f)}-\tau_j^{(c)}}}
\end{pmatrix} \in \mathbb{R}^{M_f \times M_c} \tag{4.34}$$

So, our restriction and interpolation matrices are defined in (4.32) and (4.34) using the Barycentric Lagrange interpolation operator.

Rediscretization for Coarse-Grid Operators. In the linear case the Galerkin condition (4.8) is used to define the coarse-grid operator as

$$\mathbf{M}_{\text{coll},c}(\mathbf{U}_c) = R_f^c \, \mathbf{M}_{\text{coll},f}(I_c^f \, \mathbf{U}_c) = R\left(\mathcal{P}\mathbf{U}_c - \Delta t \, \mathbf{Q}_{\text{coll},f} \mathbf{F}_f(\mathcal{P}\mathbf{U}_c)\right),$$

which works well because the linear operator and the transfer operators commute. However, when the operator is nonlinear, this approach may fail. For a nonlinear function $\mathbf{F}_f$, it is generally not true that

$$\mathbf{F}_f\left(I_c^f \, \mathbf{U}_c\right) = I_c^f \, \mathbf{F}_f(\mathbf{U}_c).$$

Thus, the projected fine-grid operator $R_f^c \, \mathbf{M}_{\text{coll},f}(I_c^f \, \mathbf{U}_c)$ does not accurately represent the true nonlinear dynamics on the coarse grid.

Because of these issues, the Galerkin approach for constructing the collocation operator $\mathbf{M}_{\text{coll},c}$ can fail to produce a reliable coarse-grid correction in MLSDC method.

In such cases, it is generally preferable to use a *rediscretization* strategy, where the coarse operator (4.19) is derived directly from a discretization of the coarse problem using the coarse-level collocation nodes (as described in Section 2.1). This approach avoids the pitfalls associated with the nonlinearity and ensures that the coarse-grid operator faithfully represents the behavior of the original problem.

4.4 Numerical examples

In this section, we analyze the MLSDC method for the Penning trap problem introduced in Section 3.3. We examine the convergence order of the MLSDC method by investigating the relative error in the first and third components. We also vary the number of quadrature nodes on the coarse level to study its effect on the global convergence order of the MLSDC method. Note that a single MLSDC iteration consists of one SDC sweep on the coarse level followed by one SDC sweep on the fine level.

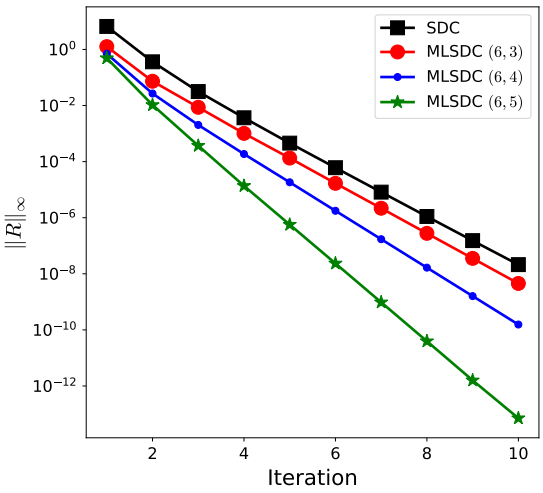

Figure 4.5: The infinity norm of the residual versus the number of iterations for SDC and MLSDC methods. (M_f, M_c) represents the number of Gauss-Legendre quadrature nodes on the fine $M_f = 6$ and coarse level $M_c = 3, 4, 5$ for MLSDC method and the number of quadrature nodes is 6 for SDC method with 10 iterations.

4.4.1 Residual of SDC and MLSDC

Figure 4.5 shows the infinity norm of the residual with respect to the number of iterations for both the SDC and MLSDC methods. The notation (M_f, M_c) indicates the number of Gauss-Legendre quadrature nodes, where $M_f = 6$ denotes the number of fine-level nodes, and $M_c = 3, 4, 5$ represent the number of coarse-level nodes in the MLSDC method. For the SDC method, the number of Gauss-Legendre quadrature nodes is fixed at 6, with $K = 10$ iterations. The results demonstrate that the MLSDC method produces smaller residuals in the same number of iterations compared to the SDC method. Furthermore,

increasing the number of quadrature nodes on the coarse level in the MLSDC method leads to a reduction in the residual.

4.4.2 Global error of MLSDC

Figure 4.6 shows the relative error of the MLSDC method in the first and third velocity components using Gauss-Legendre quadrature nodes with random initialization. The number of quadrature nodes on the fine and coarse levels is set to 5 and 3, respectively, with a fixed final time of $t_{\text{end}} = 2$.

Since an additional SDC iteration is performed on the coarse level, when $K = 1$ the order of convergence is 2 in the v_1-direction and 4 in the v_3-direction. We also observe that in the v_1-direction each iteration increases the global convergence order by two, whereas in the v_3-direction—where the force is independent of velocity—each iteration increases the global order by only one.

Interestingly, when comparing with the SDC iterations shown in Figure 3.8, we observe that each SDC iteration increases the global error order by one in the v_1-direction and by two in the v_3-direction. The reason for this discrepancy remains unclear.

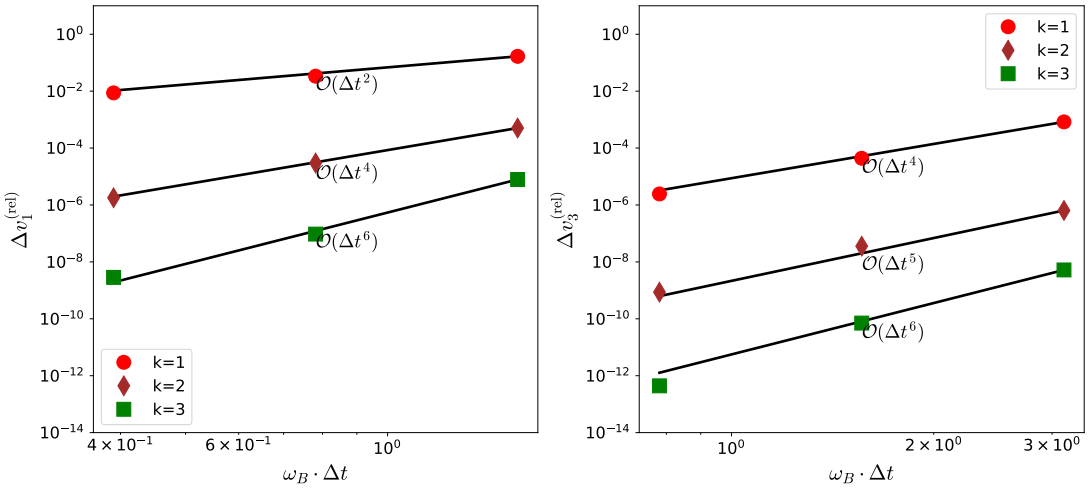

Figure 4.6: Relative global error $\Delta v_i^{(\text{rel})}$ for $i = 1, 3$ in the v_1-direction (left) and v_3-direction (right) in the Penning trap, plotted against the time step size. The method uses 5 fine and 3 coarse Gauss-Legendre collocation nodes with $K = 1,\ 2,\ 3$ MLSDC iterations.

Chapter 5

Macroscopic models using perturbation analysis

In the previous Chapter 4, we introduced the MLSDC method for second-order problems and proposed two coarsening strategies. The second strategy 2 involves using macroscopic models as a coarse solver. These models can be obtained either analytically or numerically [34]. Perturbation analysis [6] is employed to derive the analytical form of these models, which are often referred to as *asymptotic* or *reduced models*.

In this chapter, we explain how to obtain macroscopic models through perturbation analysis. First, we discuss both regularly and singularly perturbed ODEs, as described in [6, Chapter 3]. Next, we present macroscopic models for the Duffing equation [6, pp. 93–95], the harmonic oscillator problem [7], and the Lorentz equation [8, 63].

We will first provide an overview of various definitions in this context [67, Section 1.2.1], which will be used throughout the remainder of this dissertation.

Remark 5.1. For a given domain D and ε-interval $I : 0 < \varepsilon \leq \varepsilon_1$ with given positive constant ε_1, the statement

$$u(x, \varepsilon) = \mathcal{O}(v(x, v)) \text{ in } I \tag{5.1}$$

means that for each x in D there exists a positive number $k(x)$ such that

$$|u(x, \varepsilon)| \leq k(x)|v(x, \varepsilon)| \tag{5.2}$$

for all ε in I. Similarly, we say that

$$u(x, \varepsilon) = \mathcal{O}(v(x, \varepsilon)) \text{ as } \varepsilon \to 0, \tag{5.3}$$

if for each x in D, there exists a positive number $k(x)$ and a neighborhood N of $\varepsilon = 0$ such that (5.1) holds for all ε in the intersection of N with I.

Remark 5.2. For a given domain D, the statement

$$u(x, \varepsilon) = o(v(x, \varepsilon)) \text{ as } \varepsilon \to 0 \tag{5.4}$$

means that for each point x in D and any given $\delta > 0$, there exists an $\varepsilon-$interval $I(x, \delta) : 0 < \varepsilon \leq \varepsilon_1(x, \delta)$, with a positive upper bound $\varepsilon_1(x, \delta)$ which depends on x and δ, such that

$$|u(x, \varepsilon)| \leq \delta|v(x, \varepsilon)| \tag{5.5}$$

for all ε in I. The inequality (5.5) indicates that $|u|$ becomes arbitrarily small compared to $|v|$ as $\varepsilon \to 0$. Often, the notation

$$u \ll v$$

is used to indicate (5.4).

Definition 5.3. Consider a sequence of functions $\{\phi_n(\varepsilon)\},\ n = 1, 2, \ldots$. Such a sequence is called *an asymptotic sequence* if

$$\phi_{n+1}(\varepsilon) = o(\phi_n(\varepsilon)) \text{ as } \varepsilon \to 0$$

for each $n = 1, 2, \ldots$.

Definition 5.4. Let $u(x, \varepsilon)$ be defined in some domain D of x and some neighborhood of $\varepsilon = 0$. Let $\{\phi_n(\varepsilon)\}$ be a given asymptotic sequence. The series $\sum_{n=1}^{N} \phi_n(\varepsilon)u_n(x)$ is called *the asymptotic expansion of* $u(x, \varepsilon)$ to N terms (N may be a finite integer or infinity) as $\varepsilon \to 0$ with respect to the sequence $\{\phi_n(\varepsilon)\}$ if

$$u(x, \varepsilon) - \sum_{n=1}^{M} \phi_n(\varepsilon)u_n(x) = o(\phi_M(\varepsilon)) \text{ as } \varepsilon \to 0 \tag{5.6}$$

for each $M = 1, 2, \ldots, N$.

If $N = \infty$, the following notation is generally used to indicate an asymptotic expansion

$$u(x, \varepsilon) \simeq \sum_{n=1}^{\infty} \phi_n(\varepsilon)u_n(x) \text{ as } \varepsilon \to 0.$$

Remark 5.5. A definition of an asymptotic expansion equivalent to (5.6) is

$$u(x, \varepsilon) - \sum_{n=1}^{M} \phi_n(\varepsilon)u_n(x) = \mathcal{O}(\phi_{M+1}(\varepsilon)) \text{ as } \varepsilon \to 0 \tag{5.7}$$

for each $M = 1, 2, \ldots, N - 1$. An asymptotic expansion is said to be *uniformly valid* in D if the order relations in (5.6) or (5.7) hold uniformly in D.

5.1 Regular and singular perturbed ODEs

In this section, we explain perturbation analysis for ODEs as presented in [6, Chapter 3]. A system of first-order differential equations can be written as

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}(t, \mathbf{y}, \varepsilon), \tag{5.8}$$

where $\mathbf{y} \in \mathbb{R}^d$ and $\mathbf{f} \in \mathbb{R}^d$ are vector-valued functions, $t \in \mathbb{R}$ is an independent scalar variable and ε is a small parameter. The system (5.8) is said to be *regularly perturbed* if

$$\mathbf{f}(t, \mathbf{y}, \varepsilon) \simeq \sum_{k=0}^{\infty} \mathbf{f}^{(k)}(t, \mathbf{y})\varepsilon^k. \tag{5.9}$$

Conversely, a system of the form

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}(t, \mathbf{y}, \frac{1}{\varepsilon}), \tag{5.10}$$

is called *singularly perturbed* if

$$\mathbf{f}(t, \mathbf{y}, \frac{1}{\varepsilon}) \simeq \sum_{k=0}^{\infty} \mathbf{f}^{(k)}(t, \mathbf{y}) \frac{1}{\varepsilon^k}. \tag{5.11}$$

If the solution of the system can be expressed as an asymptotic series

$$\mathbf{y}(t, \varepsilon) \simeq \mathbf{y}^{(0)}(t) + \varepsilon \mathbf{y}^{(1)}(t) + \varepsilon^2 \mathbf{y}^{(2)}(t) + \mathcal{O}(\varepsilon^3) \tag{5.12}$$

then expansions of this type referred to as *direct expansion* [68].

Substituting (5.9) and (5.12) in (5.8) and equating coefficients of equal powers of ε in the right and in the left side of system (5.8), we get a sequence of equations to find the vector functions

$$\frac{d\mathbf{y}^{(0)}}{dt} = \mathbf{f}(t, \mathbf{y}^{(0)}, 0), \tag{5.13a}$$

$$\frac{d\mathbf{y}^{(1)}}{dt} = \mathbf{f}(t, \mathbf{y}^{(0)}, \mathbf{y}^{1}, 0), \tag{5.13b}$$

$$\vdots \tag{5.13c}$$

$$\frac{d\mathbf{y}^{(k)}}{dt} = \mathbf{f}(t, \mathbf{y}^{(0)}, \ldots, \mathbf{y}^{(k-1)}, 0) \tag{5.13d}$$

The differential system determining $\mathbf{y}^{(0)}(t)$ is called the *generating system* for system (5.8). The systems of equations for $\mathbf{y}^{(1)}(t)$, $\mathbf{y}^{(2)}(t), \ldots$ is linear. If the generating system (5.13a) is also linear, then (5.8) is called a *quasilinear system.*

For the second order ODEs, the equation

$$\frac{d^2x}{dt^2} = \mathbf{f}(t, x, \varepsilon), \tag{5.14}$$

where the right hand side $\mathbf{f}$ is expanded as

$$\mathbf{f}(t, x, \varepsilon) \simeq \sum_{k=0}^{\infty} \mathbf{f}^{(k)}(t, x)\varepsilon^k, \tag{5.15}$$

is regularly perturbed since after the change of variables $\dot{x} = v$, it reduces to the system

$$\begin{aligned} \frac{dx}{dt} &= v, \\ \frac{dv}{dt} &= \mathbf{f}(t, x, \varepsilon). \end{aligned}$$

Solution of regularly perturbed differential equation may be represented in the form

$$x(t, \varepsilon) \simeq \sum_{k=0}^{\infty} x^{(k)}(t)\varepsilon^k.$$

5.2 Duffing equation

In this section, we analyze methods for obtaining macroscopic models for a regularly perturbed system using the direct expansion (5.12), as presented in [6, pp. 93-95]. Consider the following Duffing equation

$$\ddot{x} + w^2 x + \varepsilon b x^3 = 0 \tag{5.17}$$

which models the vibrations of a mass on a nonlinear spring. Here x and t are dimensionless variables, ε is a small parameter, ω and b are coefficients of the linear and nonlinear restoring forces respectively, and the initial conditions are given by

$$x(0) = a, \quad v(0) = \dot{x}(0) = 0, \tag{5.18}$$

for some $a \in \mathbb{R}$. We seek a solution for problem (5.17)-(5.18) in the form of a series

$$x = x^{(0)} + \varepsilon x^{(1)} + \varepsilon^2 x^{(2)} + \mathcal{O}(\varepsilon^3). \tag{5.19}$$

Substituting (5.19) into (5.17) yields

$$\ddot{x}^{(0)} + \varepsilon \ddot{x}^{(1)} + \varepsilon^2 \ddot{x}^{(2)} + w^2(x^{(0)} + \varepsilon x^{(1)} + \varepsilon^2 x^{(2)}) + \varepsilon b(x^{(0)} + \varepsilon x^{(1)} + \varepsilon^2 x^{(2)})^3 + \mathcal{O}(\varepsilon^3) = 0. \tag{5.20}$$

By equating the terms that do not contain ε, we obtain the generating equation (or the zeroth-order macroscopic model)

$$\ddot{x}^{(0)} + w^2 x^{(0)} = 0. \tag{5.21}$$

Plugging in (5.19) into (5.18) provides initial conditions

$$x^{(0)}(0) = a, \quad v^{(0)}(0) = \dot{x}^{(0)}(0) = 0. \tag{5.22}$$

Matching coefficients of equal powers of ε on both sides, we derive the equations for the first and second approximations to determine $x^{(1)}$ and $x^{(2)}$

$$\ddot{x}^{(1)} + w^2 x^{(1)} + b(x^{(0)})^3 = 0, \tag{5.23a}$$

$$\ddot{x}^{(2)} + w^2 x^{(2)} + 3b x^{(1)} (x^{(0)})^2 = 0. \tag{5.23b}$$

Using (5.17) and (5.22), it follows that

$$x^{(k)}(0) = \dot{x}^{(k)}(0) = 0 \quad \text{for} \quad k = 1,\ 2. \tag{5.24}$$

The general solution of the generating equation (5.21) is

$$x^{(0)} = M_0 \cos(z) + N_0 \sin(z), \quad z = wt,$$

Applying the initial conditions (5.22), we find $M_0 = a,\ N_0 = 0$, hence

$$x^{(0)} = a \cos(z). \tag{5.25}$$

Substituting (5.25) into (5.23a) and using the identity

$$\cos^3(z) = \frac{1}{4}(\cos(3z) + 3\cos(z)).$$

yields

$$\ddot{x}^{(1)} + w^2 x^{(1)} = -\frac{a^3 b}{4}(\cos(3z) + 3\cos(z)). \tag{5.26}$$

The general solution of this equation is a sum of a particular solution and the solution of the corresponding homogeneous equation. Using the superposition principle, we set $x^{(1)} = x_1^{(1)} + x_2^{(1)}$, where

$$\ddot{x}_1^{(1)} + w^2 x_1^{(1)} = -\frac{a^3 b}{4}\cos(3z) \quad \text{and} \quad \ddot{x}_2^{(1)} + w^2 x_2^{(1)} = -\frac{3a^3 b}{4}\cos(z).$$

We assume a solution of the form $x_1^{(1)} = A\cos(3z)$, and $x_2^{(1)} = Bt\sin(z)$. After evaluating of the constants A and B, we obtain

$$x^{(1)} = M_1\cos(z) + N_1\sin(z) - \frac{a^3 b}{32w^2}(12z\sin(z) - \cos(3z)).$$

Using initial conditions (5.24), we find the constants

$$M_1 = -\frac{a^3 b}{32w^2}, \quad N_1 = 0.$$

Thus, the two-term of macroscopic approximation for the solution of (5.17) is

$$x(t,\varepsilon) \simeq a\cos(z) - \varepsilon\frac{a^3 b\sin(z)(\sin(2z) + 6z)}{16w^2}. \tag{5.27}$$

In a similar manner, substituting the expressions for $x^{(0)}$ and $x^{(1)}$ in (5.23b), one can obtain the second approximation $x^{(2)}$.

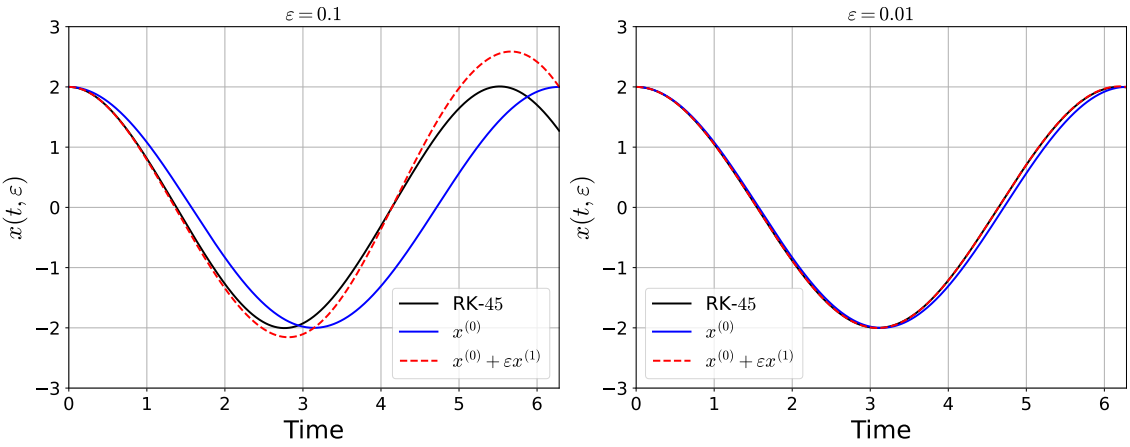

Figure 5.1: Numerical solution using RK-45 (solid line), $\mathcal{O}(\varepsilon^1) : x^{(0)}$, zeroth-order macroscopic model solution (dotted line), and $\mathcal{O}(\varepsilon^2) : x^{(0)} + \varepsilon x^{(1)}$, first-order macroscopic model solution (dash doted line) with $\varepsilon = 0.1$ on the left and $\varepsilon = 0.01$ on the right [6, Fig. 3.3].

In Figure 5.1, the solid line shows the solution of equation (5.17) computed using a Runge-Kutta-45 method with parameters $w = b = 1$ and $a = 2$ over the interval $[0, 2\pi]$. Two cases are illustrated: $\varepsilon = 0.1$ (left) and $\varepsilon = 0.01$ (right). The dotted line represents the zeroth-order model solution given by formula (5.25), while the dash-dotted line

corresponds to the solution obtained by $x^{(0)} + \varepsilon x^{(1)}$. Note that for $\varepsilon = 0.1$, the zeroth-order macroscopic model is less accurate, with its error growing over time even when the $x^{(0)} + \varepsilon x^{(1)}$ approximation is used. However, when $\varepsilon = 0.01$, both macroscopic model solutions provide a highly accurate approximation of the solution to equation (5.17), as computed using the Runge-Kutta-45 method.

5.3 Harmonic oscillator

In this section, we derive macroscopic models for the harmonic oscillator problem using perturbation analysis techniques as presented in [7, pp. 159–165]. The equation is given by

$$\varepsilon\ddot{x} + \varepsilon\hat{\kappa}\dot{x} + x = \cos(t) \tag{5.28}$$

where the external forcing term is $\cos(t)$, the dimensionless mass and damping parameters are given by $(\varepsilon,\ \varepsilon\hat{\kappa})$, where $\hat{\kappa}$ is a constant with a small parameter ε. The exact solution to the equation (5.28) is

$$x(t) = e^{-\frac{\hat{k}}{2}t}\left(c_1\cos\left(\frac{\sqrt{4-\varepsilon\hat{k}}}{a\sqrt{\varepsilon}}t\right) + c_2\sin\left(\frac{\sqrt{4-\varepsilon\hat{k}}}{a\sqrt{\varepsilon}}t\right)\right) + \frac{1-\varepsilon}{(1-\varepsilon)^2+\varepsilon^2+\hat{k}^2}\cos(t) + \frac{\varepsilon\hat{k}}{(1-\varepsilon)^2+\varepsilon^2+\hat{k}^2}\sin(t) \tag{5.29}$$

Define a new time scale as

$$\nu = \frac{t}{\sqrt{\varepsilon}} \tag{5.30}$$

and we introduce a new expansion scheme in which unknowns are expressed as a function of ν instead of t. With the new time scale, the asymptotic expansion is given by

$$x(t,\varepsilon) \simeq x^{(0)}(\nu) + \sqrt{\varepsilon}x^{(1)}(\nu) + \varepsilon x^{(2)}(\nu) + o(\varepsilon) \tag{5.31}$$

which is equivalent to

$$x(t,\varepsilon) \simeq x^{(0)}\left(\frac{t}{\sqrt{\varepsilon}}\right) + \sqrt{\varepsilon}x^{(1)}\left(\frac{t}{\sqrt{\varepsilon}}\right) + \varepsilon x^{(2)}\left(\frac{t}{\sqrt{\varepsilon}}\right) + o(\varepsilon) \tag{5.32}$$

To determine macroscopic models using (5.32), we compute the t-derivatives for fixed ε,

$$\left.\frac{\partial x}{\partial t}\right|_{\varepsilon}(\nu;\varepsilon) = \frac{1}{\sqrt{\varepsilon}}\frac{dx^{(0)}}{d\nu} + \frac{dx^{(1)}}{d\nu} + \sqrt{\varepsilon}\frac{dx^{(2)}}{d\nu} + o(\sqrt{\varepsilon}) \tag{5.33}$$

$$\left.\frac{\partial^2 x}{\partial t^2}\right|_{\varepsilon}(\nu;\varepsilon) = \frac{1}{\varepsilon}\frac{d^2x^{(0)}}{d\nu^2} + \frac{1}{\sqrt{\varepsilon}}\frac{d^2x^{(1)}}{d\nu^2} + \frac{d^2x^{(2)}}{d\nu^2} + o(1) \tag{5.34}$$

The notation $\left.\frac{\partial x}{\partial t}\right|_{\varepsilon}$ indicates that ε is held constant when differentiating, assuming it is a fixed parameter for the oscillator.

We express the forcing term in (5.28) in terms of ν and then expand in powers of $\sqrt{\varepsilon}$

$$\cos(t) = \cos(\sqrt{\varepsilon}\nu) = 1 - \frac{\varepsilon}{2}\nu^2 + o(\varepsilon^2) \tag{5.35}$$

Substituting the derivatives and expansions into (5.28) gives

$$\frac{d^2x^{(0)}}{d\nu^2} + \sqrt{\varepsilon}\frac{d^2x^{(1)}}{d\nu^2} + \varepsilon\frac{d^2x^{(2)}}{d\nu^2} + \hat{\kappa}\sqrt{\varepsilon}\frac{dx^{(0)}}{d\nu} + \hat{\kappa}\varepsilon\frac{dx^{(1)}}{d\nu} + x^{(0)} + \sqrt{\varepsilon}x^{(1)} + \varepsilon x^{(2)} = 1 - \varepsilon\frac{\nu^2}{2} + o(\varepsilon). \tag{5.36}$$

By collecting terms of like powers of ε, we obtain the following hierarchy of perturbation equations

$$o(1): \quad \frac{d^2x^{(0)}}{d\nu^2} + x^{(0)} = 1 \tag{5.37}$$

$$o(\sqrt{\varepsilon}): \quad \frac{d^2x^{(1)}}{d\nu^2} + x^{(1)} = -\hat{\kappa}\frac{dx^{(0)}}{d\nu} \tag{5.38}$$

$$o(\varepsilon): \quad \frac{d^2x^{(2)}}{d\nu^2} + x^{(2)} = -\hat{\kappa}\frac{dx^{(1)}}{d\nu} - \frac{\nu^2}{2} \tag{5.39}$$

At leading order (i.e., at $o(1)$), we find the equation for an undamped oscillator with constant forcing. The solution is

$$x^{(0)} = a_0\cos(\nu) + b_0\cos(\nu) + 1 \tag{5.40}$$

where a_0 and b_0 are constants that can be found by using the initial conditions.

Next, we solve the asymptotic equation of $o(\sqrt{\varepsilon})$

$$\frac{d^2x^{(1)}}{d\nu^2} + x^{(1)} = -\hat{\kappa}(a_0\cos(\nu) - b_0\sin(\nu)) \tag{5.41}$$

The solution is the sum of a homogeneous solution, $x_h^{(1)} = a_1\cos(\nu) + b_1\cos(\nu)$, and the coefficients a_1 and b_1 will have to be computed from the initial conditions as before. A particular solution found using the method of variation parameters [69]

$$x_p^{(1)} = f(\nu)\sin(\nu) + g(\nu)\cos(\nu) \tag{5.42}$$

where f and g remain to be determined. Inserting this ansatz into the differential equation (5.41) yields

$$(\ddot{f} - 2\dot{g})\sin(\nu) + (\ddot{g} + 2\dot{f})\cos(\nu) = -\hat{\kappa}(a_0\cos(\nu) - b_0\sin(\nu)) \tag{5.43}$$

Comparing coefficients, we find the constraints

$$\ddot{f} - 2\dot{g} = \hat{\kappa}b_0 \quad \text{and} \quad \ddot{g} + 2\dot{f} = -\hat{\kappa}a_0 \tag{5.44}$$

The desired solutions are polynomials in ν of degree less than two, and we let $f(\nu) = A_f\nu + B_f$ and $g(\nu) = A_g\nu + B_g$. Without loss of generality, we may also assume $B_f = 0$

and $B_g = 0$, as these terms can be covered by the homogeneous part of the solution. Solving (5.44), yields

$$x_p^{(1)} = -\nu\frac{\hat{\kappa}}{2}(a_0\sin(\nu) + b_0\cos(\nu)) \tag{5.45}$$

so that

$$x^{(1)} = x_h^{(1)} + x_p^{(1)} = a_1\sin(\nu) + b_1\cos(\nu) - \nu\frac{\hat{\kappa}}{2}(a_0\sin(\nu) + b_0\cos(\nu)). \tag{5.46}$$

Higher-order solutions may be derived similarly.

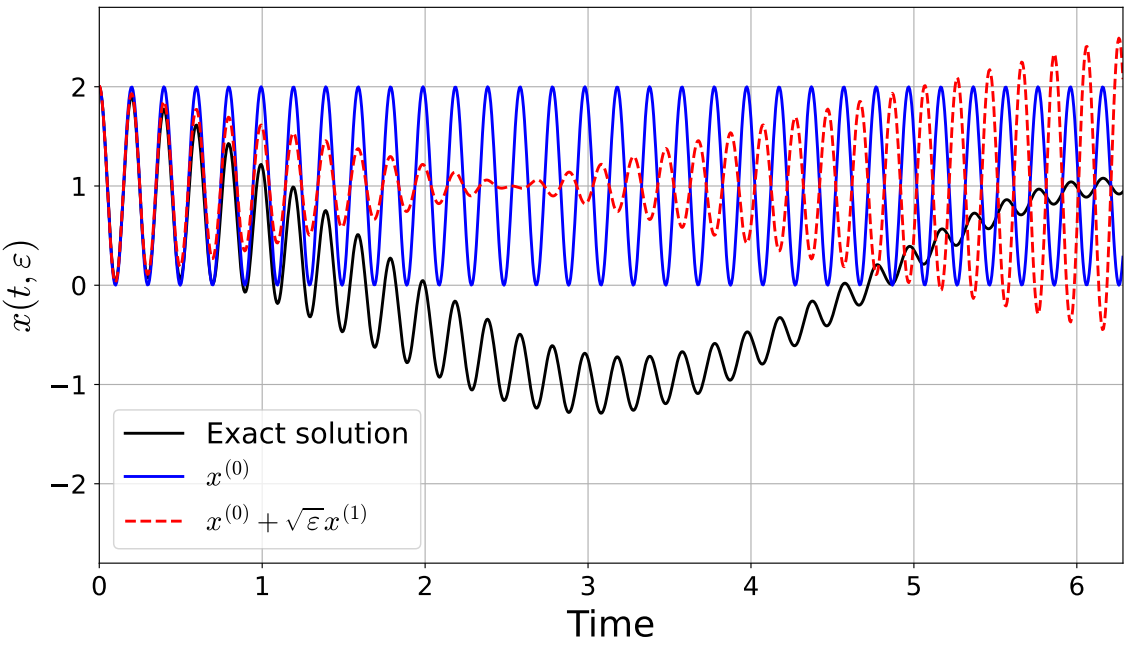

Figure 5.2: The solution of exact (5.29) and macroscopic model solutions with parameters $x_0 = 2,\ \dot{x}_0 = 0,\ \varepsilon = 0.001$ and $\hat{\kappa} = 0.8$. $x(t,\varepsilon)$ (Exact solution): black line; $x^{(0)}$: blue line; $x^{(0)} + \sqrt{\varepsilon}x^{(1)}$: red dashed line [7, Figure 11.]

This approach based on the fast-time variable is accurate for short times. As shown in Figure 5.2, although the solution improves with higher-order approximations, this improvement only holds in the early stages of evolution. Over time, the solution accuracy decreases significantly, and the error grows with increasing approximation order.

5.4 Lorentz equation

In this section, we present macroscopic models for the specific cases of the Lorentz equation. Most of the results presented here have been published in [8, 70]. However, we include only the results obtained in the following paper [8]. For a detailed understanding of deriving macroscopic models for these problems, we refer to [70].

We aim to solve the following six-dimensional system for $0 < \varepsilon \ll 1$

$$\frac{dx_\varepsilon}{dt} = v_\varepsilon, \qquad x_\varepsilon(s) = x, \tag{5.47a}$$

$$\frac{dv_\varepsilon}{dt} = \frac{1}{\varepsilon}(v_\varepsilon \times \mathbf{B}_\varepsilon(x_\varepsilon)) + \mathbf{E}(t, x_\varepsilon), \qquad v_\varepsilon(s) = v \tag{5.47b}$$

where (x, v) is an initial condition given at the initial time $t = s$. The system (5.47) models the dynamics of a charged particle under the influence of an external electromagnetic field. In this context, $x_\varepsilon : \mathbb{R} \mapsto \mathbb{R}^3$ represents the position, $v_\varepsilon : \mathbb{R} \mapsto \mathbb{R}^3$ represents the velocity, and $\mathbf{E} : \mathbb{R} \times \mathbb{R}^3 \mapsto \mathbb{R}^3$ and $\frac{1}{\varepsilon}\mathbf{B}_\varepsilon : \mathbb{R}^3 \mapsto \mathbb{R}^3$ represent the given electric and magnetic fields. We assume that $|\mathbf{B}_\varepsilon| = 1$ and that the mass and the charged particle are both equal to 1. The parameter $1/\varepsilon$ preceding the $v_\varepsilon \times \mathbf{B}_\varepsilon(x_\varepsilon)$ term indicates that the magnetic field is significantly stronger than the electric field, in line with plasma confinement considerations [71]. The frequency at which a particle will orbit in a perpendicular magnetic field is known as the cyclotron frequency defined as

$$\omega_{cy} := \frac{q}{m}B = \frac{1}{\varepsilon} \tag{5.48}$$

where $B = 1/\varepsilon$ denotes the strength of the magnetic field and the charge-to-mass ratio q/m is assumed to be 1 [30, Chapter 5].

5.4.1 Two-scale asymptotic expansion

We summarize the principle and the main result of the two-scale asymptotic expansion method used to obtain macroscopic models. The equation (5.47) is a particular instance of the more general singularly perturbed dynamical system

$$\frac{d\mathcal{X}_\varepsilon}{dt} = \mathbf{a}(t, \mathcal{X}_\varepsilon) + \frac{1}{\varepsilon}\mathbf{b}(t, \mathcal{X}_\varepsilon), \quad \mathcal{X}_\varepsilon(s) = \mathcal{X}, \tag{5.49}$$

where $\mathcal{X}_\varepsilon : \mathbb{R} \mapsto \mathbb{R}^d$ and $\mathbf{a}$ and $\mathbf{b}$ are given fields, satisfying certain assumptions with s representing the initial time. Following the methodology in [70], we provide a brief overview of the asymptotic two-scale expansion technique to approximate the solution $\mathcal{X}_\varepsilon$ as $\varepsilon \to 0$. Under regularity assumptions on $\mathbf{a}$ and $\mathbf{b}$ and assuming the periodicity in θ of the solution $\mathbf{Z}(t; \theta, z)$ to the equation

$$\frac{d\mathbf{Z}}{d\theta} = \mathbf{b}(t, \mathbf{Z}), \quad \mathbf{Z}(t, \theta, z) = z \tag{5.50}$$

for every $t \in \mathbb{R}$ and every $z \in \mathbb{R}^d$, it is proven in [70] that $\mathcal{X}_\varepsilon$ can be expressed as the following two-scale expansion in time

$$\mathcal{X}_\varepsilon(t) = \mathcal{X}^{(0)}\left(t, \frac{t-s}{\varepsilon}\right) + \varepsilon\mathcal{X}^{(1)}\left(t, \frac{t-s}{\varepsilon}\right) + \varepsilon^2\mathcal{X}^{(2)}\left(t, \frac{t-s}{\varepsilon}\right) + \mathcal{O}(\varepsilon^3) \tag{5.51}$$

where the functions $\mathcal{X}^{(i)}(t, \theta)$ are periodic in θ for each $i \in \mathbb{N}$. In this framework, singularly perturbed ODEs that characterize the terms of the expansion (5.51) are derived in [70, Theorem 1.1 & 1.3]. Furthermore, strong convergence theorems are proved, justifying the approximation results. For instance, for the zeroth-order macroscopic model, we have

$$\mathcal{X}_\varepsilon(t) \simeq \mathcal{X}^{(0)}\left(t, \frac{t-s}{\varepsilon}\right), \quad \text{as} \quad \varepsilon \to 0$$

and for the second-order macroscopic model

$$\mathcal{X}_\varepsilon(t) \simeq \mathcal{X}^{(0)}\left(t, \frac{t-s}{\varepsilon}\right) + \varepsilon\mathcal{X}^{(1)}\left(t, \frac{t-s}{\varepsilon}\right), \quad \text{as} \quad \varepsilon \to 0.$$

More precisely, we have

$$\left|\mathcal{X}_\varepsilon(t) - \mathcal{X}^{(0)}\left(t, \frac{t-s}{\varepsilon}\right)\right| \leq C\varepsilon \tag{5.52}$$

and

$$\left|\mathcal{X}_\varepsilon(t) - \mathcal{X}^{(0)}\left(t, \frac{t-s}{\varepsilon}\right) - \varepsilon\mathcal{X}^{(1)}\left(t, \frac{t-s}{\varepsilon}\right)\right| \leq C\varepsilon^2 \tag{5.53}$$

For the sake of completeness, we give below the result concerning the two-scale limit model [70, Theorem 1.1] in the case of a six dimensional space ($d = 6$).

Theorem 5.6. *We assume that*[1] $a \in \left(\mathbf{C}_b^1(\mathbb{R}\times\mathbb{R}^6)\right)^6$ *and* $\mathbf{b} \in \left(\mathbf{C}_b^2(\mathbb{R}\times\mathbb{R}^6)\right)^6$. *Assume also that the solution of* (5.50) *is* 2π*-periodic in* θ, *for every* $t \in \mathbb{R}$ *and every* $\mathbf{z} \in \mathbb{R}^6$. *Then, for every initial condition* $\mathcal{X} \in \mathbb{R}^6$, *every* $\varepsilon > 0$, *and every* $\Delta S > 0$, *the solution* $\mathcal{X}_\varepsilon$ *of* (5.49) *exists on* $[s, s+\Delta S]$, *is unique and satisfies*

$$\lim_{\varepsilon\to 0}\sup_{t\in[s,s+\Delta S]}\left|\mathcal{X}_\varepsilon(t) - \mathcal{X}^{(0)}\left(t, \frac{t-s}{\varepsilon}\right)\right| = 0, \tag{5.54}$$

where $|\cdot|$ *stands for the Euclidean norm on* $\mathbb{R}^6$ *and* $\mathcal{X}^{(0)}$ *satisfies*

$$\mathcal{X}^{(0)}(t,\theta) = \mathbf{Z}(t,\theta,\mathcal{Y}^{(0)}(t)) \tag{5.55}$$

where $\mathcal{Y}^{(0)}$ *is the solution to*

$$\frac{d\mathcal{Y}^{(0)}}{dt} = \alpha(t,\mathcal{Y}^{(0)}), \quad \mathcal{Y}^{(0)} = \mathcal{X}, \tag{5.56}$$

with α *defined by*

$$\alpha(t,\mathcal{Y}) = \frac{1}{2\pi}\int_0^{2\pi}\{\nabla\mathbf{Z}(t,\theta,\mathcal{Y})\}^{-1}\left\{\mathbf{a}(t,\mathbf{Z}(t,\theta,\mathcal{Y})) - \frac{\partial\mathbf{Z}}{\partial t}(t,\theta,\mathcal{Y})\right\}d\theta. \tag{5.57}$$

Remark 5.7. We remark that the limit model (5.56) does not contain high oscillations in time so that cheap numerical schemes can be used to compute $\mathcal{Y}^{(0)}$. Then, when $\mathbf{Z}$ is known in (5.50), we obtain the term $\mathcal{X}^{(0)}$ by (5.55), as an approximation of the solution $\mathcal{X}_\varepsilon$ in the sense of (5.54). Through obtained at a low computational cost, the approximation $\mathcal{X}^{(0)}$ still contains information about the high oscillations in the solution, through the operator $\mathbf{Z}$. These facts underline that the solution to the limit model give by (5.55)-(5.56) is good candidate for a coarse problem in MLSDC framework.

In the subsequent sections, we develop this framework for equations of the type of (5.47), be using the notation $\mathcal{X} = (x, v)^T$, where, as in the classical mechanics $x = (x_1, x_2, x_3)^T$ stands for the position vector and $v = (v_1, v_2, v_3)^T$ for the velocity vector.

5.4.2 The case of a constant magnetic field

We consider the system (5.47) with a constant magnetic field $\mathbf{B}_\varepsilon = \overrightarrow{e_1}$, where $\{\overrightarrow{e_1}, \overrightarrow{e_2}, \overrightarrow{e_3}\}$ is the standard basis in $\mathbb{R}^3$ along with a given external electric field. In this setup, the term $v_\varepsilon \times \mathbf{B}_\varepsilon(x_\varepsilon)$ in the velocity equation (5.47) becomes $(v_\varepsilon)^\perp = (0, (v_\varepsilon)_3, -(v_\varepsilon)_2)^T$. Thus, the basic assumption of periodicity of the solution to (5.50) is satisfied. A common feature of the test cases, we address in this section is that we can compute analytic solutions for equation (5.47) and the corresponding macroscopic model.

[1] $\mathbf{C}_b^m$ stands for the space of continuous functions with bounded derivatives to the order m.

A uniform time varying electric field. We consider an electric field that oscillates quickly in time. In this case, the system (5.47) is expressed as

$$\frac{dx_\varepsilon}{dt} = v_\varepsilon, \qquad x_\varepsilon(s) = x, \tag{5.58a}$$

$$\frac{dv_\varepsilon}{dt} = \frac{1}{\varepsilon}(v_\varepsilon)^\perp + \mathbf{E}(\frac{t}{\varepsilon}), \qquad v_\varepsilon(s) = v, \tag{5.58b}$$

where $\mathbf{E}$ is defined as $\mathbf{E}(\tau) = (E_1, E_2(\tau), E_3(\tau))^T$, with $E_1 \in \mathbb{R}$ and E_2, E_3 being $2\pi-$periodic functions, see [70, Section 3.1]. This model can describe ion cyclotron resonance relevant in isotope separation in plasma, as discussed in [72]. In magnetized plasmas, ions are heated by an oscillating perpendicular electric field at frequencies matching the ion cyclotron frequency causing an increase in motion amplitude over time. For illustration, we consider the electric field

$$E_1 = 0, \quad E_2(\tau) = \sin(\tau), \quad E_3(\tau) = \cos(\tau). \tag{5.59}$$

However, similar results can be derived for a general electric field with these properties [70]. For the following matrices, we define

$$P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \mathcal{N}(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) \\ 0 & -\sin(\theta) & \cos(\theta) \end{pmatrix}, \mathcal{R}(\theta) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \sin(\theta) & 1-\cos(\theta) \\ 0 & \cos(\theta) & \sin(\theta) \end{pmatrix}. \tag{5.60}$$

Note that the matrix $\mathcal{N}$ is denoted by R in the original paper [8]. We express the solution of (5.58)-(5.59) in the form

$$\begin{pmatrix} x_\varepsilon(t) \\ v_\varepsilon(t) \end{pmatrix} = \mathcal{A} \begin{pmatrix} x \\ v \end{pmatrix} + \mathcal{B} \tag{5.61}$$

where

$$\mathcal{A} = \begin{pmatrix} I_3 & (t-s)P + \varepsilon \mathcal{R}\left(\frac{t-s}{\varepsilon}\right) \\ O_3 & \mathcal{N}\left(\frac{t-s}{\varepsilon}\right) \end{pmatrix} \in \mathbb{R}^{6\times 6} \tag{5.62}$$

and

$$\mathcal{B} = \begin{pmatrix} \varepsilon(t-s)(0, -\cos(t/\varepsilon), \sin(t/\varepsilon))^T + \varepsilon^2(0, \sin(t/\varepsilon), \cos(t/\varepsilon) - \cos(s/\varepsilon))^T \\ (t-s)(0, \sin(t/\varepsilon), \cos(t/\varepsilon))^T \end{pmatrix} \in \mathbb{R}^6$$

with I_3 as the 3×3 identity matrix and O_3 the 3×3 zero matrix.

To derive the macroscopic model for equation (5.58), we apply the results from [70, Theorem 3.1, 3.2] to obtain the second-order two-scale method. The approximation of the solution is

$$\mathbf{G}(t) = \begin{pmatrix} x^{(0)}\left(t, \frac{t-s}{\varepsilon}\right) \\ v^{(0)}\left(t, \frac{t-s}{\varepsilon}\right) \end{pmatrix} + \varepsilon \begin{pmatrix} x^{(1)}\left(t, \frac{t-s}{\varepsilon}\right) \\ v^{(1)}\left(t, \frac{t-s}{\varepsilon}\right) \end{pmatrix} \tag{5.63}$$

where the terms in the expansion are

$$\begin{pmatrix} x^{(0)}(t,\theta) \\ v^{(0)}(t,\theta) \end{pmatrix} = \begin{pmatrix} y^{(0)}(t) \\ \mathcal{N}(\theta)u^{(0)}(t) \end{pmatrix} \tag{5.64}$$

and

$$\begin{pmatrix} x^{(1)}(t,\theta) \\ v^{(1)}(t,\theta) \end{pmatrix} = \begin{pmatrix} y^{(1)}(t) + \mathcal{R}(\theta)u^{(0)}(t) \\ \mathcal{N}(\theta)u^{(1)}(t) + \mathcal{N}(\theta)\left(\int_0^\theta d\sigma - \frac{\theta}{2\pi}\int_0^{2\pi} d\sigma\right)(\mathcal{N}(-\sigma)\mathbf{E}(\sigma)) \end{pmatrix}. \tag{5.65}$$

For the electric field in (5.59), $(y^{(0)}(t), u^{(0)}(t))$ solves the system

$$\frac{dy^{(0)}}{dt} = \begin{pmatrix} u_1^{(0)} \\ 0 \\ 0 \end{pmatrix}, \quad y^{(0)}(s) = x, \tag{5.66a}$$

$$\frac{du^{(0)}}{dt} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \qquad u^{(0)}(s) = v, \tag{5.66b}$$

with (x, v) as the initial condition in (5.58) and $(y^{(1)}(t), u^{(1)}(t))$ satisfies

$$\frac{dy^{(1)}}{dt} = \begin{pmatrix} u_1^{(1)} \\ -1 \\ 0 \end{pmatrix}, \quad y^{(1)}(s) = 0, \tag{5.67a}$$

$$\frac{du^{(1)}}{dt} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad u^{(1)}(s) = 0. \tag{5.67b}$$

The solutions to equations (5.66)-(5.67) are

$$\begin{cases} y^{(0)}(t) = (v_1(t-s) + x_1, x_2, x_3)^T, \\ u^{(0)}(t) = (v_1, v_2, (t-s) + v_3)^T, \end{cases} \tag{5.68}$$

and

$$\begin{cases} y^{(1)}(t) = (0, -(t-s), 0)^T, \\ u^{(1)}(t) = (0, 0, 0)^T. \end{cases} \tag{5.69}$$

By substituting these solutions in (5.64)-(5.65) and incorporating the result into (5.63), we obtain the analytical form of the first-order two-scale approximation $\mathbf{G}(t)$. The approximate solution $\mathbf{G}$ can be written as

$$\mathbf{G}(t) = \mathcal{A} \begin{pmatrix} x \\ v \end{pmatrix} + \mathcal{C} \tag{5.70}$$

where $\mathcal{A}$ is defined as in (5.62) and $\mathcal{C}$ is given by

$$\mathcal{C} = \begin{pmatrix} \varepsilon(t-s)(0, -\cos((t-s)/\varepsilon), \sin((t-s)/\varepsilon))^T \\ (t-s)(0, \sin((t-s)/\varepsilon), \cos((t-s)/\varepsilon))^T \end{pmatrix}. \tag{5.71}$$

Remark 5.8. In the general case where $\mathbf{E}$ has the form

$$\mathbf{E}(\tau) = (E_1, E_2(\tau), E_3(\tau))^T,$$

with $E_1 \in \mathbb{R}$ and 2π-periodic functions E_2, E_3 the solutions of the full model of the macroscopic one keeps similar expressions to those in (5.61) and (5.70). More precisely, the matrix $\mathcal{A}$ remains unchanged, with differences appearing in vectors $\mathcal{B}$ and $\mathcal{C}$ which will include averages in the fast variable against $\sin(\cdot)$ and $\cos(\cdot)$ of the functions E_2 and E_3.

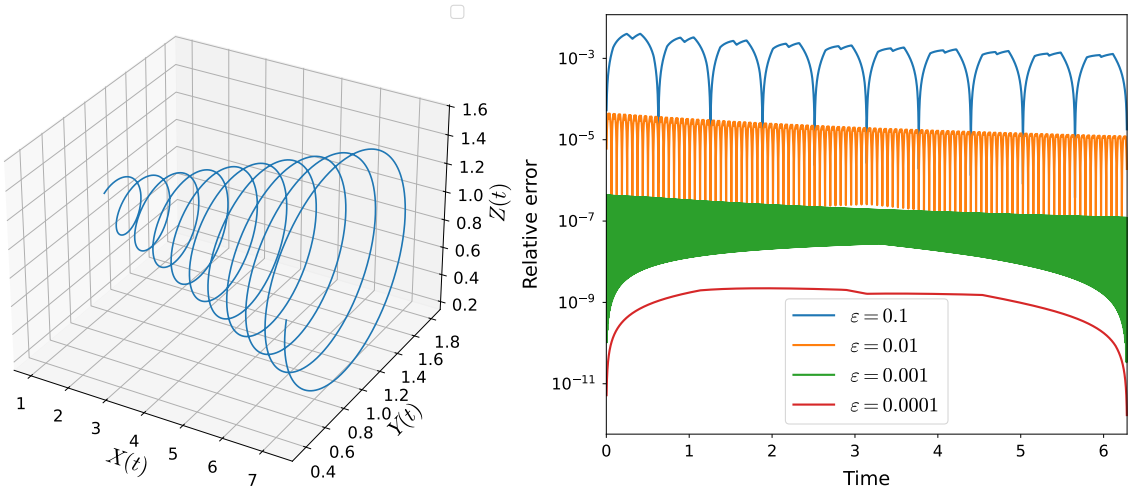

Figure 5.3: On the left, the exact solution of problem (5.47) is visualized as a 3D trajectory of position in the $(X(t), Y(t), Z(t))$ on space with $\varepsilon = 0.1$. On the right, the relative error between the exact solution (5.61) and the macroscopic model solution (5.70) is plotted as a function of time for different values of ε.

We now discuss how accurately approximate this macroscopic model (5.70) to the original problem (5.61). To do this, we plot the relative error

$$\text{Error}(T_n) := \frac{\|\mathcal{G}_n - \mathcal{X}(T_n)\|_1}{\|\mathcal{X}(T_n)\|_1} \tag{5.72}$$

where $\|\cdot\|_1$ stands for the ℓ_1 norm in $\mathbb{R}^6$, $\mathcal{G}_n$ stands for the macroscopic model solution at time T_n and $\mathcal{X}(T_n)$ stands for the original (microscopic) model solution at time T_n. Recall that $\mathcal{G}_n$ and $\mathcal{X}(T_n)$ have analytical forms for this case which are given by the equations (5.61) and (5.70).

Figure 5.3 illustrates the exact solution (5.61) in a three-dimensional phase space $(X(t),\ Y(t),\ Z(t))$ on the left panel with $\varepsilon = 0.1$. The trajectory exhibits a characteristic spiral behaviour, demonstrating the system's evolution over time. Furthermore, the right panel in Figure 5.3 shows the relative error between the exact solution (5.61) and the macroscopic model solution (5.70) as a function of time for different values of $\varepsilon \in \{0.1, 0.01, 0.001, 0.0001\}$, with a final time of $T_{\text{end}} = 2\pi$ and initial conditions are

$$x = (1,\ 1,\ 1)^T,\ v = (1,\ 1,\ 1)^T. \tag{5.73}$$

When $\varepsilon = 0.1$, the relative error of the macroscopic model's approximate solution is on the order of 10^{-2}. However, as ε decreases, the relative error diminishes significantly. For example, when $\varepsilon = 10^{-4}$, the macroscopic model achieves an accuracy with a relative error of approximately 10^{-9}. Moreover, we observe that for larger values of ε, the error exhibits oscillatory behavior because the macroscopic solution captures the slow trend but fails to resolve the fast-scale dynamics. As ε decreases, these oscillations become less pronounced, thereby improving the overall accuracy.

A non-uniform stationary electric field. In this section, we consider an electric field that depends on space rather than time, using the framework presented in [70, Section

3.2]. In this scenario, system (5.47) is given by

$$\frac{dx_\varepsilon}{dt} = v_\varepsilon, \qquad x_\varepsilon(s) = x, \tag{5.74a}$$

$$\frac{dv_\varepsilon}{dt} = \frac{1}{\varepsilon}(v_\varepsilon)^\perp + \mathbf{E}(x_\varepsilon), \qquad v_\varepsilon(s) = v. \tag{5.74b}$$

By following a standard strategy, we can identify an explicit form of a linear operator $\mathbf{E}$ that results in a highly oscillating but bounded solution over time. Consider the electric field given by

$$\mathbf{E}(x) = c\begin{pmatrix} -x_1 \\ x_2/2 \\ x_3/2 \end{pmatrix} \tag{5.75}$$

where $c > 0$ is an arbitrary constant. This system describes the dynamics of a charged particle in an ideal Penning trap [73], assuming the charge and mass of the particle are both set to 1. Under the condition $\varepsilon < \sqrt{1/(2c)}$, the solution of (5.74)-(5.75) is

$$x_\varepsilon(t) = \begin{pmatrix} c_1\cos(\sqrt{c}(t-s)) + c_2\sin(\sqrt{c}(t-s)) \\ a_1\sin(a_\varepsilon(t-s)) - a_2\cos(a_\varepsilon(t-s)) + b_1\sin(b_\varepsilon(t-s)) - b_2\cos(b_\varepsilon(t-s)) \\ a_1\cos(a_\varepsilon(t-s)) + a_2\sin(a_\varepsilon(t-s)) + b_1\cos(b_\varepsilon(t-s)) + b_2\sin(b_\varepsilon(t-s)) \end{pmatrix}, \tag{5.76a}$$

$$v_\varepsilon(t) = \frac{dx_\varepsilon(t)}{dt} \tag{5.76b}$$

where

$$a_\varepsilon = \frac{1+\sqrt{1-2c\varepsilon^2}}{2\varepsilon}, \quad b_\varepsilon = \frac{1-\sqrt{1-2c\varepsilon^2}}{2\varepsilon}, \tag{5.77}$$

and $a_1,\ a_2,\ b_1,\ b_2,\ c_1,\ c_2$ are constants determined by the initial conditions.

Remark 5.9.

1. The constant c in (5.75) describes the geometry of the trap and the voltage between the electrodes, while $1/\varepsilon$ represents the magnetic field's magnitude. The stability condition for a periodic trajectory [73] is

$$\frac{1}{\varepsilon} < \sqrt{2c}. \tag{5.78}$$

   Otherwise, the particle escapes from the trap due to a magnetic field weaker than the electric field, leading to a solution with increasing amplitude over time.

2. The three frequencies $\sqrt{c},\ a_\varepsilon$ and b_ε are noted in the literature [73] as $w_x,\ w_+$ and w_- respectively and they satisfy the relationship

$$w_\pm = \frac{1}{2}(w_{cy} \pm \sqrt{w_{cy}^2 - 2w_x^2}) \tag{5.79}$$

   where w_{cy} is defined in (5.48).

3. It is clear that the motion in the $\overrightarrow{e_1}$ direction is decoupled from the motion in the other two directions. A charged particle in an ideal Penning trap undergoes three independent motions with characteristic frequencies: a modified cyclotron motion (at frequency w_+), the axial motion (at frequency w_x), and the magnetron motion or the $\mathbf{E}\times\mathbf{B}$ drift (at frequency w_-).

4. We have $a_\varepsilon \sim \frac{1}{\varepsilon}$ and $b_\varepsilon \sim \varepsilon$ as $\varepsilon \to 0$. Therefore, the solution (5.76) oscillates in time at three scales $2\pi\varepsilon$, 1 and $2\pi/\varepsilon$. In addition, we can identify initial conditions leading to solutions that oscillate at the desired scales by setting the corresponding coefficients to zero.

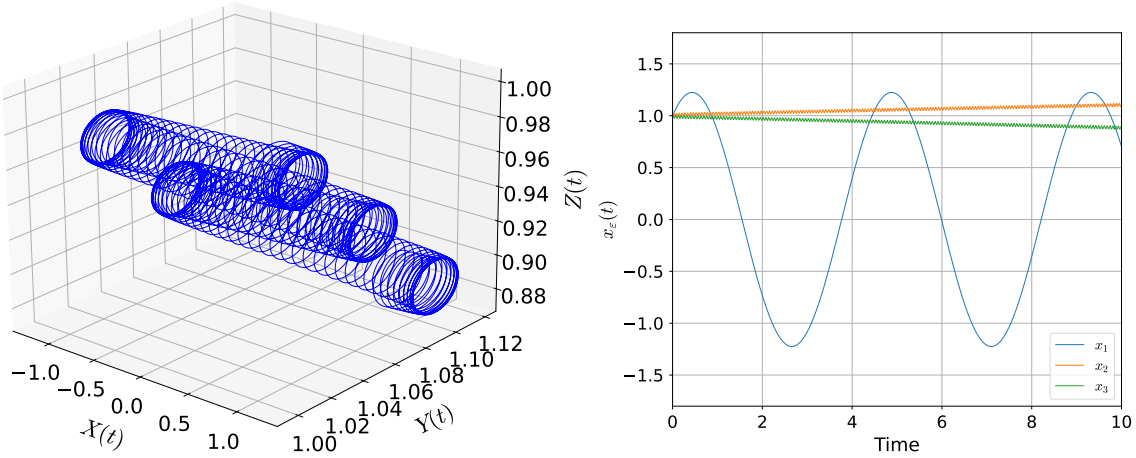

Figure 5.4: On the left, the exact position trajectory is visualized in space as $(X(t), Y(t), Z(t))$. On the right, the exact position as a function of time is shown with final time 10, $\varepsilon = 0.01$ and $c = 2$.

Figure 5.4 illustrates the exact solution for the position trajectory in 3D space (left panel) and its evolution over time (right panel) for $T_{\text{end}} = 10$, $\varepsilon = 0.01$, and $c = 2$. In the left panel, as noted in Remark 5.9, three time scales are clearly observed: $2\pi\varepsilon$, 1, and $2\pi/\varepsilon$. The right panel shows medium oscillations along the x_1-direction and two-scale oscillations along the x_2 and x_3-directions.

Next, we derive the macroscopic model for equation (5.74). Specifically, we apply [70, Theorem 3.3] to obtain a second-order two-scale model of the system (5.74)-(5.75). According to [70, Theorem 3.3], the first-order approximation of the solution to the model (5.74)-(5.75) is given by

$$\mathbf{G}(t) = \begin{pmatrix} x^{(0)}\left(t, \frac{t-s}{\varepsilon}\right) \\ v^{(0)}\left(t, \frac{t-s}{\varepsilon}\right) \end{pmatrix} + \varepsilon \begin{pmatrix} x^{(1)}\left(t, \frac{t-s}{\varepsilon}\right) \\ v^{(1)}\left(t, \frac{t-s}{\varepsilon}\right) \end{pmatrix}, \tag{5.80}$$

where the expansion terms are given by

$$\begin{pmatrix} x^{(0)}(t,\theta) \\ v^{(0)}(t,\theta) \end{pmatrix} = \begin{pmatrix} y^{(0)}(t) \\ \mathcal{N}(\theta)u^0(t) \end{pmatrix} \tag{5.81}$$

and

$$\begin{pmatrix} x^{(1)}(t,\theta) \\ v^{(1)}(t,\theta) \end{pmatrix} = \begin{pmatrix} y^{(1)}(t) + \mathcal{R}(\theta)u^{(0)}(t) \\ \mathcal{N}(\theta)u^{(1)}(t) + \mathcal{R}(\theta)\mathbf{E}(y^{(0)}(t)) \end{pmatrix}. \tag{5.82}$$

For the electric field specified in (5.75), the pair $(y^{(0)}(t), u^{(0)}(t))$ is solution to

$$\frac{dy^{(0)}}{dt} = \begin{pmatrix} u_1^{(0)} \\ 0 \\ 0 \end{pmatrix}, \quad y^{(0)}(s) = x, \tag{5.83a}$$

$$\frac{du^{(0)}}{dt} = \begin{pmatrix} -cy_1^{(0)} \\ 0 \\ 0 \end{pmatrix}, \quad u^{(0)}(s) = v, \tag{5.83b}$$

with (x, v) representing the initial conditions in (5.74) and $(y^{(1)}(t), u^{(1)}(t))$ solves

$$\frac{dy^{(1)}}{dt} = \begin{pmatrix} u_1^{(1)} \\ \frac{c}{2}y_3^{(0)} \\ -\frac{c}{2}y_2^{(0)} \end{pmatrix}, \quad y^{(1)}(s) = 0, \tag{5.84a}$$

$$\frac{du^{(1)}}{dt} = \begin{pmatrix} -cy_1^{(1)} \\ -\frac{c}{2}u_3^{(0)} \\ \frac{c}{2}u_2^{(0)} \end{pmatrix}, \quad u^{(1)}(s) = 0. \tag{5.84b}$$

Equations (5.83)-(5.84) are straightforward to solve yielding

$$\begin{cases} y^{(0)}(t) = (x_1\cos(\sqrt{c}(t-s)) + \frac{v_1}{\sqrt{c}}\sin(\sqrt{c}(t-s)), x_2, x_3)^T, \\ u^{(0)}(t) = (-x_1\sqrt{c}\sin(\sqrt{c}(t-s)) + v_1\cos(\sqrt{c}(t-s)), v_2, v_3)^T, \end{cases} \tag{5.85}$$

and respectively

$$\begin{cases} y^{(1)}(t) = (0, \frac{c}{2}x_3(t-s), -\frac{c}{2}x_2(t-s))^T, \\ u^{(1)}(t) = (0, -\frac{c}{2}v_3(t-s), \frac{c}{2}v_2(t-s))^T. \end{cases} \tag{5.86}$$

Incorporating (5.81)-(5.82) into (5.80) produces

$$\mathbf{G}(t) = \begin{pmatrix} y^{(0)}(t) \\ \mathcal{N}(\frac{t-s}{\varepsilon})u^{(0)}(t) \end{pmatrix} + \varepsilon \begin{pmatrix} y^{(1)}(t) + \mathcal{R}(\frac{t-s}{\varepsilon})u^{(0)}(t) \\ \mathcal{N}(\frac{t-s}{\varepsilon})u^{(1)}(t) + \mathcal{R}(\frac{t-s}{\varepsilon})\mathbf{E}(y^{(0)}(t)) \end{pmatrix}. \tag{5.87}$$

Plugging in the explicit values for $y^{(0)},\ u^{(0)},\ y^{(1)},\ u^{(1)},\ \mathbf{E}$ and the matrices $\mathcal{N}$ and $\mathcal{R}$ in the above formula (5.60) yields the analytical approximation $\mathbf{G}(t)$ for the solution $(x_\varepsilon(t), v_\varepsilon(t))$ when ε is sufficiently small for any time $t \in [s, s + \Delta S]$.

5.4.3 The case of a variable magnetic field

We study the case of a magnetic field with a strong variable component and a constant bounded component, as described in [70, Section 3.4]. Moreover, we focus on scenarios without an electric field. Specifically, we consider equation (5.47) in the form

$$\frac{x_\varepsilon}{dt} = x_\varepsilon, \qquad x_\varepsilon(s) = x, \tag{5.88a}$$

$$\frac{dv_\varepsilon}{dt} = \frac{1}{\varepsilon}\left(v_\varepsilon \times \mathcal{M}(x_\varepsilon)\right) + v_\varepsilon \times \vec{e}_3, \qquad v_\varepsilon(s) = v, \tag{5.88b}$$

where

$$\mathcal{M}(x) = \frac{1}{\sqrt{x_1^2 + x_2^2}} \begin{pmatrix} -x_2 \\ x_1 \\ 0 \end{pmatrix}. \tag{5.89}$$

First, note that the assumption of the $2\pi-$periodicity of the solution $\mathbf{Z}$ to equation (5.50) is satisfied. Unlike the test cases in Section 5.4.2, we do not have an analytic expression for the solution of (5.88).

Next, we detail the two-scale macroscopic model that approximates equation (5.88) as $\varepsilon \to 0$. According to [70, Theorem 3.6], the limit term in the expansion is

$$\begin{pmatrix} x^{(0)}(t,\theta) \\ v^{(0)}(t,\theta) \end{pmatrix} = \begin{pmatrix} y^{(0)}(t) \\ \mathcal{Z}_v(t;\theta, y^{(0)}(t), u^{(0)}(t)) \end{pmatrix} \tag{5.90}$$

where the components of $\mathcal{Z}(t;\theta,x,v) = (\mathcal{Z}_x(t;\theta,x,v), \mathcal{Z}_v(t;\theta,x,v))^T$ are $\mathcal{Z}_x(t;\theta,x,v) = x$ and $\mathcal{Z}_v(t;\theta,x,v) = C(\theta,x)v$, with

$$C(\theta,x) = \begin{pmatrix} \frac{x_1^2\cos(\theta)+x_2^2}{x_1^2+x_2^2} & \frac{x_1x_2(\cos(\theta)-1)}{x_1^2+x_2^2} & -\frac{x_1\sin(\theta)}{\sqrt{x_1^2+x_2^2}} \\ \frac{x_1x_2(\cos(\theta)-1)}{x_1^2+x_2^2} & \frac{x_2^2\cos(\theta)+x_1^2}{x_1^2+x_2^2} & -\frac{x_2\sin(\theta)}{\sqrt{x_1^2+x_2^2}} \\ \frac{x_1\sin(\theta)}{\sqrt{x_1^2+x_2^2}} & \frac{x_2\sin(\theta)}{\sqrt{x_1^2+x_2^2}} & \cos(\theta) \end{pmatrix} \tag{5.91}$$

and where $(y^{(0)}(t), u^{(0)}(t))$ satisfies

$$\frac{dy^{(0)}}{dt} = \bar{A}(y^{(0)})u^{(0)}, \quad y^{(0)}(s) = x, \tag{5.92a}$$

$$\frac{du^{(0)}}{dt} = \bar{\beta}(y^{(0)}, u^{(0)}), \quad u^{(0)}(s) = v, \tag{5.92b}$$

with (x,v) being the initial condition in (5.88) and

$$\bar{A}(y) = \frac{1}{y_1^2+y_2^2}\begin{pmatrix} y_2^2 & -y_1y_2 & 0 \\ -y_1y_2 & y_1^2 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \ \bar{\beta}(y,u) = \begin{pmatrix} \frac{u_2(u_1y_2-u_2y_1)}{y_1^2+y_2^2} \\ \frac{u_1(u_2y_1-u_1y_2)}{y_1^2+y_2^2} \\ 0 \end{pmatrix} \tag{5.93}$$

Thus, to approximate $(x_\varepsilon(t), v_\varepsilon(t))$ when ε is small enough, first solve the system (5.92). Then. the approximation $\mathbf{G}$ is

$$\mathbf{G}(t) = \begin{pmatrix} y^{(0)}(t) \\ C\left(\frac{t-s}{\varepsilon}, y^{(0)}(t)\right) u^{(0)}(t) \end{pmatrix}. \tag{5.94}$$

We solve both models numerically, since no analytical expressions of their solutions are available. More precisely, we solve the system (5.88) by Runge-Kutta-45 [11, pp. 176-179] and the limit model in (5.94) by the explicit Runge-Kutta 4 method [11, pp. 135-141]. We use for the macroscopic model approximation a time step equal to 0.625, whereas for the original dynamics, we use a time step about $2\pi\varepsilon/80$ to accurately solve the cyclotron motion. We consider two initial conditions

$$x = (0,1,1)^T, \quad v = (1,\varepsilon,0)^T \tag{5.95}$$

and

$$x = (1,1,1)^T, \quad v = (1,\varepsilon,0)^T \tag{5.96}$$

We analyze the trajectories of two position particles of the problem (5.88) under different initial conditions with $\varepsilon = 0.01$ and final time 5. The left plot in Figure 5.5 shows the first particle's trajectory with initial condition (5.95), which oscillates on two distinct time

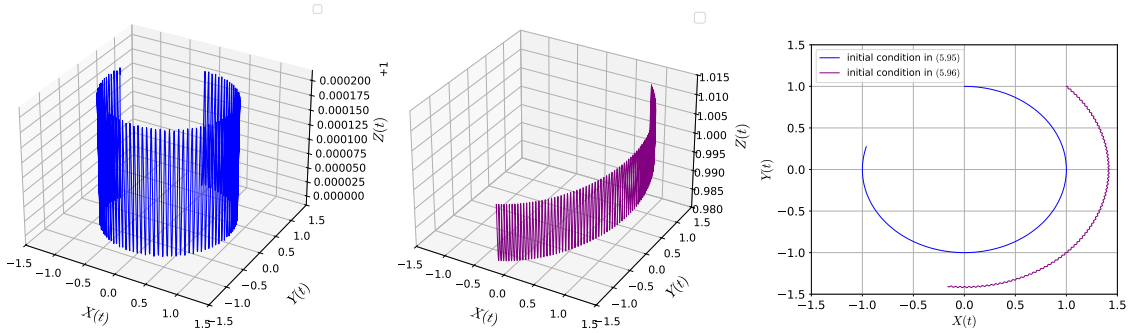

Figure 5.5: The position trajectories until final time 5 of two particles with $\varepsilon = 0.01$: (5.95) at the left panel and (5.96) at the center, following the model in (5.88). The projection of their motion on the perpendicular plane to $\vec{e}_3$ is the right panel. [8, Figure 1.]

scales: a rapid oscillation of order ε and a slower oscillation of order 1. The middle plot depicts the second particle's trajectory with initial condition (5.96), where similar oscillations are observed along with an additional slow linear drift in the $\vec{e}_3$ direction. Finally, the right plot shows the paths of both particles when only their $X(t)$ and $Y(t)$ coordinates are considered. The first particle follows a circular path, while the second particle's path is stretched out in one direction.

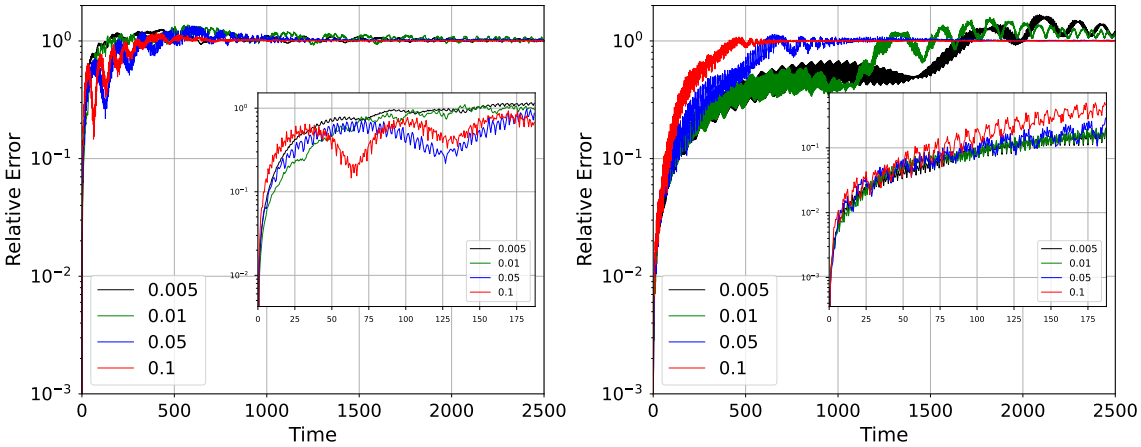

Figure 5.6: Evolution with respect to time of the relative error (5.72) of the numerical approximation of the macroscopic model in (5.94) with respect to the numerical solution of (5.88) with initial condition in (5.96) (at left) and that in (5.95) (at right), for several values of ε [8, Figure 4.].

We plot the relative error (5.72) between numerical approximations of microscopic model (5.88) and macroscopic model (5.94) for several values of $\varepsilon = \{0.1, 0.05, 0.01, 0.005\}$ in Figure 5.6. We can see that the behaviour of the error displays significant difference between two initial conditions (5.96) on the left and (5.95) on the right. More precisely, we observe that at final time $T = 100$, the asymptotic model does not provide a good approximation of the original model when $\varepsilon = \{0.1,\ 0.05\}$ in the case of the initial condition give by (5.96). On the contrary, when the initial condition is given by (5.95), the error is acceptable. However, for both initial conditions, we deduce from Figure 5.6 that the errors

become larger for times exceeding 2000, regardless of the value of ε. Note that our results differ slightly from those presented in [8, Figure 4], as they solve the system (5.88) using the G^4 method of order 4 described in [74].

Chapter 6

Micro-macro multi-level SDC

In Chapter 5, using perturbation analysis, we derived macroscopic models for specific types of singularly and regularly perturbed ODEs. These problems involve a small parameter ε, and as $\varepsilon \to 0$, fast oscillations arise. Note that we refer to these original, ε-dependent problems as the *microscopic models*. Consequently, solving the microscopic problems directly requires impractically small time steps [34]. As shown in Chapter 5, the macroscopic models can effectively approximate the microscopic model in the limit $\varepsilon \to 0$. These models have been widely used in parareal-in-time algorithms for singularly perturbed ODEs [8, 37, 38, 39]; similar ideas have also been applied to stochastic differential equations [41].

For these microscopic problems, standard SDC and even MLSDC treat all scales uniformly. This uniform treatment can lead to inefficiencies when fast (micro) processes force very small time steps, while the slow (macro) dynamics could be resolved with larger ones.

In this Chapter, we introduce the *micro–macro MLSDC (M3LSDC)* method, which combines the concepts presented in Chapters 4 and 5. The M3LSDC approach is designed to separate the fast (micro) dynamics from the slow (macro) ones by splitting the computation into fine (micro) and coarse (macro) levels. The coarse level is less expensive than MLSDC, since it only requires solving the macroscopic models using large time steps.

6.1 Micro-macro MLSDC

In this section, we give a general idea of the M3LSDC method. Assume that we are given a singularly or regularly perturbed system like (5.10) or (5.8). For this problem, we can derive a collocation problem in the form of (2.9)

$$\mathbf{U}_\varepsilon = \mathbf{U}_{\varepsilon,0} + \Delta t \mathbf{Q}_{\text{coll}} \mathbf{F}_\varepsilon(\mathbf{U}_\varepsilon) \tag{6.1}$$

where subscript ε meaning that our variables depend on a small parameter ε.

Furthermore, we can obtain collocation problems for the macroscopic models using the rediscretization approach 4.3.3. The collocation problem for zeroth-order macroscopic model is

$$\mathbf{U}^{(0)} = \mathbf{U}_0^{(0)} + \Delta\nu \mathbf{Q}_{\text{coll}}^{(0)} \mathbf{F}^{(0)}(\mathbf{U}^{(0)}), \tag{6.2}$$

and for the first-order model

$$\mathbf{U}^{(1)} = \mathbf{U}_0^{(1)} + \Delta\nu \mathbf{Q}_{\text{coll}}^{(1)} \mathbf{F}^{(1)}(\mathbf{U}^{(1)}), \tag{6.3}$$

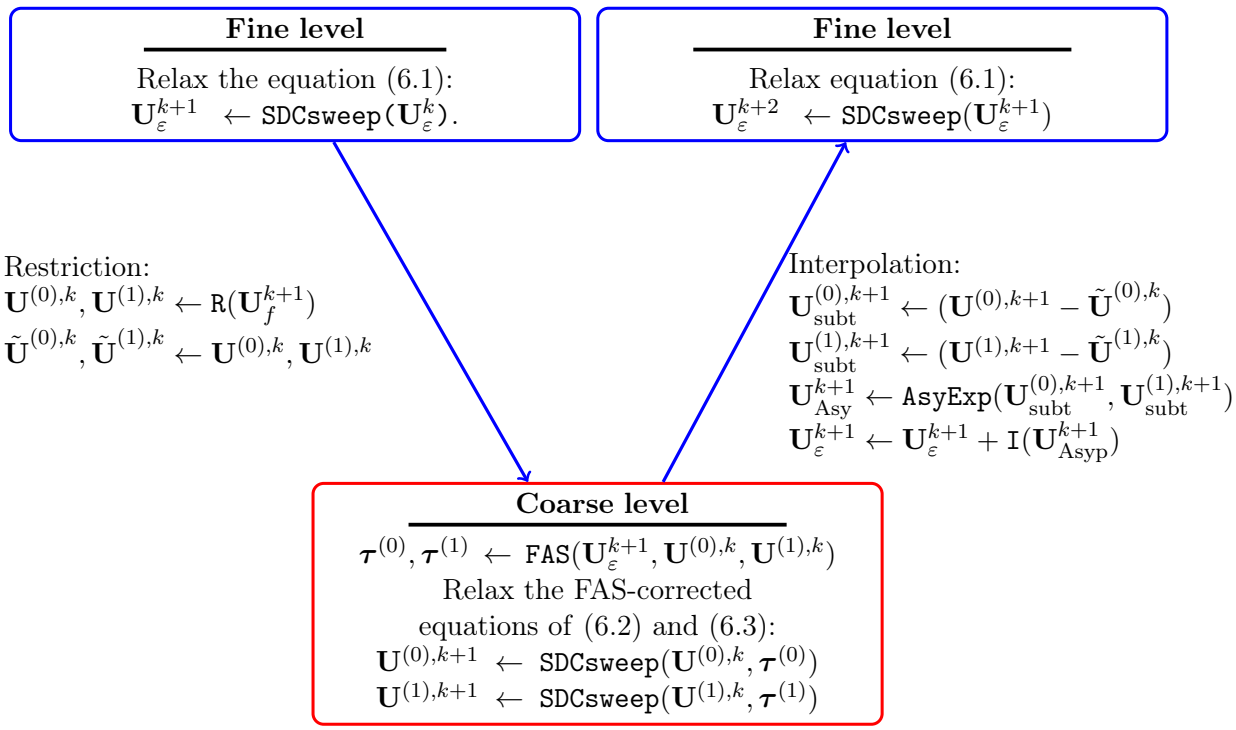

Figure 6.1: Overview of the M3LSDC method algorithm, illustrating the interplay between levels. The algorithm uses restriction and interpolation operators for inter-level communication, and a single SDC sweeps are applied at each level to iteratively refine the solution.

where $\Delta\nu = \nu_{n+1} - \nu_n$ and $\nu(t, \varepsilon)$ is time scale change between the micro and macro models. Also suppose that our asymptotic expansion is give by

$$\mathbf{U}_\varepsilon = \mathbf{U}^{(0)} + \varepsilon \mathbf{U}^{(1)}. \tag{6.4}$$

We assume the suitable restriction R and interpolation I operators are available.

Using all of the given information, we now explain the basic idea of a single M3LSDC iteration (see Figure 6.1). At the fine level, we relax the collocation problem (6.1) using an SDC sweep. Next, we restrict the fine-level solution and compute the initial guess for the zeroth-order (6.2) and first-order (6.3) macroscopic models; these values are stored for later use during interpolation. On the coarse level, we compute the tau correction using the fine-level solution and the restricted values. Then, we relax the FAS-corrected macroscopic models on the coarse level by performing an SDC iteration. To transfer the coarse-level corrections back to the fine grid, we use the asymptotic expansion (6.4) (denoted by `AsyExp`) along with an interpolation operator, and add the resulting corrections to the existing fine-level solution. Finally, we perform another SDC iteration on the fine level to improve accuracy. This cycle is repeated until the desired tolerance on the fine level is achieved.

Remark 6.1. In our numerical examples, we begin on the coarse level. More precisely, we first restrict our initial values and perform a relaxation on the coarse level using an SDC sweep. Then, we interpolate to the fine level and perform another SDC sweep.

6.2 FAS correction for M3LSDC

In the MLSDC method, we relax the microscopic problem at each level while maintaining the same time scale across levels. In contrast, in the M3LSDC method, different time scales and different problems are employed at each level. Consequently, M3LSDC requires a specific type of FAS correction for each microscopic problem to obtain a coarse problem. As usual, we first need to define the fine and coarse-level operators. Using the microscopic collocation problem (6.1), the fine level operator is given by

$$\mathbf{A}_f(\mathbf{U}_\varepsilon) = \mathbf{U}_\varepsilon - \Delta t \mathbf{Q}_{\text{coll}} \mathbf{F}_\varepsilon(\mathbf{U}_\varepsilon). \tag{6.5}$$

To find a coarse-level operator, we combine the two macroscopic collocation problems, (6.2) and (6.3), into a single problem by defining the following matrices

$$\mathbf{U}_c := \begin{pmatrix} \mathbf{U}^{(0)} \\ \mathbf{U}^{(0)} \end{pmatrix}, \ \mathbf{Q}_{\text{coll},c} := \begin{pmatrix} \mathbf{Q}_{\text{coll}}^{(0)} & \mathbf{O} \\ \mathbf{O} & \mathbf{Q}_{\text{coll}}^{(1)} \end{pmatrix}, \ \mathbf{F}_c(\mathbf{U}_c) := \begin{pmatrix} \mathbf{F}^{(0)}(\mathbf{U}^{(0)}) \\ \mathbf{F}^{(1)}(\mathbf{U}^{(1)}) \end{pmatrix} \tag{6.6}$$

where $\mathbf{O}$ is the zeroth-matrix defined in equation (2.9). Using these notations, the coarse level operator is

$$\mathbf{A}_c(\mathbf{U}_c) = \mathbf{U}_c - \Delta\nu \mathbf{Q}_{\text{coll},c} \mathbf{F}_c(\mathbf{U}_c). \tag{6.7}$$

Then, the fine level residual can be computed by

$$\mathbf{r}_f = \mathbf{U}_{\varepsilon,0} - \mathbf{A}_f(\mathbf{U}_\varepsilon) \tag{6.8}$$

Based on argument (4.15), we can directly obtain coarse level problem

$$\mathbf{A}_c(\mathbf{U}_c) = \mathbf{A}_c(R\mathbf{U}_\varepsilon) + R(\mathbf{U}_{\varepsilon,0} - \mathbf{A}_f(\mathbf{U}_\varepsilon)) \tag{6.9}$$

Suppose that the restriction operator is linear. Then, we can simplify expression (6.9), which yields

$$\mathbf{A}_c(\mathbf{U}_c) = \mathbf{A}_c(R\mathbf{U}_\varepsilon) + R\mathbf{U}_{\varepsilon,0} - R\left(\mathbf{A}_f(\mathbf{U}_\varepsilon)\right). \tag{6.10}$$

This leads to

$$\mathbf{A}_c(\mathbf{U}_c) = \mathbf{U}_{0,c} + \boldsymbol{\tau}_c \tag{6.11}$$

where $\mathbf{U}_{0,c} := R\mathbf{U}_{\varepsilon,0}$ and tau correction is given by

$$\boldsymbol{\tau}_c := \mathbf{A}_c(R\mathbf{U}_\varepsilon) - R\mathbf{A}_f(\mathbf{U}_\varepsilon). \tag{6.12}$$

Inserting (6.5) and (6.7) into (6.12) yields

$$\begin{aligned} R\mathbf{U}_\varepsilon - \Delta\nu \mathbf{Q}_{\text{coll},c} \mathbf{F}_c(R\mathbf{U}_\varepsilon) - R\left(\mathbf{U}_\varepsilon - \Delta t \mathbf{Q}_{\text{coll}} \mathbf{F}_\varepsilon(\mathbf{U}_\varepsilon)\right) = \\ R\mathbf{U}_\varepsilon - \Delta\nu \mathbf{Q}_{\text{coll},c} \mathbf{F}_c\left(R\mathbf{U}_\varepsilon\right) - R\mathbf{U}_\varepsilon + \Delta t R \mathbf{Q}_{\text{coll}} \mathbf{F}_\varepsilon(\mathbf{U}_\varepsilon). \end{aligned} \tag{6.13}$$

So, the tau correction can be computed by

$$\boldsymbol{\tau}_c = \Delta t R \mathbf{Q}_{\text{coll}} \mathbf{F}_\varepsilon(\mathbf{U}_\varepsilon) - \Delta\nu \mathbf{Q}_{\text{coll},c} \mathbf{F}_c(R\mathbf{U}_\varepsilon). \tag{6.14}$$

Then, the FAS-corrected coarse level problem turns into

$$\mathbf{U}_c - \Delta\nu \mathbf{Q}_{\text{coll},c} \mathbf{F}_c(\mathbf{U}_c) = \mathbf{U}_{0,c} + \boldsymbol{\tau}_c. \tag{6.15}$$

Remark 6.2. The main difference between the FAS corrections in MLSDC and M3LSDC lies in the alignment of time scales and the formulation of the underlying problems at different levels.

In MLSDC, the same time scale is used across all levels. This allows the FAS correction to be computed by simply restricting the fine-level residual onto the coarse level, as given by (4.21). Because both levels share a common structure and problem formulation, this correction straightforwardly transfers fine-level information to the coarse level.

In contrast, M3LSDC uses different time scales and often different problem formulations at each level. This discrepancy means the correction must be tailored to each coarse problem, considering both the differences in time discretization and the underlying operator definitions. This results in a more complex FAS correction that ensures the coarse-level approximation accurately reflects the dynamics of the fine level.

6.3 Transfer operators for M3LSDC

In MLSDC, the transfer operators (restriction and interpolation) are typically based on standard multilevel techniques. A polynomial or spectral interpolation operator transfers the solution between levels.

In contrast, the M3LSDC method enhances these operators by incorporating macroscopic modeling. The restriction operator is designed to filter out fast-scale oscillations by averaging the fine-level solution, effectively capturing the slow, macroscopic behavior governed by a small parameter (e.g., ε). This ensures that the coarse-level representation is consistent with the asymptotic expansion of the solution.

The interpolation process is augmented with corrections derived from the asymptotic expansion (e.g., including terms like $\mathbf{U}^{(0)}+\varepsilon\mathbf{U}^{(1)}$, or even higher-order contributions). This allows the fine-level solution to be recovered more accurately by incorporating macroscopic information.

Assume that the fine level has M_f collocation nodes and the coarse level has M_c nodes. Our goal is to define transfer operators that connect the fine-level solution with its macroscopic representation. In particular, we wish to construct a restriction operator

$$R : \mathbf{U}_\varepsilon \mapsto \begin{pmatrix} \mathbf{U}^{(0)} \\ \mathbf{U}^{(1)} \end{pmatrix}, \tag{6.16}$$

where $R : \mathbb{R}^{M_f} \to \mathbb{R}^{2M_c}$. Here, $\mathbf{U}_\varepsilon$ is the solution computed on the fine grid, while $\mathbf{U}^{(0)}$ and $\mathbf{U}^{(1)}$ represent the solution components corresponding to the zeroth-order and the first-order macroscopic models, respectively.

Similarly, the interpolation operator is defined as

$$I : \begin{pmatrix} \mathbf{U}^{(0)} \\ \mathbf{U}^{(1)} \end{pmatrix} \mapsto \mathbf{U}_\varepsilon, \tag{6.17}$$

with $I : \mathbb{R}^{2M_c} \to \mathbb{R}^{M_f}$. In the remainder of this thesis, we discuss various choices and implementations for the restriction operators

6.3.1 Injection

The first candidate is the usual *injection* restriction operator

$$R : \mathbf{U}_\varepsilon \mapsto \begin{pmatrix} \mathbf{U}_\varepsilon \\ \mathbf{0} \end{pmatrix} \tag{6.18}$$

where $\mathbf{0}$ is a vector with all entries equal to zero. The fine-scale (micro) solution $\mathbf{U}_\varepsilon$ is directly transferred to the coarse level, with the additional components set to zero. However, the problem is setting the additional coarse components to zero may not accurately represent the true macro-scale behavior, potentially leading to a mismatch between the microscopic and macroscopic models.

6.3.2 Averaging

The second candidate, the *averaging* operator, computes the coarse value as a weighted average of fine-scale values. It is commonly used in the parareal-in-time methods [75] and multiscale methods [34, 76]. To apply this averaging operator, first restrict fine values $\mathbf{U}_{\varepsilon,f} \in \mathbb{R}^{M_f}$ to coarse one using the barycentric restriction operator in Section 4.3.3 which leads to a new coarse value $\mathbf{U}_{\varepsilon,c} \in \mathbb{R}^{M_c}$. Then, we can apply the following averaging operator to separate this coarse restricted solution for macroscopic models

$$R : \mathbf{U}_{\varepsilon,c} :\mapsto \begin{pmatrix} \mathbf{Q}_{\text{coll},c}\mathbf{U}_{\varepsilon,c} \\ \left(\mathbf{U}_{\varepsilon,c} - \mathbf{U}^{(0)}\right)/\varepsilon \end{pmatrix} \tag{6.19}$$

where $\mathbf{Q}_{\text{coll},c}$ is the collocation matrix on coarse level. First, we compute the average of the restricted values using

$$\mathbf{Q}_{\text{coll},c}\mathbf{U}_{\varepsilon,c},$$

which equals $\mathbf{U}^{(0)}$. Next, using an asymptotic expansion, we fill in the additional values by computing

$$\frac{\mathbf{U}_{\varepsilon,c} - \mathbf{U}^{(0)}}{\varepsilon}.$$

The averaging operator removes high-frequency oscillations, leading to a cleaner coarse representation [34].

Remark 6.3. In our numerical examples, we only use the averaging operator; we have not yet implemented other restriction operators.

6.3.3 Constrained Optimization

An alternative approach is to formulate the restriction as a constrained optimization problem. In this approach, the coarse-level representation is obtained by solving an optimization problem that seeks the best approximation of the fine-scale solution [77]. This can be formally stated as

$$R : \mathbf{U}_\varepsilon \mapsto \min_{\mathbf{U}^{(0)},\mathbf{U}^{(1)}} \left(\|\mathbf{U}^{(0)} - \mathbf{U}_0^{(0)}\|_2 + \|\mathbf{U}^{(1)} - \mathbf{U}_0^{(1)}\|_2 \right) \text{ subject to (6.4)} \tag{6.20}$$

where $\mathbf{U}_0^{(0)}$ and $\mathbf{U}_0^{(1)}$ are initial guesses for the coarse level collocation problems (6.2) and (6.3) and $\|\cdot\|_2$ is the Euclidean norm. Using constrained optimization, we optimize the objective function $\|\mathbf{U}^{(0)}-\mathbf{U}_0^{(0)}\|_2+\|\mathbf{U}^{(1)}-\mathbf{U}_0^{(1)}\|_2$ with respect to $\mathbf{U}^{(0)}$ and $\mathbf{U}^{(1)}$, subject to the constraints in (6.4). By explicitly enforcing constraints, the optimized restriction operator can yield a coarse-level solution that more accurately represents the essential features of the fine-scale data. However, solving a constrained optimization problem at every restriction step introduces additional computational overhead compared to more straightforward methods.

Remark 6.4.
- Note that this restriction operator is nonlinear, hence, we cannot use directly FAS-corrected coarse problem (6.11) and tau correction (6.12) formula. So, we need to relax coarse problem (6.9) without finding tau correction.
- For the objective function, we can also choose different norms. However, we have not yet tested this restriction operator.

6.3.4 Interpolation Operator

We have introduced restriction operators, but we now provide additional details on the corresponding interpolation operators (6.17) that map the coarse-level solution back to the fine grid.

To reconstruct the fine-level solution, we use the asymptotic expansion given in (6.4), which yields

$$I : \begin{pmatrix} \mathbf{U}^{(0)} \\ \mathbf{U}^{(1)} \end{pmatrix} \mapsto \mathbf{U}^{(0)} + \varepsilon \mathbf{U}^{(1)}. \tag{6.21}$$

Remark 6.5.
1. The interpolation operator leverages the asymptotic structure of the solution, ensuring that both the dominant (zeroth-order) behavior and the first-order corrections (scaled by ε) are accurately represented on the fine grid.
2. This approach is particularly effective when ε is small, as the asymptotic expansion provides a reliable approximation of the true solution.

6.4 Coarse operator using the analytical solution of macroscopic models

Often, the solution of the macroscopic model can be computed analytically. In such cases, we can leverage this analytical solution to derive an improved FAS-corrected coarse operator.

Assume that we have an exact solution $\mathbf{V}_c$ of the coarse problem,

$$\mathbf{A}_c(\mathbf{V}_c) = \mathbf{U}_{0,c}, \tag{6.22}$$

where $\mathbf{A}_c$ denotes the coarse operator and $\mathbf{U}_{0,c}$ represents the coarse-level data.

Now, let $\mathbf{U}_c$ be the solution of the FAS-corrected coarse problem,

$$\mathbf{A}_c(\mathbf{U}_c) = \mathbf{U}_{0,c} + \boldsymbol{\tau}_c, \tag{6.23}$$

with $\boldsymbol{\tau}_c$ being the tau correction term.

Since $\mathbf{V}_c$ satisfies (6.22), we can rewrite (6.23) by subtracting the equation for $\mathbf{V}_c$:

$$\mathbf{A}_c(\mathbf{U}_c) - \mathbf{A}_c(\mathbf{V}_c) = \tau_c. \tag{6.24}$$

Assuming linearity of $\mathbf{A}_c$ (macroscopic models are usually linear [6]), this becomes

$$\mathbf{A}_c(\mathbf{U}_c - \mathbf{V}_c) = \boldsymbol{\tau}_c. \tag{6.25}$$

Define the difference between the FAS-corrected solution and the exact solution as

$$\Delta\mathbf{U}_c := \mathbf{U}_c - \mathbf{V}_c. \tag{6.26}$$

Then, the above equation can be written as

$$\mathbf{A}_c(\Delta\mathbf{U}_c) = \boldsymbol{\tau}_c. \tag{6.27}$$

If $\mathbf{A}_c$ is invertible, we can solve for $\Delta\mathbf{U}_c$:

$$\Delta\mathbf{U}_c = \mathbf{A}_c^{-1}\tau_c. \tag{6.28}$$

Thus, the updated coarse-level solution is given by

$$\mathbf{U}_c = \mathbf{V}_c + \mathbf{A}_c^{-1}\tau_c. \tag{6.29}$$

Remark 6.6. Rather than solving (6.23) directly, one can relax the difference equation (6.27) using a SDC sweep. After this relaxation, the updated solution is computed as in (6.28).

6.5 M3LSDC Algorithm: Step-by-Step Description

The M3LSDC algorithm proceeds as follows. The initial conditions x_0 and v_0 are distributed across all collocation nodes on the fine level, so that the initial solution is given by

$$\mathbf{U}_f^0 = [x_0, \ldots, x_0,\ v_0, \ldots, v_0]. \tag{6.30}$$

A single M3LSDC iteration consists of the following steps:

1. **Restriction from Fine to Coarse:**
   - **Barycentric Lagrange Interpolation:** First, we apply the barycentric Lagrange interpolation operator (see Section 4.3.3) to reduce the number of quadrature nodes on the coarse level.
   - **Macroscopic Model Initialization:** Next, we use an additional restriction operator (described in Section 6.3) to generate an initial guess for the macroscopic models and compute tau correction (see Section 6.2).
2. **SDC Sweep on the Macroscopic Models:** With the initial guess from restricted fine solution, we perform a SDC sweep on the FAS-corrected macroscopic model collocation problem.
3. **Interpolation from Coarse to Fine:**

- **Asymptotic Expansion:** The coarse solution is first combined using the asymptotic expansion (typically in the form $\mathbf{U}^{(0)} + \varepsilon\mathbf{U}^{(1)}$) to form an approximation of the fine-level solution.
- **Barycentric Interpolation:** Then, the barycentric interpolation operator is applied to map the coarse solution back onto the fine grid. Compute the error with old fine level solution and add to it.

4. **Update Fine-Level Solution:** The result from the interpolation step provides the updated fine-level solution. Perform one SDC sweep with the updated fine level solution. This completes a single M3LSDC iteration.

The above procedure is repeated $(K+1)$ times to obtain the solution after the $(K+1)$th M3LSDC iteration. This process is summarized in Algorithm 7.

Algorithm 7: Kth M3LSDC iteration

Data: Initial condition $\mathbf{U}_0$, K-number of iterations and M number of quadrature nodes

Result: Solution after $(K+1)$th M3LSDC iteration $\mathbf{U}_f^{K+1}$

`# Set initial guess spread`

$\mathbf{U}_f^0 \leftarrow \mathbf{U}_0$

for $k = 0, \ldots, K$ **do**

`# Cycle from fine to coarse`

`# Restrict and save restriction (used later during interpolation)`

$\mathbf{U}_c^k \leftarrow$ `Restrict2_Nodes`$(\mathbf{U}_f^k)$ `# Using operator in Section 4.3.3`

$\mathbf{U}_c^{(0),k}, \mathbf{U}_c^{(1),k} \leftarrow$ `Restrict2_Rmodel`$(\mathbf{U}_c^k)$ `# Using one of the operators in Section 6.3`

$\tilde{\mathbf{U}}_c^{(0),k}, \tilde{\mathbf{U}}_c^{(1),k} \leftarrow \mathbf{U}_c^{(0),k}, \mathbf{U}_c^{(1),k}$ `# Store the values`

`# Compute FAS correction and sweep`

$\boldsymbol{\tau}_c \leftarrow$ `FAS`$(\mathbf{U}_c^{(0),k}, \mathbf{U}_c^{(0),k})$

$\mathbf{U}_c^{(0),k+1}, \mathbf{U}_c^{(1),k+1} \leftarrow$ `SDCsweep`$(\mathbf{U}_c^{(0),k}, \mathbf{U}_c^{(0),k}, \boldsymbol{\tau}_c)$

`# Cycle from coarse to fine`

`# Use asymptotic expansion`

$\mathbf{U}_c^{k+1} \leftarrow$ `AsymExpan`$(\mathbf{U}_c^{(0),k+1} - \tilde{\mathbf{U}}_c^{(0),k}, \mathbf{U}_c^{(1),k+1} - \tilde{\mathbf{U}}_c^{(1),k})$

`# Interpolate coarse correction`

$\mathbf{U}_f^{k+1} \leftarrow \mathbf{U}_f^k +$ `Interpolate_nodes`$(\mathbf{U}_c^{k+1})$

`# Perform fine sweep`

$\mathbf{U}_f^{k+1} \leftarrow$ `SDCsweep`$(\mathbf{U}_f^{k+1})$

end

6.6 Numerical example

In this section, we compare the residual of SDC, MLSDC, and M3LSDC methods. In the M3LSDC method, the coarse-level problem can be solved using either a zeroth-order macroscopic model or first-order reduced models. More precisely,

- **Zeroth-order model ($\mathcal{O}(\varepsilon^1)$: $\mathbf{U}^{(0)}$):** This corresponds to solving only the zeroth-order model (6.2) on the coarse level.
- **First-order model ($\mathcal{O}(\varepsilon^2)$: $\mathbf{U}^{(0)} + \varepsilon\mathbf{U}^{(1)}$):** This involves solving both the first-order (6.2) and first-order (6.3) models on the coarse level, followed by applying the asymptotic expansion (6.4).

For the Duffing equation (5.17), We use the following parameters: $\omega = 1.0$, $b = 1.0$, with initial conditions $x_0 = 2.0$ and $v_0 = 0.0$, and a time step of 0.5. Figure 6.2 shows the infinity norm of the residual as a function of the iteration count. The numbers of Gauss quadrature nodes used at the fine and coarse levels are $M_f = 5$ and $M_c = 3, 4$, respectively, for different values of ε (0.1 and 0.001). For the SDC iteration, we use 5 Gauss quadrature nodes, and the averaging restriction operator (6.3.2) is applied in the M3LSDC iteration.

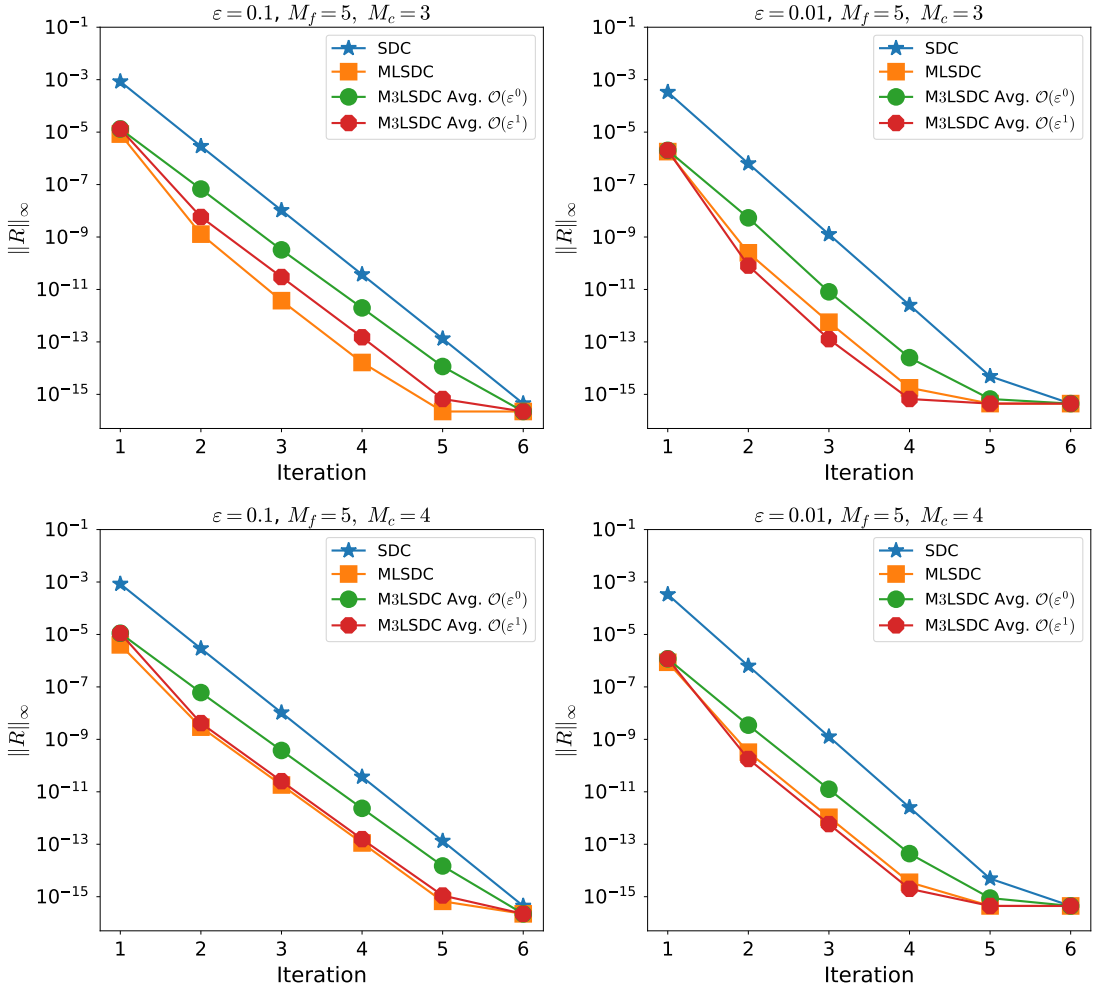

Figure 6.2: Infinity norm of residual with respect to the number of iteration $K = 6$ with different values of ε. SDC uses 5 Gauss quadrature nodes and the number of Gauss quadrature nodes on the fine and coarse levels are $M_f = 5$ and $M_c = 3,\ 4$ respectively for both MLSDC and M3LSDC. Averaging restriction operator is used for the M3SDC method.

For $\varepsilon = 0.01$ (right panel), the M3LSDC residual computed using the first-order model ($\mathcal{O}(\varepsilon^2)$) achieves smaller residuals after the same number of iterations compared to the SDC method and the M3LSDC residual as well as M3LSDC residual with the zeroth-order model ($\mathcal{O}(\varepsilon^1)$) on the coarse level. In contrast, for the larger value $\varepsilon = 0.1$ (left panel), the M3LSDC residual with the first-order model is nearly identical to the MLSDC residual at the same iteration count. Thus, employing macroscopic models on the coarse level yields residuals that are comparable to those obtained with the MLSDC method.

Chapter 7

Conclusion and outlook

We presented spectral deferred correction methods (SDC) for general second-order problems. We provide a theoretical analysis of spectral deferred corrections (SDC) applied to second-order problems. Using the damped harmonic oscillator as a test problem, we investigate the convergence and stability of SDC, comparing it against a collocation method based on Picard iteration and a Runge-Kutta-Nyström-4 (RKN-4) method. These results show that SDC converges over a much wider region than Picard and RKN-4 in cases with very strong damping.

The main theoretical result of this thesis is a proof that every SDC iteration increases the global convergence order by one when the force depends on velocity and by two when the force is independent of velocity. We also show that the order of the local position error is one higher than that of the local velocity error. Our theoretical predictions are validated through numerical examples for the Penning trap benchmark.

Moreover, we introduce the multi-level SDC (MLSDC) method for second-order problems. A FAS correction is used to increase the accuracy on the coarse level. The main motivation of MLSDC is that it shifts computational work from the fine level to a less expensive coarse level, thereby reducing the number of costly fine-level SDC sweeps. To make coarse-level sweeps computationally cheap, we propose two coarsening strategies: (1) using fewer quadrature nodes on the coarse level and (2) using macroscopic models as the coarse problem. Numerical results for the former strategy demonstrate that MLSDC achieves a smaller residual than SDC for the same number of iterations. Furthermore, we observe that the global convergence order of MLSDC increases by two when the force depends on velocity and by one when it is independent of velocity. However, we do not have a theoretical proof for this behavior; the convergence analysis of the MLSDC method for second-order problems is left for future work.

Furthermore, we introduce the micro–macro MLSDC (M3LSDC) method for second-order, multiscale problems. On the fine level, we solve the full microscopic model, which captures all detailed dynamics including fast oscillatory behavior. On the coarse level, we solve a macroscopic model that approximates the slow dynamics by filtering out the fast scales. Using larger time steps on the coarse level significantly reduces the overall computational cost. In contrast, with MLSDC, increasing the coarse time step size requires combining multiple fine steps into one large coarse step, which results in a higher computational cost. The main difficulty in M3LSDC is that the problem involves multiple time scales, each with distinct dynamic characteristics. This means that a single

FAS correction cannot be universally applied because each time scale may require its own tailored correction. Moreover, the restriction operator must be designed to accurately capture and transfer the dynamics across these varying time scales. We derive a FAS correction that couples the two levels and propose possible choices for restriction operators for M3LSDC. Our numerical experiments for the Duffing equation demonstrate that the residual of M3LSDC performs comparably to, or even outperforms, the residual of standard SDC and MLSDC.

Future work will focus on refining coarsening strategies, optimizing transfer operators, and extending M3LSDC to more complex and higher-dimensional systems.

Bibliography

[1] I. Akramov, S. Götschel, M. Minion, D. Ruprecht, and R. Speck, "Spectral deferred correction methods for second-order problems," *SIAM Journal on Scientific Computing*, vol. 46, p. A1690–A1713, May 2024. The Creative Commons Attribution 4.0 International (CC BY) License.

[2] W. Commons, "File:penning trap1.svg — wikimedia commons, the free media repository," 2025. [Online; accessed 6-February-2025]. Licensed under CC0 1.0 Universal Public Domain Dedication.

[3] M. Winkel, R. Speck, and D. Ruprecht, "A high-order Boris integrator," *Journal of Computational Physics*, vol. 295, pp. 456–474, 2015. © 2015 Elsevier Inc. All rights reserved.

[4] W. L. Briggs, V. E. Henson, and S. F. McCormick, *A Multigrid Tutorial, Second Edition*. Society for Industrial and Applied Mathematics, second ed., 2000.

[5] OpenAI, "Chatgpt," 2024. Accessed: October 10, 2024.

[6] A. L. Smirnov, S. Bauer, S. Filippov, P. Tovstik, and R. Vaillancourt, *Asymptotic Methods in Mechanics of Solids*, vol. 167. 05 2015.

[7] R. Klein, S. Vater, E. Paeschke, and D. Ruprecht, *Multiple scales methods in meteorology*, pp. 127–196. Vienna: Springer Vienna, 2010.

[8] L. Grigori, S. A. Hirstoaga, V.-T. Nguyen, and J. Salomon, "Reduced model-based parareal simulations of oscillatory singularly perturbed ordinary differential equations," *Journal of Computational Physics*, vol. 436, p. 110282, 2021.

[9] E. Hairer, C. Lubich, and G. Wanner, *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*. Springer Verlag Berlin Heidelberg New York, 2002.

[10] E. Hairer, C. Lubich, and G. Wanner, "Geometric numerical integration illustrated by the Störmer–Verlet method," *Acta Numerica*, vol. 12, p. 399–450, 2003.

[11] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff problems*. Springer-Verlag Berlin Heidelberg, 2nd ed., 1993.

[12] S. Blanes and P. Moan, "Practical symplectic partitioned Runge–Kutta and Runge–Kutta–Nyström methods," *Journal of Computational and Applied Mathematics*, vol. 142, no. 2, p. 313–330, 2002.

[13] S. O. Imoni, F. O. Otunta, and T. R. Ramamohan, "Embedded implicit Runge–Kutta Nyström method for solving second-order differential equations," *International Journal of Computer Mathematics*, vol. 83, no. 11, p. 777–784, 2006.

[14] P. J. van der Houwen and B. P. Sommeijer, "Explicit Runge–Kutta (–Nyström) methods with reduced phase errors for computing oscillating solutions," *SIAM Journal on Numerical Analysis*, vol. 24, no. 3, p. 595–617, 1987.

[15] P. J. Van der Houwen and B. P. Sommeijer, "Diagonally implicit Runge–Kutta–Nyström methods for oscillatory problems," *SIAM Journal on Numerical Analysis*, vol. 26, no. 2, p. 414–429, 1989.

[16] X. Bai and J. L. Junkins, "Modified Chebyshev-Picard iteration methods for orbit propagation," *The Journal of the Astronautical Sciences*, vol. 58, pp. 583–613, 2011.

[17] A. Dutt, L. Greengard, and V. Rokhlin, "Spectral deferred correction methods for ordinary differential equations," *BIT Numerical Mathematics*, vol. 40, no. 2, pp. 241–266, 2000.

[18] A. Christlieb, B. W. Ong, and J.-M. Qiu, "Integral deferred correction methods constructed with high order Runge-Kutta integrators," *Mathematics of Computation*, vol. 79, pp. 761–783, 2010.

[19] T. Hagstrom and R. Zhou, "On the spectral deferred correction of splitting methods for initial value problems," *Communications in Applied Mathematics and Computational Science*, vol. 1, no. 1, pp. 169–205, 2006.

[20] A. C. Hansen and J. Strain, "On the order of deferred correction," *Applied Numerical Mathematics*, vol. 61, pp. 961–973, 2011.

[21] J. Huang, J. Jia, and M. Minion, "Accelerating the convergence of spectral deferred correction methods," *Journal of Computational Physics*, vol. 214, no. 2, pp. 633 – 656, 2006.

[22] M. L. Minion, "Semi-implicit spectral deferred correction methods for ordinary differential equations," *Communications in Mathematical Sciences*, vol. 1, no. 3, pp. 471–500, 2003.

[23] M. L. Minion, "Semi-implicit projection methods for incompressible flow based on spectral deferred corrections," *Applied Numerical Mathematics*, vol. 48, no. 3–4, pp. 369 – 387, 2004. Workshop on Innovative Time Integrators for PDEs.

[24] E. L. Bouzarth and M. L. Minion, "A multirate time integrator for regularized Stokeslets," *Journal of Computational Physics*, vol. 229, no. 11, pp. 4208 – 4224, 2010.

[25] Q. Shen and B. Kochunas, "High-order accurate solutions of the point kinetics equations with the spectral deferred correction method," *Nuclear Science and Engineering*, 2023.

[26] R. Speck, D. Ruprecht, M. Emmett, M. Minion, M. Bolten, and R. Krause, "A multi-level spectral deferred correction method," *BIT Numerical Mathematics*, vol. 55, p. 843–867, Aug. 2014.

[27] F. P. Hamon, M. Schreiber, and M. L. Minion, "Multi-level spectral deferred corrections scheme for the shallow water equations on the rotating sphere," *Journal of Computational Physics*, vol. 376, pp. 435–454, 2019.

[28] G. Kremling and R. Speck, "Convergence of multilevel spectral deferred corrections," *Communications in Applied Mathematics and Computational Science*, vol. 16, pp. 227–265, 2021.

[29] J. P. Boris *et al.*, "Relativistic plasma simulation-optimization of a hybrid code," in *Proc. Fourth Conf. Num. Sim. Plasmas*, pp. 3–67, 1970.

[30] D. J. Griffiths, *Introduction to electrodynamics*. Cambridge University Press, 2023.

[31] K. Tretiak and D. Ruprecht, "An arbitrary order time-stepping algorithm for tracking particles in inhomogeneous magnetic fields," *Journal of Computational Physics: X*, vol. 4, p. 100036, 2019.

[32] K. Tretiak, J. Buchanan, R. Akers, and D. Ruprecht, "Performance of the BGSDC integrator for computing fast ion trajectories in nuclear fusion reactors," *Computer Physics Communications*, vol. 264, p. 107876, 2021.

[33] K. Smedt, D. Ruprecht, J. Niesen, S. Tobias, and J. Nättilä, "New applications for the Boris spectral deferred correction algorithm for plasma simulations," *Applied Mathematics and Computation*, vol. 442, p. 127706, 2023.

[34] G. A. Pavliotis and A. Stuart, *Multiscale methods: averaging and homogenization*, vol. 53. Springer Science & Business Media, 2008.

[35] A. Abdulle, E. Weinan, B. Engquist, and E. Vanden-Eijnden, "The heterogeneous multiscale method," *Acta Numerica*, vol. 21, pp. 1–87, 2012.

[36] W. Ee, B. Engquist, X. Li, W. Ren, and E. Vanden-Eijnden, "Heterogeneous multiscale methods: A review," *Communications in Computational Physics*, vol. 2, 06 2007.

[37] S. Hirstoaga, "A first-order reduced model for a highly oscillating differential equation with application in penning traps," *SIAM Journal on Scientific Computing*, pp. S225–S245, 08 2024.

[38] F. Legoll, T. Lelièvre, and G. Samaey, "A micro-macro parareal algorithm: Application to singularly perturbed ordinary differential equations," *SIAM Journal on Scientific Computing*, vol. 35, no. 4, pp. A1951–A1986, 2013.

[39] B. Adel, L. Boudin, and S. Kaber, "Parallel in time algorithms with reduction methods for solving chemical kinetics," *Communications in Applied Mathematics and Computational Science*, vol. 5, 09 2010.

[40] Y. Maday, "Parareal in time algorithm for kinetic systems based on model reduction," *High-dimensional partial differential equations in science and engineering*, vol. 41, pp. 183–194, 2007.

[41] I. Bossuyt, S. Vandewalle, and G. Samaey, "Monte-carlo/moments micro-macro parareal method for unimodal and bimodal scalar mckean-vlasov sdes," 2024.

[42] C. Chicone, *Ordinary Differential Equations with Applications*, vol. 34. 2006.

[43] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*. Society for Industrial and Applied Mathematics, 1995.

[44] L. Verlet, "Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules," *Phys. Rev.*, vol. 159, pp. 98–, 1967.

[45] P. Agarwal, J. Mohamed, and B. Samet, *Fixed Point Theory in Metric Spaces: Recent Advances and Applications*. 2018.

[46] V. I. Arnold, *Ordinary differential equations*. Springer Science & Business Media, 1992.

[47] K. Atkinson, *An introduction to numerical analysis*. John wiley & sons, 1991.

[48] I. Grant and W. Phillips, *Electromagnetism*. Manchester Physics Series, Wiley, 2013.

[49] C. Birdsall and A. Langdon, *Plasma Physics Via Computer Simulation*. Adam Hilger series on plasma physics, McGraw-Hill, 1985.

[50] T. Tang, H. Xie, and X. Yin, "High-order convergence of spectral deferred correction methods on general quadrature nodes," *Journal of Scientific Computing*, vol. 56, no. 1, pp. 1–13, 2013.

[51] W. Gautschi, "The asymptotic behaviour of powers of matrices. II.," *Duke Mathematical Journal*, vol. 20, no. 3, pp. 375 – 379, 1953.

[52] J. Fregin, E. Bronasco, G. Vilmart, D. Ruprecht, and H. Ranocha, "Spectral Deferred Corrections in the framework of Runge-Kutta methods." Unpublished, 2025.

[53] G. Caklovic, "The infinity norm bounds and characteristic polynomial for high order RK matrices," *ArXiv*, vol. abs/2203.04086, 2022.

[54] D. Ruprecht and R. Speck, "Spectral deferred corrections with fast-wave slow-wave splitting," *SIAM Journal on Scientific Computing*, vol. 38, no. 4, pp. A2535–A2557, 2016.

[55] M. F. Causley and D. C. Seal, "On the convergence of spectral deferred correction methods," *Communications in Applied Mathematics and Computational Science*, vol. 14, pp. 33–64, 2019.

[56] G. Strang, *Linear Algebra and its applications*. Brooks Cole, 4th ed., 1993.

[57] B. Sommeijer, "Explicit, high-order Runge-Kutta-Nyström methods for parallel computers," *Applied Numerical Mathematics*, vol. 13, no. 1-3, pp. 221–240, 1993.

[58] F. Penning, "Die Glimmentladung bei niedrigem Druck zwischen koaxialen Zylindern in einem axialen Magnetfeld," *Physica*, vol. 3, pp. 873–894, 1936.

[59] T. Eronen, V. S. Kolhinen, V. V. Elomaa, D. Gorelov, U. Hager, J. Hakala, A. Jokinen, A. Kankainen, P. Karvonen, S. Kopecky, I. D. Moore, H. Penttilä, S. Rahaman, S. Rinta-Antila, J. Rissanen, A. Saastamoinen, J. Szerypo, C. Weber, and J. Äystö, "Jyfltrap: a penning trap for precision mass spectroscopy and isobaric purification," *The European Physical Journal A*, vol. 48, pp. 1–21, 2012.

[60] L. Patacchini and I. Hutchinson, "Explicit time-reversible orbit integration in particle in cell codes with static homogeneous magnetic field," *Journal of Computational Physics*, vol. 228, pp. 2604–2615, 2009.

[61] K. Smedt, D. Ruprecht, J. Niesen, S. Tobias, and J. Nättilä, "New applications for the boris spectral deferred correction algorithm for plasma simulations," *Applied Mathematics and Computation*, vol. 442, p. 127706, 2023.

[62] U. Trottenberg, C. Oosterlee, and A. Schuller, *Multigrid*. Academic Press, 2000.

[63] S. Hirstoaga, "A first-order reduced model for a highly oscillating differential equation with application in penning traps," *SIAM Journal on Scientific Computing*, pp. S225–S245, 08 2024.

[64] J.-P. Berrut and L. N. Trefethen, "Barycentric lagrange interpolation," *SIAM Review*, vol. 46, no. 3, pp. 501–517, 2004.

[65] A. Burden, R. Burden, and J. Faires, *Numerical Analysis, 10th ed.* 01 2016.

[66] P. Henrici, *Elements of Numerical Analysis*. Wiley, 1964.

[67] J. Kevorkian and J. D. Cole, "Multiple scale and singular perturbation methods," 1996.

[68] A. Nayfeh, *Introduction to Perturbation Techniques*. Wiley, 1981.

[69] B. Dacorogna, *Direct methods in the calculus of variations*, vol. 78. Springer Science & Business Media, 2007.

[70] E. Frénod, "Application of the averaging method to the gyrokinetic plasma," *Asymptot. Anal.*, vol. 46, pp. 1–28, 2006.

[71] R. Hazeltine and J. Meiss, *Plasma Confinement*. ACM Press Frontier Series, Basic Books, 1992.

[72] F. Golse, "The vlasov-poisson system with strong magnetic field in quasineutral regime," *Mathematical Models and Methods in Applied Sciences*, vol. 9, pp. 661–714, 05 2003.

[73] M. Kretzschmar, "Particle motion in a penning trap," *European Journal of Physics*, vol. 12, p. 240, sep 1991.

[74] Y. He, Y. Sun, J. Liu, and H. Qin, "Volume-preserving algorithms for charged particle dynamics," *Journal of Computational Physics*, vol. 281, 01 2015.

[75] T. Haut and B. Wingate, "An asymptotic parallel-in-time method for highly oscillatory pdes," *SIAM Journal on Scientific Computing*, vol. 36, no. 2, pp. A693–A713, 2014.

[76] W. Ee and B. Engquist, "Multiscale modeling and computation," *Notices of the American Mathematical Society*, vol. 50, 01 2011.

[77] A. Taghibakhshi, S. MacLachlan, L. Olson, and M. West, "Optimization-based algebraic multigrid coarsening using reinforcement learning," in *Advances in Neural Information Processing Systems* (M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, eds.), vol. 34, pp. 12129–12140, Curran Associates, Inc., 2021.