COMPUTATIONAL METHODS FOR MULTI-SCALE TEMPORAL PROBLEMS: ALGORITHMS, ANALYSIS AND NUMERICAL EXPERIMENTS

A Dissertation

Presented to

Faculty of the Department of Civil and Environmental Engineering

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

in Civil and Environmental Engineering

by

Saeid Karimi

December 2016

COMPUTATIONAL METHODS FOR MULTI-SCALE TEMPORAL PROBLEMS: ALGORITHMS, ANALYSIS AND NUMERICAL EXPERIMENTS

____
Saeid Karimi

Approved:

____
Chair of the Committee,
Kalyana Babu Nakshatrala, Assistant Professor,
Department of Civil and Environmental Engineering

Committee Members:

____
Keh-Han Wang, Professor,
Department of Civil and Environmental Engineering

____
Kaspar J. Willam, Professor,
Department of Civil and Environmental Engineering

____
Gino J. Lim, Professor,
Department of Industrial Engineering

____
Yashashree Kulkarni, Associate Professor,
Department of Mechanical Engineering

____
Suresh K. Khator, Associate Dean,
Cullen College of Engineering

____
Roberto Ballarini, Professor and Chair,
Department of Civil and Environmental Engineering

COMPUTATIONAL METHODS FOR MULTI-SCALE TEMPORAL PROBLEMS: ALGORITHMS, ANALYSIS AND NUMERICAL EXPERIMENTS

An Abstract

of a Dissertation

Presented to

Faculty of the Department of Civil and Environmental Engineering

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

in Civil and Environmental Engineering

by

Saeid Karimi

December 2016

Abstract

A major challenge in numerical simulation of most natural phenomena is the presence of disparate temporal and spatial scales. Capturing all the fine features can be computationally prohibitive. Hence, development of efficient and accurate multi-scale numerical algorithms has gained immense attention from engineers and scientists. Typically, a single numerical method cannot efficiently capture all the aforementioned features. Due to the assumptions made in construction of numerical methods and mathematical models, the range of applicability to various length and time-scales is often limited. A direction in resolving this issue is to apply different numerical methods in different regions of the computational domain. This strategy enables computation of necessary details as desired by the user. In this work, we propose numerical methodologies based on domain partitioning techniques that allow different time-steps and time-integrators in different regions of the computational domain. The first problem of interest is elastodynamics, which can pose various temporal scales in impact, contact and wave propagation problems. A monolithic (strong) coupling algorithm based on non-overlapping domain partitioning is proposed. The proposed algorithm is based on the theory of differential/algebraic equations and its numerical stability, energy conservation and accuracy is studied in detail. Following these findings, we extend this algorithm to advection-diffusion-reaction problems. The proposed algorithm proves useful especially in cases where the relative strength of the involved processes changes dramatically with respect to spatial coordinates. Numerical stability and accuracy of this method is studied and its application to fast bimolecular chemical reactions is showcased. Further on, we confine our attention to single and multiple-relaxation-time lattice Boltzmann methods for the advection-diffusion equation and study their performance in preserving the maximum principle and the non-negative constraint. Finally, a computational framework based on overlapping domain decomposition methods is proposed. This framework is designed for advection-diffusion problems and allows coupling of the finite element method and lattice Boltzmann methods with different time-steps and grid sizes. Additionally, a new method for enforcing the Dirichlet and Neumann boundary conditions on the numerical solution from the lattice Boltzmann method is proposed. This method is based on maximization of entropy and ensures non-negativity of the discrete distributions on the boundary of the domain. We study the performance of this framework through numerical experiments and showcase its application to fast and equilibrium chemical reactions.

Table of Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION AND MOTIVATION

One of the persisting challenges in computational sciences is the presence of disparate mathematical scales in the domain of computation [3]. Examples of these mathematical scales can be found in a variety of practical problems from solid and fluid mechanics; for instance, solute and contaminant transport in porous media, contact problems, dynamic crack propagation and wave propagation. Furthermore, in interaction-type (multi-physics) problems such as fluid-structure interaction, soil-structure interaction and thermal-structure interaction this disparity in mathematical scales is even more accentuated. Some, but not all, of the difficulties in numerical simulation of multi-scale problems are as follows:

(i) The physical response of the system of interest can change dramatically throughout the domain. For instance, material properties and the physical/chemical processes can be quite different from one point to another. For instance, in impact problems changes in dynamic and kinematic variables of the system occur at a much higher pace near the point of impact and are slower or damped away from it. Another well-known example is the advection-diffusion-reaction processes that can exhibit steep gradients (interior or interior layers) depending on the relative strength of the involved terms.

(ii) The mathematical model in use may be incapable of including all the important physical/chemical features. Usually, the underlying mathematical models for engineering and scientific problems are based on simplifying assumptions. However, these assumptions may result in inaccuracy in the numerical approximation.

(iii) The numerical method in use may not be able to capture all features of the physical or chemical processes accurately or efficiently enough. In other words, very fine discretization in the spatial or temporal domains might be needed, or artificial numerical effects could be a hurdle toward a reliable estimate. As an example, simulation of deformation of tall buildings under wind load, such as in aeroelasticity, cannot be performed using one numerical technique for the fluid and the solid mechanic problem. A reason is that predicting the response of a structure to wind loading involves spatial and temporal scales that are very different to the one used for predicting the response of the fluid (air in this example).

Because of the mentioned reasons, and possibly many more, developing mathematical and numerical frameworks for simulation of multi-scale and multi-physics problems has become a major topic of interest. Over the past few years, multitude of numerical methods have been developed to tackle the challenge of disparate time and length-scales. Some of the most popular approaches towards resolving the multiple length and time-scales are the Variational Multi-Scale (VMS) method [4], the Generalized Finite Element Methods (GFEM) [5], Multi-Scale Finite Element Methods (MSFEM) [6] and the Domain Decomposition Methods [7] in general. The main objective of this work is to present new methods based on domain decomposition techniques that allow different time-steps, time-integrators and finite element formulations in different subdomains. Hence, one can resolve multiple time-scales in the a variety of time-dependent problems.

The rest of this work is organized as follows. In Chapter 2, a monolithic (strong) and multi-time-step coupling method for elastodynamics is presented. In Chapter 3, this coupling method is extended to first-order time-dependent problem arising from advection-diffusion-reaction equations. In Chapter 4, performance of single and multiple-relaxation-time lattice Boltzmann methods in preserving maximum principle and the non-negative constraints is studied. In Chapter 5, a coupling strategy for hybrid integration of advection-diffusion problems using lattice Boltzmann and finite element methods is provided. Conclusions and possible directions for future research are in Chapter 6.

Chapter 2

MONOLITHIC MULTI-TIME-STEP COUPLING METHOD FOR ELASTODYNAMICS

2.1 Introduction and Motivation

Coupled problems (such as fluid-structure interaction, structure-structure interaction and thermal-structure interaction) have been the subject of intense research in recent years in both computational mechanics and applied mathematics. The report compiled by the Blue Ribbon Panel on Simulation-Based Engineering Science emphasizes that the ability to solve coupled problems will be vital to accelerate the advances in engineering and science through simulation [3]. Developing stable and accurate numerical strategies for coupled problems can be challenging due to several reasons. These problems may involve multiple temporal scales and different spatial scales. One may have to deal with different types of equations for different aspects of physics, which could be coupled nonlinear equations. It is noteworthy that there exists neither a complete mathematical theory (for existence, uniqueness, and sharp estimates) nor a comprehensive computational framework to solve any given coupled problem. Some of the current research efforts are targeted towards resolving the aforementioned issues. Other research efforts are towards developing linear and nonlinear solvers, parallel frameworks, and tools for heterogeneous computing environments (including GPU-based computing) for coupled problems.

Herein, we shall present a numerical approach that can handle moderate disparity in temporal scales. We shall take elastodynamics as the benchmark problem, as it serves two purposes. This problem is important in its own right. In addition, the problem serves as a model problem to develop numerical algorithms for fluid-structure interaction problems, which can be much more involved than a problem typically encountered in elastodynamics. In a fluid-structure interaction simulation, in addition to a coupling algorithm, robust mesh motion algorithms, data transfer algorithms to interpolate data across mismatching meshes, and stable solvers for fluids and solids are needed.

It is now well-recognized that neither implicit nor explicit time-stepping schemes will be totally advantageous to meet all the desired features in a numerical simulation (e.g., see the discussion in references [8, 9]). Many factors (which include mesh, physical properties of the subdomain, accuracy, stability, total time of interest) affect the choice of the time-stepping scheme(s) [10]. It is sometimes much more economical to adopt different time-steps and/or time-stepping schemes in different subdomains. To this end mixed methods and multi-time-step methods have been developed.

Multi-time-step and mixed methods

Mixed methods refer to a class of algorithms that employ different time-stepping schemes in different subdomains. Some early efforts on mixed methods are [11–17]. The use of different time-steps in different subdomains is referred to as multi-time-stepping or subcycling. Some representative works in this direction are [18–20]. But many of the prior efforts on mixed methods and multi-time-stepping suffer from one or more of the following deficiencies: (i) The method cannot handle multiple subdomains. (ii) The method may not be accurate for disparate material properties, and for highly graded meshes. (iii) The method may suffer from very stringent stability limits, which may not be practical to meet realistic problems. (iv) The accuracy and stability depend on the preferential treatment of certain subdomains. For example, in the application of the conventional staggered coupling method, one domain is made to advance before another. The accuracy and stability depends on the choice of the subdomain that has to advance first [21].

We conjecture that the main source of the aforementioned numerical deficiencies is due to the fact that the prior works tried to develop coupling methods for transient problems by extending the strategies that were successful in developing partitioned schemes for static problems. However, it should be emphasized that designing coupling algorithms or partitioned schemes for transient problems require special attention compared to static problems. The governing equations for both undecomposed and decomposed static problems are algebraic equations. In the case of transient problems, the governing equations of an undecomposed problem are ordinary differential equations (ODEs) whereas the governing equations of a decomposed problem are differential-algebraic equations (DAEs).

Many of the prior works just employed the time-stepping schemes that are primarily developed for ODEs to construct partitioned schemes. However, it is well-known in the numerical analysis literature that care should be taken in applying popular time integrating schemes developed for ODEs to solve DAEs. The title of Petzold's seminal work [22] – "Differential/algebraic equations are not ODEs" – succinctly summarizes this fact. This viewpoint was also taken in references [9, 23] to develop coupling methods for first-order transient systems.

This chapter aims to develop a coupling method that allows different time-steps and different time integrators in different parts of the computational domain, which will be achieved using the results from the theory of differential-algebraic equations (e.g., Ascher and Petzold [24]). In recent years, the trend is to use dual Schur approach to develop multi-time-step coupling algorithms for second-order transient systems. A notable work in this direction is by Gravouil and Combescure (e.g., Reference [8]), which we shall refer to as the GC method. Based on the GC method, Pegon and Magonette developed a parallel inter-field method (the PM method), reference [25] is devoted to analysis of this method. Bursi et al. extended the PM method by employing the generalized α-method in [26]. Real time partitioned time-integration using the LSRT methods has been of interest recently in [27]. Mahjoubi and Krenk proposed a multi-time-step coupling method using state-pace time integration in [28], a more general presentation of which appears in [29]. Another work that is relevant to the current chapter is by Prakash and Hjelmstad [30], which we shall refer to as the PH method. It is worth to critically review the GC and PH methods.

A critical analysis of the GC and PH methods

The GC method is a multi-time-step coupling method for structural problems based on Newmark family of time integrators. The GC coupling method is built based on the following assumptions:

(GC1) Enforcing the continuity of velocity on the interface at the fine time-steps.

(GC2) Linear interpolation of interface velocities.

(GC3) Linear interpolation of Lagrange multiplier within the coarsest time-step.

The GC method is shown to exhibit excessive numerical damping (for example, see reference [30] and the numerical results presented in Section 2.6 of this chapter). The PH method is based on a modification to the GC method, and is constructed based on the following assumptions:

(PH1) Employed continuity of velocities along the subdomain interface at coarse time-steps.

(PH2) Linear interpolation of *all* kinematic variables (displacements, velocities, accelerations of the nodes on the subdomain interface and in the interior of the subdomains) within a coarse time-step.

(PH3) The method as it is presented in reference [30] is valid only for two subdomains.

(PH4) The subdomain that has the largest time-step has a more significant role in formulating the algorithm.

In Section 2.4, we shall show that Assumption (PH2) is not consistent with the underlying physics and need not be consistent with the underlying numerical time-stepping scheme. It is also claimed that the PH method is energy preserving implying that the coupling does not affect the total physical energy of the system. In a subsequent section, we shall present various notions of energy preserving by a coupling algorithm, and show that the PH method is not energy preserving (on the contrary to what has been claimed in Reference [30]).

Main contributions of this chapter

The proposed coupling method is developed by selecting the ideal combination from the assumptions of the GC and PH methods, and thereby eliminating all the deficiencies that these two methods suffer from. This chapter has made several advancements in multi-time-step coupling of second-order transient systems, and some of the main ones are as follows:

(i) Developing a coupling method that can handle multiple subdomains, allows different time-steps in different subdomains, allows different time-stepping schemes under the Newmark family in different subdomains, and is stable and accurate.

(ii) A stability proof using the energy method to obtain sufficient conditions for multi-time-step coupling is presented. Unlike many of the earlier works, the contribution of interface and subdomains is taken into account to derive the stability criteria. Unlike the prior works on multi-time-step coupling [8, 30], the proof is constructed by taking into account the contributions from all the subdomains and the interface, which is the correct form.

(iii) Documented the deficiencies of backward difference formulae (BDF) and implicit Runge-Kutta (IRK) schemes (which are popular for solving differential-algebraic equations) for solving second-order transient systems with invariants (e.g., conservation of energy).

(iv) New notions of energy preservation are introduced and conditions under which the proposed method satisfies any of those notions are also derived.

(v) A systematic study (both on the theoretical and numerical fronts) on the effect of subcycling and system time-step on the accuracy is presented. Specifically, we have shown that subcycling need not always improve accuracy. A criterion is devised to guide whether subcycling will improve accuracy or not. An attractive feature is that this criterion can be calculated on the fly during a numerical simulation.

An outline of the chapter

The remainder of this chapter is organized as follows. Section 2.2 briefly outlines Newmark family of time stepping schemes. Section 2.3 presents the governing equations for multiple subdomains with a discussion on the numerical treatment of interface constraints. Section 2.4 presents the proposed multi-time-step coupling method. A systematic theoretical analysis of the proposed coupling method (which includes stability analysis based on the energy method, influence of perturbations, bounds on interface drifts) is presented in Section 2.5. In Section 2.6, some of the theoretical predictions are verified using a simple lumped parameter system. Section 2.7 is devoted to numerical energy dissipation properties of the proposed coupling method. Some deficiencies of employing backward difference formulae and implicit

Runge-Kutta schemes for developing coupling algorithms for elastodynamics are discussed in Section 2.8. Several representative numerical examples are presented in Section 2.9 to illustrate the performance of the proposed coupling method. Conclusions are drawn in Section 2.10.

2.2 Newmark Family of Time-Stepping Schemes

Consider a system of second-order ordinary differential equations as

$$\boldsymbol{M}\ddot{\boldsymbol{u}}(t) + \boldsymbol{K}\boldsymbol{u}(t) = \boldsymbol{f}(t) \quad t \in (0, T], \tag{2.1}$$

where t denotes time, T denotes the time interval of interest, $\boldsymbol{M}$ is a symmetric positive definite matrix, $\boldsymbol{K}$ is a symmetric positive semidefinite matrix, and a superposed dot denotes derivative with respect to the time. The above system of equations can arise from a semi-discrete finite element discretization of the governing equations in linear elastodynamics [31]. In this case, $\boldsymbol{M}$ is referred to as the mass matrix, $\boldsymbol{K}$ is the stiffness matrix, and $\boldsymbol{u}(t)$ is the nodal displacement vector. Of course, one has to augment the above equation with initial conditions, which, in the context of elastodynamics, will be the prescription of the initial displacement vector and the initial velocity vector. One popular approach for solving equation (2.1) numerically is to employ a time-stepping scheme from the Newmark family [32]. We now present the Newmark time-stepping schemes in the context of undecomposed problem (i.e., the computational domain is not decomposed into subdomains). In the subsequent sections, we shall extend the presentation to multiple subdomains with the possibility of using different time-steps and/or different time integrators under Newmark family in different subdomains.

Let the time interval of interest T be divided into N sub-intervals such that

$$[0,T] = \bigcup_{n=1}^{N}[t_{n-1},t_n], \tag{2.2}$$

where $0 = t_0 < t_1 < \cdots < t_N = T$ are referred to as time levels. To make the presentation simple, we shall assume that the sub-intervals are uniform, in other words

$$\Delta t = t_n - t_{n-1} \quad \forall n = 1,\cdots,N, \tag{2.3}$$

where Δt is commonly referred to as the time-step. It should be, however, noted that the presentation can be easily extended to incorporate variable time-steps.

Remark 1 *In our development of the proposed multi-time-step coupling method, we shall use different kinds of time-steps (e.g., subdomain time-step, system time-step). These time-steps will be introduced in a subsequent section. For the present discussion, such a distinction is not required, as for single subdomain there is only one time-step.*

We shall employ the following notation to denote displacement, velocity and acceleration nodal vectors at discrete time levels as

$$\boldsymbol{d}^{(n)} = \boldsymbol{u}(t = t_n), \quad \boldsymbol{v}^{(n)} = \left.\frac{d\boldsymbol{u}}{dt}\right|_{t=t_n} \text{ and } \quad \boldsymbol{a}^{(n)} = \left.\frac{d^2\boldsymbol{u}}{dt^2}\right|_{t=t_n}. \tag{2.4}$$

Newmark family of time stepping schemes, which is a two-parameter family of time integrators written as

$$\boldsymbol{d}^{(n+1)} = \boldsymbol{d}^{(n)} + \Delta t\ \boldsymbol{v}^{(n)} + \frac{\Delta t^2}{2}\left((1-2\beta)\boldsymbol{a}^{(n)} + 2\beta\boldsymbol{a}^{(n+1)}\right) \text{ and} \tag{2.5a}$$

$$\boldsymbol{v}^{(n+1)} = \boldsymbol{v}^{(n)} + \Delta t\left((1-\gamma)\boldsymbol{a}^{(n)} + \gamma\boldsymbol{a}^{(n+1)}\right), \tag{2.5b}$$

where β and γ are user-specified parameters. A numerical solution at $(n+1)$-th time level can be obtained by simultaneously solving equations (2.5a)–(2.5b) with the equation

$$\boldsymbol{M}\boldsymbol{a}^{(n+1)} + \boldsymbol{K}\boldsymbol{d}^{(n+1)} = \boldsymbol{f}^{(n+1)}, \tag{2.6}$$

where

$$\boldsymbol{f}^{(n+1)} := \boldsymbol{f}(t = t_{n+1}). \tag{2.7}$$

It is well-known that one needs to choose $\gamma \geq 1/2$ for numerical stability [33]. The time-stepping scheme will be unconditionally stable if $2\beta \geq \gamma$, and will be conditionally stable if $2\beta < \gamma$. Some popular time-stepping schemes under the Newmark family are the central difference scheme $(\gamma = 1/2, \beta = 0)$, the average acceleration scheme $(\gamma = 1/2, \beta = 1/4)$, and the linear acceleration scheme $(\gamma = 1/2, \beta = 1/6)$. The central difference scheme is also referred to as the velocity Verlet scheme, which is the case in the molecular dynamics literature (e.g., see reference [34]). The central difference scheme is explicit, second-order accurate, and conditionally stable. The average acceleration scheme is implicit, second-order accurate, and unconditionally stable. The linear acceleration scheme is implicit, second-order accurate, and conditionally stable. For further details on Newmark family of time-stepping schemes in the context of undecomposed problem, see references [31, 33, 35].

2.3 Governing Equations for Multiple Subdomains

We now write governing equations for multiple subdomains. We will also outline various ways to write subdomain interface conditions, and discuss their pros and cons. To this end, let us divide the domain Ω into S non-overlapping subdomains, which will be denoted by $\Omega_1, \cdots, \Omega_S$. That is,

$$\Omega = \bigcup_{i=1}^{S} \Omega_i \quad \text{and} \quad \Omega_i \cap \Omega_j = \varnothing \quad \text{for } i \neq j. \tag{2.8}$$

We shall assume that the meshes in the subdomains are conforming along the subdomain interface, as shown in Figure 2.1. There are several ways to enforce the continuity along the interface, and hence, several ways to write the governing equations for multiple subdomains. Herein, we shall employ the dual Schur approach [36], which is also employed in the references that are relevant to this chapter (i.e., references [8, 30]).

We shall denote the number of displacement degrees-of-freedom in the i-th subdomain by N_i. The size of the velocity and acceleration nodal vectors in the i-th subdomain will also be $N_i \times 1$. The interface continuity conditions can be compactly written using signed Boolean matrices. A signed Boolean matrix is a matrix with entries either 0, −1, or +1 such that each row has *at most* one non-zero entry. Let us denote the total number of interface constraints by N_C. The size of the matrix $\boldsymbol{C}_i$ will be $N_C \times N_i$.

The governing equations for constrained multiple subdomains in a (time) continuous setting to be

$$\boldsymbol{M}_i \ddot{\boldsymbol{u}}_i(t) + \boldsymbol{K}_i \boldsymbol{u}_i(t) = \boldsymbol{f}_i(t) + \boldsymbol{C}_i^{\mathrm{T}} \boldsymbol{\lambda}(t) \quad \forall i = 1, \cdots, S, \text{and} \tag{2.9a}$$

$$\sum_{i=1}^{S} \boldsymbol{C}_i \boldsymbol{u}_i\left(t\right) = \boldsymbol{0} \tag{2.9b}$$

where the displacement vector of the i-th subdomain is denoted by $\boldsymbol{u}_i\left(t\right)$, and the external force applied to the i-th subdomain is denoted by $\boldsymbol{f}_i(t)$. The mass and stiffness matrices of the i-th subdomain are denoted by $\boldsymbol{M}_i$ and $\boldsymbol{K}_i$ respectively. In this chapter, we shall assume that the matrices $\boldsymbol{M}_i$ are symmetric and positive definite, and the matrices $\boldsymbol{K}_i$ to be symmetric and positive semi-definite. Equation (2.9b) is an algebraic constraint enforcing kinematic continuity of displacements along the subdomain interface. The vector $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers arising due to the enforcement of constraints. The above equations should be augmented with appropriate initial conditions. A brief discussion on the derivation of the above equations can be found in Appendix. Equation (2.9) form a system of differential-algebraic equations. For the benefit of broader audience, we now briefly discuss differential-algebraic equations.

Remark 2 *If one wants to including physical damping, equation* (2.9a) *should be replaced with*

$$\boldsymbol{M}_i \ddot{\boldsymbol{u}}_i(t) + \boldsymbol{D}_i \dot{\boldsymbol{u}}_i + \boldsymbol{K}_i \boldsymbol{u}_i(t) = \boldsymbol{f}_i(t) + \boldsymbol{C}_i^{\mathrm{T}} \boldsymbol{\lambda}(t) \quad \forall i = 1, \cdots, S, \tag{2.10}$$

where $\boldsymbol{D}_i$ is the damping matrix for the i-th subdomain. One can then easily extend the proposed multi-time-step coupling method to include contribution from physical damping. However, a more challenging task is to characterize the performance of the

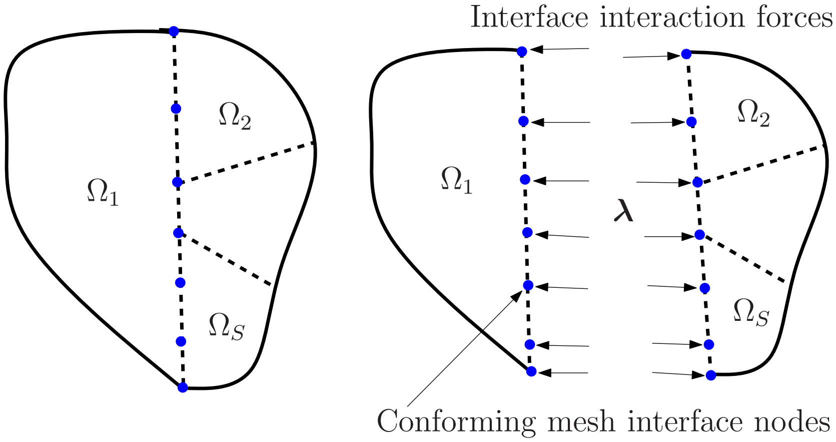


Figure 2.1: A pictorial description of multiple subdomains: The domain Ω is decomposed into S subdomains, which are denoted by $\Omega_1, \cdots, \Omega_S$. The subdomain interface is indicated using dashed curves. The mesh is assumed to be conforming along the subdomain interface.

coupling method due to damping. This will depend on several issues like: whether the damping is due to viscoelasticity, plasticity, viscoplasticity or frictional contact? Whether the damping matrix be modeled as Rayleigh damping (which basically assumes that the damping matrix is a linear combination of the mass matrix and the stiffness matrix)? A systematic treatment of these issues are beyond the scope of this chapter, and will be addressed in our future works.

Differential-algebraic equations

A differential-algebraic equation (DAE) is defined as an equation involving unknown functions and their derivatives. A DAE, in its most general form, can be written as

$$\boldsymbol{g}\left(\dot{\boldsymbol{x}}(t), \boldsymbol{x}(t), t\right) = \boldsymbol{0} \quad t \in (0, T], \tag{2.11}$$

where the unknown function is denoted by $\boldsymbol{x}(t)$. A DAE of the form given by equation (2.11) is commonly referred to as an implicit DAE. A quantity that is useful in the study of (smooth) differential-algebraic equations is the so-called differential index, which was first introduced by Gear [37] and further popularized by Petzold and Campbell [24, 38]. For a DAE of the form given by equation (2.11), differential index is the minimum number of times one has to differentiate with respect to the independent variable t to be able to *rewrite* equation (2.11) in the form

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{h}(\boldsymbol{x}(t), t), \tag{2.12}$$

using only algebraic manipulations. It is commonly believed that the higher the differential index the greater will be the difficulty in obtaining stable numerical solutions. An important subclass of DAEs is titled as semi-explicit DAEs of Hessenberg form, which can be written as

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{p}(\boldsymbol{x}(t), \boldsymbol{y}(t), t), \text{ and} \tag{2.13a}$$

$$\boldsymbol{0} = \boldsymbol{q}(\boldsymbol{x}(t)). \tag{2.13b}$$

From the above discussion, it is evident that the DAE given by equations (2.9) is a semi-explicit DAE with differential index 3. One way of solving a higher index DAE is to employ the standard index reduction technique to obtain a *mathematically* equivalent DAE with lower differential index. It is noteworthy that index reduction can have deleterious effect on the stability and accuracy of numerical solutions (e.g., drift in the constraint). We now explore several mathematically equivalent forms of governing equations, which will have differential index ranging from 0 to 3.

Subdomain interface constraints

As stated earlier, dual Schur techniques for domain decomposition are of interest throughout this chapter. One may write several types of continuity constraints resulting in semi-explicit DAEs of different differential indices. Note that in a continuous setting all these versions are mathematically equivalent. However, from a numerical point of view, their performance can be dramatically different. In fact, some may even exhibit instabilities. Some ways of constructing dual Schur methods are discussed below, which guide future research on constructing new multi-time-step coupling methods.

$\boldsymbol{d}$-continuity method: This method considers the original set of equations given by equations (2.9). The method obtains $(\boldsymbol{u}_1(t), \cdots, \boldsymbol{u}_S(t), \boldsymbol{\lambda}(t))$ for $t \in (0, T]$ by solving

$$\boldsymbol{M}_i \ddot{\boldsymbol{u}}_i(t) + \boldsymbol{K}_i \boldsymbol{u}_i(t) = \boldsymbol{f}_i(t) + \boldsymbol{C}_i^{\mathrm{T}} \boldsymbol{\lambda}(t) \quad \forall i = 1, \cdots, S, \text{ and} \tag{2.14a}$$

$$\sum_{i=1}^{S} \boldsymbol{C}_i \boldsymbol{u}_i(t) = \boldsymbol{0}. \tag{2.14b}$$

The above equations (2.14a)–(2.14b) form a system of DAEs of differential index *three*. It has been discussed in the literature that the numerical solutions based on this method are prone to instabilities [35, 39].

$\boldsymbol{v}$-continuity method: This method obtains $(\boldsymbol{u}_1(t), \cdots, \boldsymbol{u}_S(t), \boldsymbol{\lambda}(t))$ for $t \in (0, T]$ by solving the equations

$$\boldsymbol{M}_i \ddot{\boldsymbol{u}}_i(t) + \boldsymbol{K}_i \boldsymbol{u}_i(t) = \boldsymbol{f}_i(t) + \boldsymbol{C}_i^{\mathrm{T}} \boldsymbol{\lambda}(t) \quad \forall i = 1, \cdots, S, \text{ and} \tag{2.15a}$$

$$\sum_{i=1}^{S} \boldsymbol{C}_i \dot{\boldsymbol{u}}_i(t) = \boldsymbol{0}. \tag{2.15b}$$

The above equations form a system of DAEs of differential index *two*. The $\boldsymbol{v}$-continuity method is of interest in this chapter and in the previous works by Gravouil and Combescure [8], and Prakash and Hjelmstad [30]. This form of equations provides a simple but stable framework for seeking numerical solutions, and will form the basis for the proposed multi-time-step coupling method.

$\boldsymbol{a}$-continuity method: This method obtains $(\boldsymbol{u}_1(t), \cdots, \boldsymbol{u}_S(t), \boldsymbol{\lambda}(t))$ for $t \in (0, T]$ by solving the equations

$$\boldsymbol{M}_i \ddot{\boldsymbol{u}}_i(t) + \boldsymbol{K}_i \boldsymbol{u}_i(t) = \boldsymbol{f}_i(t) + \boldsymbol{C}_i^{\mathrm{T}} \boldsymbol{\lambda}(t) \quad \forall i = 1, \cdots, S, \text{ and} \tag{2.16a}$$

$$\sum_{i=1}^{S} \boldsymbol{C}_i \ddot{\boldsymbol{u}}_i(t) = \boldsymbol{0}. \tag{2.16b}$$

The differential index of the above DAE is *unity*. A drawback of this method is that there can be significant irrecoverable drift in the displacements without employing constraint stabilization or projection methods. The drift can be attributed to the fact that there is no explicit constraint on the continuity of displacements along the subdomain interface. We, therefore, do not employ this method in this chapter.

Baumgarte stabilization method: Under this method, kinematic constraint appears as a linear combination of the kinematic constraints under the $\boldsymbol{d}$-continuity, $\boldsymbol{v}$-continuity and $\boldsymbol{a}$-continuity methods. This method obtains $(\boldsymbol{u}_1(t), \cdots, \boldsymbol{u}_S(t), \boldsymbol{\lambda}(t))$ for $t \in (0, T]$ by solving the equations

$$\boldsymbol{M}_i \ddot{\boldsymbol{u}}_i(t) + \boldsymbol{K}_i \boldsymbol{u}_i(t) = \boldsymbol{f}_i(t) + \boldsymbol{C}_i^{\mathrm{T}} \boldsymbol{\lambda}(t) \quad \forall i = 1, \cdots, S, \text{ and} \tag{2.17a}$$

$$\sum_{i=1}^{S} \boldsymbol{C}_i \ddot{\boldsymbol{u}}_i(t) + \frac{\alpha}{\Delta t} \sum_{i=1}^{S} \boldsymbol{C}_i \dot{\boldsymbol{u}}_i(t) + \frac{\beta}{\Delta t^2} \sum_{i=1}^{S} \boldsymbol{C}_i \boldsymbol{u}_i(t) = \boldsymbol{0}, \tag{2.17b}$$

where α and β are non-dimensional user-specified parameters. One can achieve damping in the drift displacements by choosing parameters satisfying the condition

$\alpha^2 - 4\beta < 0$. This method was first proposed by Baumgarte in [40] for constrained mechanical systems. Note that in [40], the coefficients α and β have dimensions of $[T]^{-1}$ and $[T]^{-2}$ respectively, but in (2.17), those coefficients are non-dimensionalized. In Reference [23], the Baumgarte stabilization method has been extended to first-order differential-algebraic equations, and the authors were able to derive sufficient conditions for stability using the energy method. To the best of the authors' knowledge deriving sufficient conditions for stability under the Baumgarte method for second-order differential-algebraic equations is still an open problem. Some notable efforts in this direction are [41–43].

Rewriting as a system of ordinary differential equations: One can differentiate further, and rewrite the a-continuity method as a system of ordinary differential equations. From the definition of differential index, it is obvious that the differential index of the resulting governing equations will be *zero*. The governing equations for this method take the form

$$\dot{\boldsymbol{u}}_i = \boldsymbol{v}_i \ , \tag{2.18a}$$

$$\dot{\boldsymbol{v}}_i = \boldsymbol{M}_i^{-1} \left(\boldsymbol{f}_i + \boldsymbol{C}_i^{\mathrm{T}} \boldsymbol{\lambda} - \boldsymbol{K}_i \boldsymbol{u}_i \right), \text{ and} \tag{2.18b}$$

$$\boldsymbol{\lambda} = \left(\sum_{i=1}^{S} \boldsymbol{C}_i \boldsymbol{M}_i^{-1} \boldsymbol{C}_i^{\mathrm{T}} \right)^{-1} \left[\sum_{i=1}^{S} \boldsymbol{C}_i \boldsymbol{M}_i^{-1} \left(\boldsymbol{K}_i \boldsymbol{u}_i - \boldsymbol{f}_i \right) \right]. \tag{2.18c}$$

The main drawback of the above method is that there will be significant irrecoverable drift in the continuity of subdomain interface displacements and velocities. As advocated by Petzold in her famous chapter [22], solving DAEs is much harder than solving systems of ODEs. Many of the popular integrators that are used for solving ODEs are not stable and accurate for solving DAEs.

Rewriting as a system of first-order differential-algebraic equations: Yet another approach is to rewrite the governing equations in first-order form, and then employ appropriate time-stepping schemes for solving first-order DAEs (e.g., backward difference formulae, implicit Runge-Kutta schemes). The first-order form can be achieved by introducing an auxiliary variable. The governing equations take the form

$$\dot{\boldsymbol{u}}(t) = \boldsymbol{v}(t) \ , \tag{2.19a}$$

$$\boldsymbol{M}_i \dot{\boldsymbol{v}}_i + \boldsymbol{K}_i \boldsymbol{u}_i = \boldsymbol{f}_i(t) + \boldsymbol{C}_i^{\mathrm{T}} \boldsymbol{\lambda}, \text{ and} \tag{2.19b}$$

$$\sum_{i=1}^{S} \boldsymbol{C}_i \boldsymbol{u}_i = \boldsymbol{0}. \tag{2.19c}$$

The differential index for the above system is *three*. If one replaces the interface constraint equation (2.19c) with either

$$\sum_{i=1}^{S} \boldsymbol{C}_i \dot{\boldsymbol{u}}_i = \boldsymbol{0} \quad \text{or} \quad \sum_{i=1}^{S} \boldsymbol{C}_i \boldsymbol{v}_i = \boldsymbol{0}, \tag{2.20}$$

then the differential index of the resulting differential-algebraic equations will be *two*. If the interface constraint equation (2.19c) is replaced wih

$$\sum_{i=1}^{S} \boldsymbol{C}_i \dot{\boldsymbol{v}}_i = \boldsymbol{0}, \tag{2.21}$$

then the resulting first-order DAEs will have index *one*.

In a subsequent section we shall show that the approach of rewriting the governing equations as first-order DAEs and then employing time-stepping schemes that are typically used for first-order transient systems is not accurate for elastodynamics. Hence, we do not employ such an approach to develop a multi-time-step coupling method. Instead, we consider the governing equations in second-order form and modify Newmark time-stepping schemes to be able to obtain stable and accurate results for resulting DAEs. In the next section, we shall extend the $\boldsymbol{v}$-continuity to be able to employ different time-steps in different subdomains, and to couple explicit and implicit time-stepping schemes.

2.4 Proposed Multi-Time-Step Coupling Method

The aim of this chapter is to solve equations (2.15a)–(2.15b) numerically by allowing each subdomain to have its own time-step and its own time integrator from the Newmark family of time stepping schemes. We first introduce notation that will help in presenting the proposed multi-time-step coupling method in a concise manner.

Notation for multi-time-step coupling

Both the GC and PH methods are devised by introducing the coarsest time-step, which is the maximum of all the subdomain time-steps. This creates bias, at least in the mathematical setting, towards the subdomain that has the maximum time-step. Herein, we alleviate this drawback by introducing the notion of system time-step, which is greater than or equal to the coarsest time-step. Moreover, this approach allows for the possibility of all subdomains to subcycle, which is illustrated in a subsequent section. Figure 3.2 gives a pictorial description of subdomain time-steps, system time-step, and the concept of subcycling. We shall define η_i to be the ratio

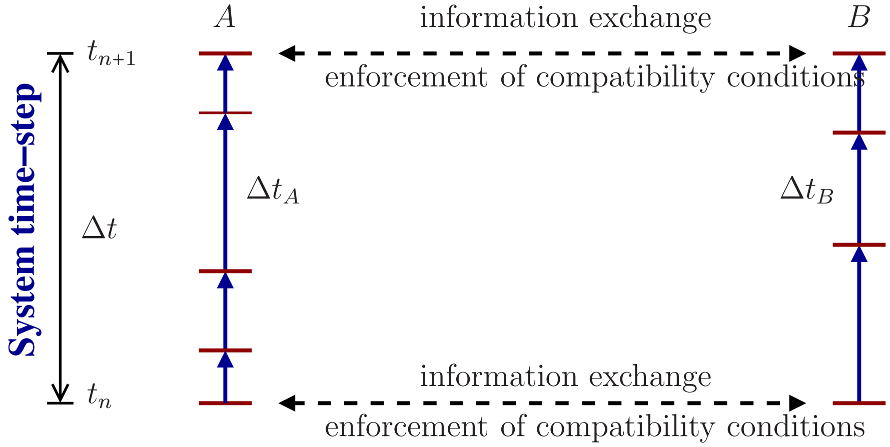


Figure 2.2: Multi-time-step time integration: A pictorial description of time levels (t_n), system time-step (Δt), subdomain time-step (Δt_i), and subcycling. Note that $\eta_i = \Delta t/\Delta t_i$. In this figure $i = A$ or B.

between system time-step (Δt) and the i-th subdomain time-step (Δt_i). That is

$$\eta_i := \frac{\Delta t}{\Delta t_i}\cdot \tag{2.22}$$

For simplicity, we shall assume that η_i is a (positive) integer.

We shall use the following notation to represent the value of a quantity of interest at subdomain time levels, e.g.,

$$\square_i^{\left(n+\frac{j}{\eta_i}\right)} \approx \square_i \left(t = n\Delta t + j\Delta t_i\right)\cdot \tag{2.23}$$

We shall employ the following notation to group the kinematic quantities

$$\boldsymbol{X}_i^{\left(n+\frac{j}{\eta_i}\right)} := \begin{bmatrix} \boldsymbol{a}_i^{\left(n+\frac{j}{\eta_i}\right)} \\ \boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)} \\ \boldsymbol{d}_i^{\left(n+\frac{j}{\eta_i}\right)} \end{bmatrix}, \quad \mathbb{X}_i^{(n+1)} := \begin{bmatrix} \boldsymbol{X}_i^{\left(n+\frac{1}{\eta_i}\right)} \\ \boldsymbol{X}_i^{\left(n+\frac{2}{\eta_i}\right)} \\ \vdots \\ \boldsymbol{X}_i^{(n+1)} \end{bmatrix} \text{ and } \quad \mathbb{X}^{(n+1)} := \begin{bmatrix} \mathbb{X}_1^{(n+1)} \\ \mathbb{X}_2^{(n+1)} \\ \vdots \\ \mathbb{X}_S^{(n+1)} \end{bmatrix}\cdot \tag{2.24}$$

The vector $\boldsymbol{X}_i^{\left(n+\frac{j}{\eta_i}\right)}$ contains all the kinematic unknowns for i-th subdomain over its subdomain time-step, $\mathbb{X}_i^{(n+1)}$ contains all the kinematic unknowns for i-th subdomain over a system time-step, and the vector $\mathbb{X}^{(n+1)}$ contains the kinematic unknowns of all subdomains over a system time-step. We define the following augmented subdomain signed Boolean matrices

$$\mathbb{C}_i := \left[\begin{array}{c|c|c|c|c} \underbrace{\mathcal{O}_i \quad \mathcal{O}_i \quad \mathcal{O}_i}_{1} & \underbrace{\mathcal{O}_i \quad \mathcal{O}_i \quad \mathcal{O}_i}_{2} & \cdots\cdots\cdots & \underbrace{\mathcal{O}_i \quad \mathcal{O}_i \quad \mathcal{O}_i}_{\eta_i - 1} & \underbrace{\mathcal{O}_i \quad \boldsymbol{C}_i \quad \mathcal{O}_i}_{\eta_i} \end{array}\right], \tag{2.25}$$

where the matrix $\mathcal{O}_i$ contains zeros of the same size as $\boldsymbol{C}_i$ (which is $N_C \times N_i$). It is evident that the size of $\mathbb{C}_i$ is $N_C \times 3\eta_i N_i$. The augmented signed Boolean matrix for the entire system is defined as

$$\mathbb{C} := \left[\begin{array}{cccc} \mathbb{C}_1 & \mathbb{C}_2 & \cdots & \mathbb{C}_S \end{array}\right]. \tag{2.26}$$

The size of $\mathbb{C}$ is $N_C \times \left(\sum_{i=1}^{S} 3\eta_i N_i\right)$. The following augmented signed Boolean matrices will be useful in taking into account the effect of interface forces using

$$\mathbb{B}_i^{\mathrm{T}} := \left[\begin{array}{ccc|ccc|c|ccc} -\frac{1}{\eta_i}\boldsymbol{C}_i & \mathcal{O}_i & \mathcal{O}_i & -\frac{2}{\eta_i}\boldsymbol{C}_i & \mathcal{O}_i & \mathcal{O}_i & \cdots\cdots\cdots & -\frac{\eta_i}{\eta_i}\boldsymbol{C}_i & \mathcal{O}_i & \mathcal{O}_i \end{array}\right]. \tag{2.27}$$

The corresponding signed Boolean matrix for the entire system can be written as

$$\mathbb{B} := \begin{bmatrix} \mathbb{B}_1 \\ \mathbb{B}_2 \\ \vdots \\ \mathbb{B}_S \end{bmatrix}. \tag{2.28}$$

We shall define the following augmented matrices for each subdomain as

$$\mathbb{L}_i := \begin{bmatrix} \boldsymbol{M}_i & \boldsymbol{O}_i & \boldsymbol{K}_i \\ -\gamma_i \Delta t_i \boldsymbol{I}_i & \boldsymbol{I}_i & \boldsymbol{O}_i \\ -\beta_i \Delta t_i^2 \boldsymbol{I}_i & \boldsymbol{O}_i & \boldsymbol{I}_i \end{bmatrix}, \quad \mathbb{R}_i := \begin{bmatrix} \boldsymbol{O}_i & \boldsymbol{O}_i & \boldsymbol{O}_i \\ (1-\gamma_i)\,\Delta t_i \boldsymbol{I}_i & \boldsymbol{I}_i & \boldsymbol{O}_i \\ \left(\frac{1}{2}-\beta_i\right)\Delta t_i^2 \boldsymbol{I}_i & \Delta t_i \boldsymbol{I}_i & \boldsymbol{I}_i \end{bmatrix}, \tag{2.29}$$

where $\boldsymbol{O}_i$ denotes a matrix containing zeros of size $N_i \times N_i$, and $\boldsymbol{I}_i$ is the identity matrix of size $N_i \times N_i$.

Multi-time-step coupling

The proposed multi-time-step coupling method is developed based on the following assumptions:

(A) Enforce the continuity of interface velocities at system time-steps.

(B) The corresponding Lagrange multipliers (which will be the interface reactions) are calculated at system time-steps. (It should be noted that the Lagrange multipliers are unknowns, and will be a part of the solution.)

(C) The Lagrange multipliers are interpolated linearly within system time-steps to approximate their values at subdomain time-steps.

(D) The equilibrium equations in each subdomain are enforced at their corresponding subdomain time levels in such a way that the coupling method can handle arbitrary number of subdomains.

Assumptions (B) and (C) take the mathematical form

$$\boldsymbol{\lambda}^{\left(n+\frac{j}{\eta_i}\right)} = \left(1 - \frac{j}{\eta_i}\right)\boldsymbol{\lambda}^{(n)} + \left(\frac{j}{\eta_i}\right)\boldsymbol{\lambda}^{(n+1)}, \tag{2.30}$$

where $\boldsymbol{\lambda}^{(n)}$ and $\boldsymbol{\lambda}^{(n+1)}$ are Lagrange multipliers at system time levels. Using equation (2.30), Assumption (D) takes the form

$$\boldsymbol{M}_i\boldsymbol{a}_i^{\left(n+\frac{j+1}{\eta_i}\right)} + \boldsymbol{K}_i\boldsymbol{d}_i^{\left(n+\frac{j+1}{\eta_i}\right)} - \frac{j+1}{\eta_i}\boldsymbol{C}_i^{\mathrm{T}}\left(\boldsymbol{\lambda}^{(n+1)} - \boldsymbol{\lambda}^{(n)}\right) = \boldsymbol{f}_i^{\left(n+\frac{j+1}{\eta_i}\right)} + \boldsymbol{C}_i^{\mathrm{T}}\boldsymbol{\lambda}^{(n)}, \tag{2.31}$$

and the relations for the time-stepping schemes for the i-th subdomain take the form

$$\boldsymbol{d}_i^{\left(n+\frac{j+1}{\eta_i}\right)} = \boldsymbol{d}_i^{\left(n+\frac{j}{\eta_i}\right)} + \Delta t_i \boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)} + \frac{(\Delta t_i)^2}{2}\left((1-2\beta_i)\,\boldsymbol{a}_i^{\left(n+\frac{j}{\eta_i}\right)} + 2\beta_i \boldsymbol{a}_i^{\left(n+\frac{j+1}{\eta_i}\right)}\right), \tag{2.32a}$$

$$\boldsymbol{v}_i^{\left(n+\frac{j+1}{\eta_i}\right)} = \boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)} + \Delta t_i \left((1-\gamma_i)\,\boldsymbol{a}_i^{\left(n+\frac{j}{\eta_i}\right)} + \gamma_i \boldsymbol{a}_i^{\left(n+\frac{j+1}{\eta_i}\right)}\right), \tag{2.32b}$$

where β_i and γ_i are the Newmark parameters for the i-th subdomain. Assumption (A) takes the mathematical form

$$\sum_{i=1}^{S} \mathbb{C}_i \mathbb{X}_i^{(n+1)} = \mathbf{0}. \tag{2.33}$$

Or, more compactly,

$$\mathbb{C}\mathbb{X}^{(n+1)} = \mathbf{0}. \tag{2.34}$$

Advance a subdomain over its subdomain time-step

Using the above notation, the governing equations to advance the state of i-th subdomain over its time-step can be compactly written as

$$\mathbb{L}_i \boldsymbol{X}_i^{\left(n+\frac{j+1}{\eta_i}\right)} - \left(\frac{j+1}{\eta_i}\right)\widetilde{\mathbb{C}}_i^{\mathrm{T}}\left(\boldsymbol{\lambda}^{(n+1)} - \boldsymbol{\lambda}^{(n)}\right) = \mathbb{P}_i^{\left(n+\frac{j+1}{\eta_i}\right)} + \widetilde{\mathbb{C}}_i^{\mathrm{T}}\boldsymbol{\lambda}^{(n)} + \mathbb{R}_i \boldsymbol{X}_i^{\left(n+\frac{j}{\eta_i}\right)}, \tag{2.35}$$

where we used the notations

$$\widetilde{\mathbb{C}}_i := \begin{bmatrix} \boldsymbol{C}_i & \mathcal{O}_i & \mathcal{O}_i \end{bmatrix} \text{and} \quad \mathbb{P}_i^{\left(n+\frac{j}{\eta_i}\right)} := \begin{bmatrix} \boldsymbol{f}_i^{\left(n+\frac{j}{\eta_i}\right)} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}. \tag{2.36}$$

Advance a subdomain over a system time-step

The governing equations to advance a subdomain over a system time-step can be compactly written as

$$\mathbb{Q}_i \mathbb{X}_i^{(n+1)} + \mathbb{B}_i \left(\boldsymbol{\lambda}^{(n+1)} - \boldsymbol{\lambda}^{(n)} \right) = \mathbb{F}_i^{(n+1)}, \tag{2.37}$$

where the matrix $\mathbb{Q}_i$ is defined as

$$\mathbb{Q}_i := \begin{bmatrix} \mathbb{L}_i & & & \\ -\mathbb{R}_i & \mathbb{L}_i & & \\ & \ddots & \ddots & \\ & & -\mathbb{R}_i & \mathbb{L}_i \end{bmatrix}. \tag{2.38}$$

Advance all subdomains over a system time-step

We now write the governing equations to advance all the subdomains from (system) time level t_n to t_{n+1} (i.e., advance all subdomains by a system time-step) in a compact form. The mathematical statement takes the following form: Find $\mathbb{X}^{(n+1)}$ and $\boldsymbol{\lambda}^{(n+1)}$ by solving the system of linear equations

$$\begin{bmatrix} \mathbb{A} & \mathbb{B} \\ \mathbb{C} & \mathbb{O} \end{bmatrix} \begin{bmatrix} \mathbb{X}^{(n+1)} \\ \boldsymbol{\lambda}^{(n+1)} - \boldsymbol{\lambda}^{(n)} \end{bmatrix} = \begin{bmatrix} \mathbb{F}^{(n+1)} \\ \mathbf{0} \end{bmatrix}, \tag{2.39}$$

where the matrix $\mathbb{A}$ is defined as

$$\mathbb{A} := \begin{bmatrix} \mathbb{Q}_1 & & & \\ & \mathbb{Q}_2 & & \\ & & \ddots & \\ & & & \mathbb{Q}_S \end{bmatrix}, \tag{2.40}$$

with the notation

$$\mathbb{F}^{(n+1)} := \begin{bmatrix} \mathbb{F}_1^{(n+1)} \\ \mathbb{F}_2^{(n+1)} \\ \vdots \\ \mathbb{F}_S^{(n+1)} \end{bmatrix} \quad \text{and} \quad \mathbb{F}_i^{(n+1)} := \begin{bmatrix} \mathbb{P}_i^{\left(n+\frac{1}{\eta_i}\right)} + \widetilde{\mathbb{C}}_i^{\mathrm{T}} \boldsymbol{\lambda}^{(n)} + \mathbb{R}_i \boldsymbol{X}_i^{(n)} \\ \mathbb{P}_i^{\left(n+\frac{2}{\eta_i}\right)} + \widetilde{\mathbb{C}}_i^{\mathrm{T}} \boldsymbol{\lambda}^{(n)} \\ \vdots \\ \mathbb{P}_i^{(n+1)} + \widetilde{\mathbb{C}}_i^{\mathrm{T}} \boldsymbol{\lambda}^{(\boldsymbol{n})} \end{bmatrix}. \tag{2.41}$$

Comments on the derivation of the PH method in Reference [30]

One main assumption in deriving the PH method is that the acceleration, velocity and displacement *all* vary linearly with time within a system time-step. It should be emphasized that such an assumption is *not self-consistent*. Moreover, this assumption need not be consistent with the underlying time stepping scheme. To wit, the assumption made in deriving the PH method takes the mathematical form

$$\boldsymbol{a}_i^{\left(n+\frac{j}{\eta_i}\right)} = \left(1 - \frac{j}{\eta_i}\right)\boldsymbol{a}_i^{(n)} + \frac{j}{\eta_i}\boldsymbol{a}_i^{(n+1)}, \tag{2.42a}$$

$$\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)} = \left(1 - \frac{j}{\eta_i}\right)\boldsymbol{v}_i^{(n)} + \frac{j}{\eta_i}\boldsymbol{v}_i^{(n+1)}, \text{and} \tag{2.42b}$$

$$\boldsymbol{d}_i^{\left(n+\frac{j}{\eta_i}\right)} = \left(1 - \frac{j}{\eta_i}\right)\boldsymbol{d}_i^{(n)} + \frac{j}{\eta_i}\boldsymbol{d}_i^{(n+1)}. \tag{2.42c}$$

Let us consider equation (2.42a), which can be interpreted as

$$\boldsymbol{a}_i(t) = \boldsymbol{a}_i^{(n)} + \frac{(t - t_n)}{(t_{n+1} - t_n)}\left(\boldsymbol{a}_i^{(n+1)} - \boldsymbol{a}_i^{(n)}\right) \quad t_n \leq t \leq t_{n+1}. \tag{2.43}$$

If the acceleration varies linearly with the time, the velocity should vary quadratically with the time, and the displacement should vary cubic with the time. Hence, equations (2.42a)–(2.42c) are *not* inherently consistent.

In addition, this assumption need not be consistent with the underlying time stepping scheme, which is typically derived by assuming an ansatz functional form for the variation of the acceleration, velocity or displacement with respect to the time. For example, Newmark average acceleration scheme ($\gamma = 1/2, \beta = 1/4$) is constructed by assuming that the acceleration is constant within a time-step [33]. The assumption made in deriving the PH method that the acceleration varies linearly with time within a system time step (i.e., equation (2.42a) or (2.43)) will not be consistent if, say, one employs the Newmark average acceleration scheme under the multi-time-step coupling method. More importantly, as shown in the previous section, such a mathematically *in*consistent assumption is not warranted to develop a multi-time-step coupling method. Also, the multi-time-step coupling method as presented in Reference [30] is restricted to two subdomains. There is no restriction on the number of subdomains in the proposed multi-time-step coupling method.

Remark 3 *As mentioned earlier, the PH method (as presented in Reference [30]) can handle only two subdomains. Preference is given to the subdomain that has the coarsest time-step. For example, in the final form of the PH method (see [30, equation (43)]), the forcing function to advance subdomain B uses* $\mathbf{S}_i$*, which is based on the quantities of subdomain A. But the forcing function to advance subdomain A does not employ any quantities of subdomain B. Recently, a tree-based approach has been proposed in Reference [44] that combines two subdomains at a time to solve multiple subdomains, which will be computationally intensive. In the case of two subdomains (i.e.,* $S = 2$*), the proposed coupling method will be same as the PH method* *if the applied external forces on the subdomain with the coarse time-step is affine with respect to time. The proposed coupling method, however, can handle multiple subdomains, and does not give preference to any subdomain. It should be emphasized that if one wants to implement in a recursive manner using a tree-based approach, the proposed method is amenable.*

2.5 A Theoretical Analysis of the Proposed Coupling Method

Stability analysis using the energy method

We shall employ the energy method to show the stability of the proposed multi-time-step coupling method. The energy method is a popular strategy employed in Mathematical Analysis to derive estimates and to perform stability analysis. The method is widely employed in the theory of partial differential equations [45], and numerical analysis [31, 46]. The basic idea behind the energy method is to choose an appropriate norm (which is referred to as the energy norm) and show that the solution is bounded under this norm. It should be noted that the energy norm may not correspond to the physical energy.

We shall now introduce the notation that is needed to apply the energy method. The jump and average operators over the system time-step are, respectively, denoted by $[\![\cdot]\!]$ and $\langle\!\langle\cdot\rangle\!\rangle$. That is,

$$[\![\boldsymbol{x}^{(n)}]\!] := \boldsymbol{x}^{(n+1)} - \boldsymbol{x}^{(n)} \text{ and} \tag{2.44a}$$

$$\langle\!\langle\boldsymbol{x}^{(n)}\rangle\!\rangle := \frac{1}{2}\left(\boldsymbol{x}^{(n)} + \boldsymbol{x}^{(n+1)}\right). \tag{2.44b}$$

The jump and average operators over the subdomain time-step of the i-th subdomain are, respectively, denoted by $[\cdot]_i$ and $\langle\cdot\rangle_i$. That is,

$$\left[\boldsymbol{x}^{\left(n+\frac{j}{\eta_i}\right)}\right]_i := \boldsymbol{x}^{\left(n+\frac{j+1}{\eta_i}\right)} - \boldsymbol{x}^{\left(n+\frac{j}{\eta_i}\right)} \text{ and} \tag{2.45a}$$

$$\left\langle\boldsymbol{x}^{\left(n+\frac{j}{\eta_i}\right)}\right\rangle_i := \frac{1}{2}\left(\boldsymbol{x}^{\left(n+\frac{j}{\eta_i}\right)} + \boldsymbol{x}^{\left(n+\frac{j+1}{\eta_i}\right)}\right)\cdot \tag{2.45b}$$

It is easy to show that, for any symmetric matrix $\boldsymbol{S}$, the jump and average operators obey the relationship

$$[\![\boldsymbol{x}^{(n)}]\!]^{\mathrm{T}}\boldsymbol{S}\left\langle\!\left\langle\boldsymbol{x}^{(n)}\right\rangle\!\right\rangle = \frac{1}{2}\left[\!\!\left[\boldsymbol{x}^{(n)\mathrm{T}}\boldsymbol{S}\boldsymbol{x}^{(n)}\right]\!\!\right]\cdot \tag{2.46}$$

A similar relation holds for $[\cdot]_i$ and $\langle\cdot\rangle_i$. It is important to note that the jump and average operators are linear. That is, for any $\alpha, \beta \in \mathbb{R}$ we have

$$[\![\alpha\boldsymbol{x} + \beta\boldsymbol{y}]\!] = \alpha[\![\boldsymbol{x}]\!] + \beta[\![\boldsymbol{y}]\!] \text{ and} \tag{2.47a}$$

$$\langle\!\langle\alpha\boldsymbol{x} + \beta\boldsymbol{y}\rangle\!\rangle = \alpha\langle\!\langle\boldsymbol{x}\rangle\!\rangle + \beta\langle\!\langle\boldsymbol{y}\rangle\!\rangle\cdot \tag{2.47b}$$

We shall call a sequence of vectors $\{\boldsymbol{x}^{(n)}\}_{n=0}^{\infty}$ to be bounded $\forall n$ if there exists a real number $0 < M < +\infty$ such that

$$\left\|\boldsymbol{x}^{(n)}\right\| < M \quad \forall n\cdot \tag{2.48}$$

For convenience, we shall use $\boldsymbol{A}_i$ to denote

$$\boldsymbol{A}_i := \boldsymbol{M}_i + (\Delta t_i)^2\left(\beta_i - \frac{\gamma_i}{2}\right)\boldsymbol{K}_i\cdot \tag{2.49}$$

The critical time-step $\Delta t_i^{\mathrm{crit}} \geq 0$ in the i-th subdomain is the maximum time-step for which the matrix $\boldsymbol{A}_i$ is positive definite. It should be emphasized that $\Delta t_i^{\mathrm{crit}}$ is the critical subdomain time-step assuming that there is no coupling between subdomains, which can be easily calculated. Let $\omega_i^{\max}$ be the maximum eigenvalue of the generalized eigenvalue problem for the i-th subdomain. That is,

$$\omega_i^2 \boldsymbol{M}_i \boldsymbol{x}_i = \boldsymbol{K}_i \boldsymbol{x}_i, \tag{2.50}$$

where $\boldsymbol{x}_i$ is the corresponding eigenvector. Then the critical time-step for the i-th subdomain can be written as

$$\Delta t_i^{\mathrm{crit}} = \begin{cases} +\infty & \text{for } 2\beta_i \geq \gamma_i \geq 1/2 \\ \frac{1}{\omega_i^{\max}\sqrt{\gamma_i/2 - \beta_i}} & \text{for } \gamma_i \geq 1/2 \text{ and } \beta_i < \gamma_i/2 \end{cases} . \tag{2.51}$$

We shall choose the subdomain time-step to be smaller than the corresponding critical time-step for the subdomain. That is,

$$\Delta t_i < \Delta t_i^{\mathrm{crit}}. \tag{2.52}$$

A detailed discussion on the critical time-steps for Newmark family of time integrators can be found in references [31, 33]. For Newmark family of time stepping schemes, it is easy to check that

$$\left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i = \Delta t_i \left(\left\langle \boldsymbol{a}_i^{\left(n+\frac{j}{\eta_i}\right)} \right\rangle_i + \left(\gamma_i - \frac{1}{2}\right) \left[\boldsymbol{a}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i \right) \text{ and} \tag{2.53a}$$

$$\left[\boldsymbol{d}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i = \Delta t_i \left\langle \boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)} \right\rangle_i + \Delta t_i^2 \left(\beta_i - \frac{\gamma_i}{2}\right) \left[\boldsymbol{a}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i . \tag{2.53b}$$

Theorem 1 *If $\Delta t_i < \Delta t_i^{\mathrm{crit}}$ in all subdomains, then the velocity and acceleration vectors for all subdomains are bounded $\forall n$ under the proposed multi-time-step coupling method.*

Proof 1 *Using the governing equation for the i-th subdomain, and the linear interpolation of the Lagrange multiplier, we obtain the equation*

$$\boldsymbol{M}_i \left[\boldsymbol{a}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i + \boldsymbol{K}_i \left[\boldsymbol{d}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i = \frac{1}{\eta_i}\boldsymbol{C}_i^{\mathrm{T}} \left\llbracket \boldsymbol{\lambda}^{(n)} \right\rrbracket \cdot \tag{2.54}$$

Using equation (2.53b)*, the above equation can be rewritten as*

$$\boldsymbol{A}_i \left[\boldsymbol{a}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i + \Delta t_i \boldsymbol{K}_i \left\langle \boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right\rangle_i = \frac{1}{\eta_i}\boldsymbol{C}_i^{\mathrm{T}} \left\llbracket \boldsymbol{\lambda}^{(n)} \right\rrbracket \cdot \tag{2.55}$$

Premultiplying both sides by $\left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i$ *and using equation* (2.53a)*, we obtain*

$$\begin{aligned}\Delta t_i \left\langle \boldsymbol{a}_i^{\left(n+\frac{j}{\eta_i}\right)}\right\rangle_i^{\mathrm{T}} \boldsymbol{A}_i \left[\boldsymbol{a}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i + \Delta t_i \left(\gamma_i - \frac{1}{2}\right)\left[\boldsymbol{a}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i^{\mathrm{T}} \boldsymbol{A}_i \left[\boldsymbol{a}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i \\ + \Delta t_i \left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i^{\mathrm{T}} \boldsymbol{K}_i \left\langle \boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right\rangle_i = \frac{1}{\eta_i} \left\llbracket \boldsymbol{\lambda}^{(n)} \right\rrbracket^{\mathrm{T}} \boldsymbol{C}_i \left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i \cdot\end{aligned} \tag{2.56}$$

Since $\gamma \geq \frac{1}{2}$ *and* $\boldsymbol{A}_i$ *is positive definite (as* $\Delta t_i < \Delta t_i^{\mathrm{crit}}$*), we can conclude that*

$$\Delta t_i \left\langle \boldsymbol{a}_i^{\left(n+\frac{j}{\eta_i}\right)}\right\rangle_i^{\mathrm{T}} \boldsymbol{A}_i \left[\boldsymbol{a}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i + \Delta t_i \left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i^{\mathrm{T}} \boldsymbol{K}_i \left\langle \boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right\rangle_i \leq \frac{1}{\eta_i} \left\llbracket \boldsymbol{\lambda}^{(n)} \right\rrbracket^{\mathrm{T}} \boldsymbol{C}_i \left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i \cdot \tag{2.57}$$

Noting that $\Delta t = \eta_i \Delta t_i$*, and the matrices* $\boldsymbol{A}_i$ *and* $\boldsymbol{K}_i$ *are symmetric, we obtain*

$$\frac{\Delta t}{2}\left[\boldsymbol{a}_i^{\left(n+\frac{j}{\eta_i}\right)^{\mathrm{T}}} \boldsymbol{A}_i \boldsymbol{a}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i + \frac{\Delta t}{2}\left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)^{\mathrm{T}}} \boldsymbol{K}_i \boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i \leq \left\llbracket \boldsymbol{\lambda}^{(n)} \right\rrbracket^{\mathrm{T}} \boldsymbol{C}_i \left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i \cdot \tag{2.58}$$

By summing over j $(j = 1, \cdots, \eta_i)$ *we obtain*

$$\frac{\Delta t}{2}\left\llbracket \boldsymbol{a}_i^{(n)^{\mathrm{T}}} \boldsymbol{A}_i \boldsymbol{a}_i^{(n)} + \boldsymbol{v}_i^{(n)^{\mathrm{T}}} \boldsymbol{K}_i \boldsymbol{v}_i^{(n)} \right\rrbracket \leq \left\llbracket \lambda^{(n)} \right\rrbracket^{\mathrm{T}} \sum_{i=1}^{S} \boldsymbol{C}_i \left\llbracket \boldsymbol{v}_i^{(n)} \right\rrbracket \cdot \tag{2.59}$$

Summing over i $(i = 1, \cdots, S)$ *and using the continuity of velocities at system time-steps, we obtain the inequality*

$$\sum_{i=1}^{S} \left\llbracket \boldsymbol{a}_i^{(n)^{\mathrm{T}}} \boldsymbol{A}_i \boldsymbol{a}_i^{(n)} + \boldsymbol{v}_i^{(n)^{\mathrm{T}}} \boldsymbol{K}_i \boldsymbol{v}_i^{(n)} \right\rrbracket \leq 0 \cdot \tag{2.60}$$

This further implies that

$$\begin{aligned}\sum_{i=1}^{S}\left(\boldsymbol{a}_i^{(n+1)^{\mathrm{T}}}\boldsymbol{A}_i\boldsymbol{a}_i^{(n+1)} + \boldsymbol{v}_i^{(n+1)^{\mathrm{T}}}\boldsymbol{K}_i\boldsymbol{v}_i^{(n+1)}\right) &\le \sum_{i=1}^{S}\left(\boldsymbol{a}_i^{(n)^{\mathrm{T}}}\boldsymbol{A}_i\boldsymbol{a}_i^{(n)} + \boldsymbol{v}_i^{(n)^{\mathrm{T}}}\boldsymbol{K}_i\boldsymbol{v}_i^{(n)}\right) \\ &\le \ldots \le \sum_{i=1}^{S}\left(\boldsymbol{a}_i^{(0)^{\mathrm{T}}}\boldsymbol{A}_i\boldsymbol{a}_i^{(0)} + \boldsymbol{v}_i^{(0)^{\mathrm{T}}}\boldsymbol{K}_i\boldsymbol{v}_i^{(0)}\right)\cdot\end{aligned} \tag{2.61}$$

Since the matrices $\boldsymbol{A}_i$ $(i = 1, \cdots, S)$ are positive definite, the matrices $\boldsymbol{K}_i$ $(i = 1, \cdots, S)$ are positive semidefinite, and the vectors $\boldsymbol{v}_i^{(0)}$ and $\boldsymbol{a}_i^{(0)}$ are bounded, one can conclude that the vectors $\boldsymbol{a}_i^{(n)}$ and $\boldsymbol{v}_i^{(n)}$ are bounded $\forall n$ and for all subdomains.

Remark 4 *Strictly speaking, in the above proof, one can only conclude that $\boldsymbol{v}_i^{(n)}$ are bounded except for vectors that have a component in the null space of $\boldsymbol{K}_i$. This is the case even for the undecomposed case (i.e., no coupling) under the energy method.*

Influence of perturbations under the proposed coupling method

We shall perform the analysis assuming no subcycling. We will follow a procedure similar to the one presented in [47] for differential-algebraic equations. We shall begin with the original system of equations over a (system) time-step as

$$\boldsymbol{M}_i\boldsymbol{a}_i^{(n+1)} + \boldsymbol{K}_i\boldsymbol{d}_i^{(n+1)} = \boldsymbol{f}_i^{(n+1)} + \boldsymbol{C}_i^{\mathrm{T}}\boldsymbol{\lambda}^{(n+1)}, \tag{2.62a}$$

$$\boldsymbol{v}_i^{(n+1)} = \boldsymbol{v}_i^{(n)} + \Delta t\left((1-\gamma_i)\,\boldsymbol{a}_i^{(n)} + \gamma_i\boldsymbol{a}_i^{(n+1)}\right), \tag{2.62b}$$

$$\boldsymbol{d}_i^{(n+1)} = \boldsymbol{d}_i^{(n)} + \Delta t\boldsymbol{v}_i^{(n)} + \frac{\Delta t^2}{2}\left((1-2\beta_i)\,\boldsymbol{a}_i^{(n)} + 2\beta_i\boldsymbol{a}_i^{(n+1)}\right) \text{ and} \tag{2.62c}$$

$$\sum_{i=1}^{S}\boldsymbol{C}_i\boldsymbol{v}_i^{(n+1)} = \boldsymbol{0}\cdot \tag{2.62d}$$

Now consider the perturbed system

$$\boldsymbol{M}_i\widehat{\boldsymbol{a}}_i^{(n+1)} + \boldsymbol{K}_i\widehat{\boldsymbol{d}}_i^{(n+1)} = \boldsymbol{f}_i^{(n+1)} + \boldsymbol{C}_i^{\mathrm{T}}\widehat{\boldsymbol{\lambda}}^{(n+1)}, \tag{2.63a}$$

$$\widehat{\boldsymbol{v}}_i^{(n+1)} = \widehat{\boldsymbol{v}}_i^{(n)} + \Delta t\left((1-\gamma_i)\,\widehat{\boldsymbol{a}}_i^{(n)} + \gamma_i\widehat{\boldsymbol{a}}_i^{(n+1)}\right) + \Delta t\boldsymbol{\varepsilon}_{v_i}, \tag{2.63b}$$

$$\widehat{\boldsymbol{d}}_i^{(n+1)} = \widehat{\boldsymbol{d}}_i^{(n)} + \Delta t\widehat{\boldsymbol{v}}_i^{(n)} + \frac{\Delta t^2}{2}\left((1-2\beta_i)\,\widehat{\boldsymbol{a}}_i^{(n)} + 2\beta_i\widehat{\boldsymbol{a}}_i^{(n+1)}\right) + \Delta t^2\boldsymbol{\varepsilon}_{d_i} \text{ and} \tag{2.63c}$$

$$\sum_{i=1}^{S}\boldsymbol{C}_i\widehat{\boldsymbol{v}}_i^{(n+1)} = \boldsymbol{\varepsilon}_\lambda. \tag{2.63d}$$

where $\boldsymbol{\varepsilon}_{v_i}$, $\boldsymbol{\varepsilon}_{d_i}$ and $\boldsymbol{\varepsilon}_\lambda$ are, respectively, the perturbations to the original system of equations (2.62a)–(2.62d). The solution to this perturbed system of equations will be $\widehat{\boldsymbol{a}}_i^{(n+1)}$, $\widehat{\boldsymbol{v}}_i^{(n+1)}$, $\widehat{\boldsymbol{d}}_i^{(n+1)}$ and $\widehat{\boldsymbol{\lambda}}^{(n+1)}$. For convenience, we shall define the quantities

$$\delta\boldsymbol{a}_i^{(n+1)} := \widehat{\boldsymbol{a}}_i^{(n+1)} - \boldsymbol{a}_i^{(n+1)}, \tag{2.64a}$$

$$\delta\boldsymbol{v}_i^{(n+1)} := \widehat{\boldsymbol{v}}_i^{(n+1)} - \boldsymbol{v}_i^{(n+1)}, \tag{2.64b}$$

$$\delta\boldsymbol{d}_i^{(n+1)} := \widehat{\boldsymbol{d}}_i^{(n+1)} - \boldsymbol{d}_i^{(n+1)} \text{ and} \tag{2.64c}$$

$$\delta\boldsymbol{\lambda}^{(n+1)} := \widehat{\boldsymbol{\lambda}}^{(n+1)} - \boldsymbol{\lambda}^{(n+1)}. \tag{2.64d}$$

By subtracting equation (2.62a) from equation (2.63a) we obtain

$$\boldsymbol{M}_i\delta\boldsymbol{a}_i^{(n+1)} + \boldsymbol{K}_i\delta\boldsymbol{d}_i^{(n+1)} = \boldsymbol{C}_i^{\mathrm{T}}\delta\boldsymbol{\lambda}^{(n+1)}. \tag{2.65}$$

Using equations (2.62c) and (2.63c), the above equation can be written as

$$\delta\boldsymbol{a}_i^{(n+1)} + \boldsymbol{B}_i^{-1}\boldsymbol{K}_i\left(\delta\boldsymbol{d}_i^{(n)} + \Delta t\delta\boldsymbol{v}_i^{(n)} + \Delta t^2(1/2-\beta_i)\delta\boldsymbol{a}_i^{(n)}\right) = \boldsymbol{B}_i^{-1}\boldsymbol{C}_i^{\mathrm{T}}\delta\boldsymbol{\lambda}^{(n+1)} - \Delta t^2\boldsymbol{B}_i^{-1}\boldsymbol{K}_i\boldsymbol{\varepsilon}_{d_i}, \tag{2.66}$$

where the matrix $\boldsymbol{B}_i$ has been defined as

$$\boldsymbol{B}_i := \boldsymbol{M}_i + \beta_i\Delta t^2\boldsymbol{K}_i. \tag{2.67}$$

The operation $\boldsymbol{B}_i^{-1}$ in equation (2.66) is justified as the matrix is positive definite and hence invertible. By multiplying both sides of equation (2.66) by $\gamma_i \Delta t$ and using equations (2.62b) and (2.63b), one can arrive at the equation

$$\delta \boldsymbol{v}_i^{(n+1)} - \delta \boldsymbol{v}_i^{(n)} - (1-\gamma_i)\Delta t \delta \boldsymbol{a}_i^{(n)} - \Delta t \boldsymbol{\varepsilon}_{v_i} + \gamma_i \Delta t \boldsymbol{B}_i^{-1} \boldsymbol{K}_i \left(\delta \boldsymbol{d}_i^{(n)} + \Delta t \delta \boldsymbol{v}_i^{(n)} + \Delta t^2 (1/2 - \beta_i) \delta \boldsymbol{a}_i^{(n)} \right)$$

$$= \gamma_i \Delta t \boldsymbol{B}_i^{-1} \boldsymbol{C}_i^{\mathrm{T}} \delta \boldsymbol{\lambda}^{(n+1)} - \gamma_i \Delta t^3 \boldsymbol{B}_i^{-1} \boldsymbol{K}_i \boldsymbol{\varepsilon}_{d_i}. \tag{2.68}$$

We shall assume that $\sum_{i=1}^{S} \boldsymbol{C}_i \delta \boldsymbol{v}_i^{(n)} = \boldsymbol{0}$. That is, the constraint is exactly satisfied at the n-th time level. Premultiplying both sides by $\boldsymbol{C}_i$, summing over i (i.e., the number of subdomains), and using equations (2.62d) and (2.63d); one can arrive at the equation

$$\begin{aligned} \boldsymbol{\varepsilon}_\lambda - \Delta t \sum_{i=1}^{S} (1-\gamma_i) \boldsymbol{C}_i \delta \boldsymbol{a}_i^{(n)} - \Delta t \sum_{i=1}^{S} \boldsymbol{C}_i \boldsymbol{\varepsilon}_{v_i} & \\ + \Delta t \sum_{i=1}^{S} \gamma_i \boldsymbol{C}_i \boldsymbol{B}_i^{-1} \boldsymbol{K}_i & \left(\delta \boldsymbol{d}_i^{(n)} + \Delta t \delta \boldsymbol{v}_i^{(n)} + \Delta t^2 (1/2 - \beta_i) \delta \boldsymbol{a}_i^{(n)} \right) \\ = \Delta t & \left(\sum_{i=1}^{S} \gamma_i \boldsymbol{C}_i \boldsymbol{B}_i^{-1} \boldsymbol{C}_i^{\mathrm{T}} \right) \delta \boldsymbol{\lambda}^{(n+1)} - \Delta t^3 \sum_{i=1}^{S} \gamma_i \boldsymbol{C}_i \boldsymbol{B}_i^{-1} \boldsymbol{K}_i \boldsymbol{\varepsilon}_{d_i}. \end{aligned} \tag{2.69}$$

By taking norm on both sides and invoking triangle inequality, one can arrive at the following estimate for $\delta \boldsymbol{\lambda}^{(n+1)}$ as

$$\| \delta \boldsymbol{\lambda}^{(n+1)} \| \leq C_\lambda \left(\frac{1}{\Delta t} \| \boldsymbol{\varepsilon}_\lambda \| + \sum_{i=1}^{S} \left(\| \boldsymbol{\varepsilon}_{v_i} \| + \Delta t^2 \| \boldsymbol{\varepsilon}_{d_i} \| + \| \delta \boldsymbol{a}_i^{(n)} \| + \Delta t \| \delta \boldsymbol{v}_i^{(n)} \| + \| \delta \boldsymbol{d}_i^{(n)} \| \right) \right), \tag{2.70}$$

where C_λ is a constant. Following a similar procedure for displacements, velocities, and accelerations we obtain

$$\|\delta \boldsymbol{d}_i^{(n+1)}\| \leq C_d \left(\|\delta \boldsymbol{d}_i^{(n)}\| + \Delta t \|\delta \boldsymbol{v}_i^{(n)}\| + \Delta t^2 \|\boldsymbol{\varepsilon}_{d_i}\| + \Delta t \|\boldsymbol{\varepsilon}_\lambda\| + \sum_{i=1}^{S} \left(\Delta t^2 \|\delta \boldsymbol{a}_i^{(n)}\| + \Delta t \|\boldsymbol{\varepsilon}_{v_i}\| \right) \right), \tag{2.71}$$

$$\|\delta \boldsymbol{v}_i^{(n+1)}\| \leq C_v \left(\|\delta \boldsymbol{v}_i^{(n)}\| + \|\boldsymbol{\varepsilon}_\lambda\| + \sum_{i=1}^{S} \left(\Delta t \|\delta \boldsymbol{a}_i^{(n)}\| + \Delta t \|\delta \boldsymbol{d}_i^{(n)}\| + \Delta t^3 \|\boldsymbol{\varepsilon}_{d_i}\| + \Delta t \|\boldsymbol{\varepsilon}_{v_i}\| \right) \right) \text{ and} \tag{2.72}$$

$$\|\delta \boldsymbol{a}_i^{(n+1)}\| \leq C_a \left(\frac{1}{\Delta t} \|\boldsymbol{\varepsilon}_\lambda\| + \Delta t \|\delta \boldsymbol{v}_i^{(n)}\| + \sum_{i=1}^{S} \left(\|\delta \boldsymbol{a}_i^{(n)}\| + \|\delta \boldsymbol{d}_i^{(n)}\| + \Delta t^2 \|\boldsymbol{\varepsilon}_{d_i}\| + \|\boldsymbol{\varepsilon}_{v_i}\| \right) \right), \tag{2.73}$$

where C_d, C_v and C_a are constants. From the above estimate (2.70), one can see that a perturbation in the constraint, $\boldsymbol{\varepsilon}_\lambda$, leads to an amplification by $1/\Delta t$ in the Lagrange multiplier. On the other hand, the perturbations in the variables $\boldsymbol{d}_i$ and $\boldsymbol{v}_i$ lead to (at most) linear growth in the Lagrange multiplier. Clearly, the estimate for the proposed coupling method under *no* subcycling follows the typical behavior of differential-algebraic equations. An extension of this study to include subcycling will require a more involved and careful analysis, and is beyond the scope of this chapter.

On drifts in interface displacement and acceleration vectors

In a time continuous setting, enforcing the continuity of either displacements, velocities or accelerations are all mathematically equivalent. However, in a numerical setting this equivalence will not hold, and the numerical performance will depend on the type of the constraint that is being enforced. As mentioned in the previous sections, we employ the continuity of velocities at the subdomain interface at every system time-step (which we referred to as the $\boldsymbol{v}$-continuity). This may lead to drift in the displacements and the accelerations along the subdomain interface. We now derive bounds on these drifts, which could serve as a valuable check for the correctness of a numerical implementation.

For the present study, we shall assume that there is no subcycling (i.e., $\eta_i = 1$), and no mixed methods are employed (i.e., $\beta_i = \beta$, $\gamma_i = \gamma$). The errors due to finite precision arithmetic and their numerical propagation are ignored. For convenience, let us denote the drift in the displacements and the drift in the accelerations along the subdomain interface as

$$\boldsymbol{a}_{\mathrm{drift}}^{(n)} := \sum_{i=1}^{S} \boldsymbol{C}_i \boldsymbol{a}_i^{(n)} \text{ and} \tag{2.74a}$$

$$\boldsymbol{d}_{\mathrm{drift}}^{(n)} := \sum_{i=1}^{S} \boldsymbol{C}_i \boldsymbol{d}_i^{(n)}. \tag{2.74b}$$

Basically, the drift in displacements (or accelerations) is the measure of error in meeting the continuity of displacements (or accelerations) across the subdomain interface. The drifts satisfy the relations

$$\boldsymbol{a}_{\mathrm{drift}}^{(n+1)} = \left(1 - \frac{1}{\gamma}\right) \boldsymbol{a}_{\mathrm{drift}}^{(n)} \text{ and} \tag{2.75a}$$

$$\boldsymbol{d}_{\mathrm{drift}}^{(n+1)} = \boldsymbol{d}_{\mathrm{drift}}^{(n)} + \left(\frac{1}{2} - \frac{\beta}{\gamma}\right) \Delta t^2 \boldsymbol{a}_{\mathrm{drift}}^{(n)}. \tag{2.75b}$$

Thus, one can draw the conclusions about the drifts:

(i) For numerical stability of a time-stepping scheme under Newmark family, $\gamma \geq 1/2$. Therefore,

$$\|\boldsymbol{a}_{\mathrm{drift}}^{(n+1)}\| \leq \|\boldsymbol{a}_{\mathrm{drift}}^{(n)}\|. \tag{2.76}$$

One has the equality only when $\gamma = 1/2$ (e.g., Newmark average acceleration scheme, central difference scheme, Newmark linear acceleration scheme).

(ii) For any time stepping scheme with $\gamma = 2\beta$ (e.g., Newmark average acceleration scheme) we have

$$\boldsymbol{d}_{\text{drift}}^{(n+1)} = \boldsymbol{d}_{\text{drift}}^{(n)} \quad n = 1, 2, \cdots \tag{2.77}$$

The above claims will be numerically substantiated in a subsequent section using the test problem outlined in subsection 2.9.

2.6 Split Degree-Of-Freedom Lumped Parameter System

Consider a split agree of freedom whose motion can be described by the following system of ordinary differential/algebraic equations

$$m_A \ddot{u}_A(t) + k_A u_A(t) = f_A(t) + \lambda(t), \tag{2.78a}$$

$$m_B \ddot{u}_B(t) + k_B u_B(t) = f_B(t) - \lambda(t), \text{ and} \tag{2.78b}$$

$$\dot{u}_A(t) - \dot{u}_B(t) = 0\cdot \tag{2.78c}$$

The following parameters are used: $m_A = 0.1$, $m_B = 0.005$, and the stiffness of springs are $k_A = 2.5$ and $k_B = 50$. The subdomain time-steps are taken as $\Delta t_A = 0.02$ and $\Delta t_B = 0.005$. The system time-step is taken as $\Delta t = 0.02$. The values of the external forces are taken to be zero, that is $f_A = 0$ and $f_B = 0$. The initial conditions are $d_0 = 0.1$ and $v_0 = 1.0$. The problem is solved over a time interval of $[0, 0.5]$. In all the cases, Newmark average acceleration scheme is used in all the subdomains. The

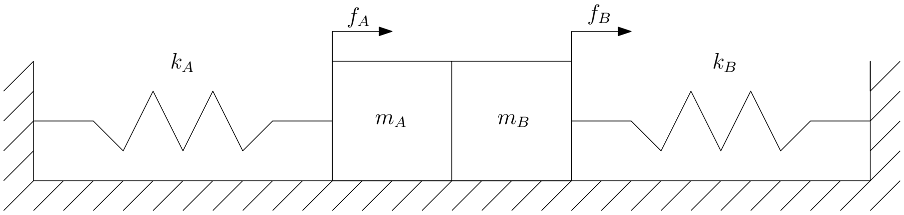


Figure 2.3: A pictorial description of the split degree-of-freedom (SDOF) lumped parameter system: The masses A and B slide on a frictionless surface.

resulting numerical results for kinematic variables are shown in Figure 2.4. Since the external forces applied are constant ($f_A = f_B = 0$) the PH method and the proposed coupling methods yield the same results. The GC method suffers from excessive damping and fails to match the exact results. Similar observation can be made about the interface force as well as total physical energy of the system, as shown in Figure 2.5.

2.7 On Energy Conserving vs. Energy Preserving Coupling

In this section we address the energy preserving and energy conserving properties of the proposed multi-time-step coupling method. Two different notions of energy preserving will be considered. In particular, the following questions will be answered:

(a) Does the coupling method add or extract energy from the system of subdomains in comparison with the case of no coupling?

(b) Do the interface forces perform net work?

(c) Under what conditions does the coupling method conserve the total energy of the system of subdomains?

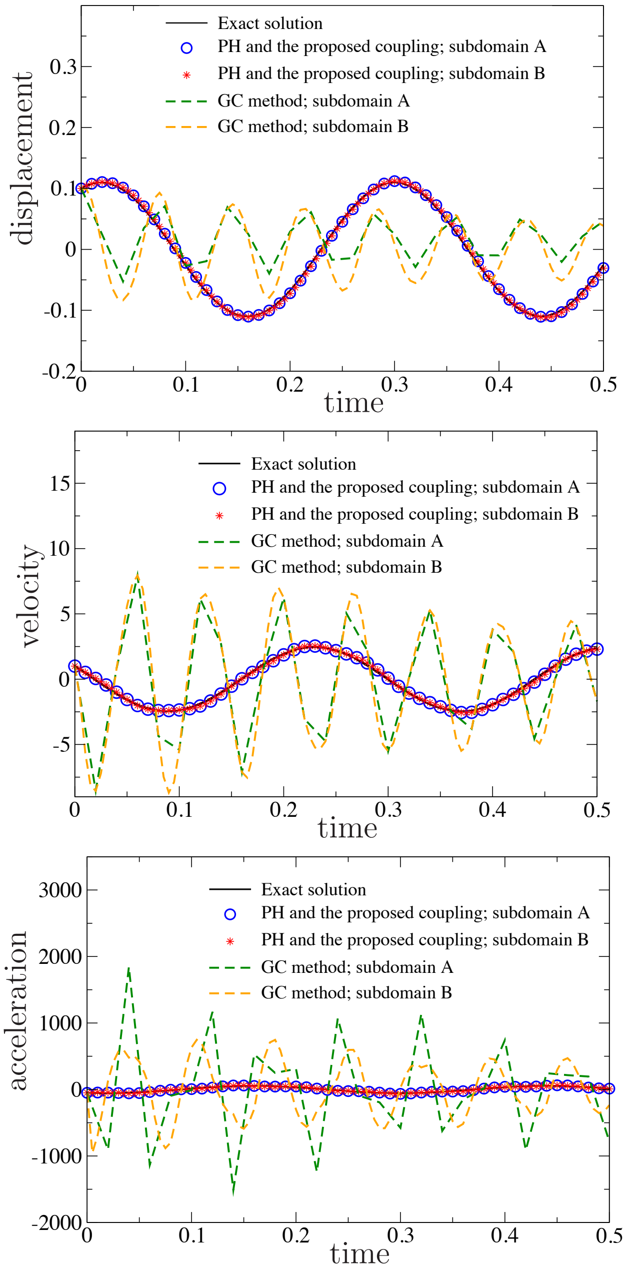


Figure 2.4: SDOF lumped parameter system: This figure compares the performance of the proposed coupling method with that of the GC and PH methods.

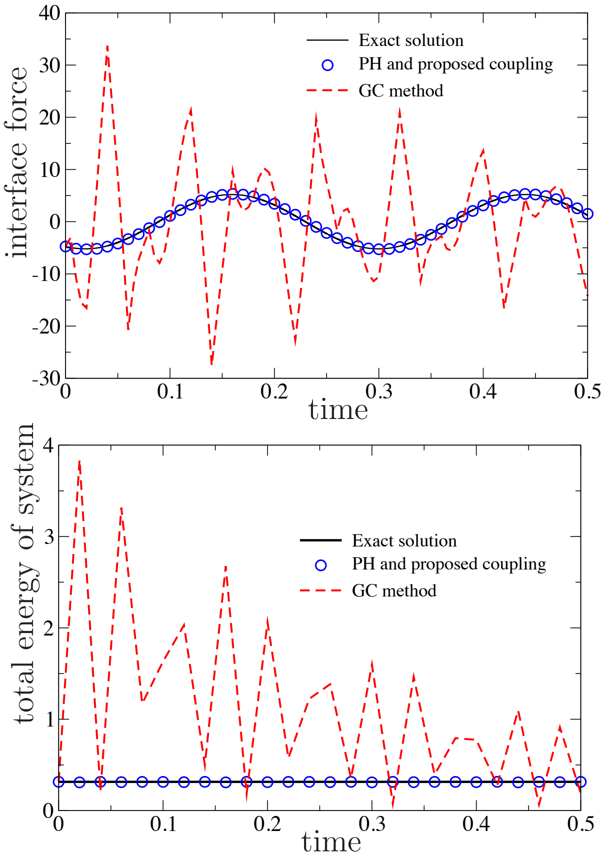


Figure 2.5: SDOF lumped parameter system: The top and bottom subfigures, respectively, show the interface force and total energy of the system. The numerical results under the GC method do not match with the analytical solution.

To this end, the kinetic energy and the potential energy of the i-th subdomain are, respectively, defined as

$$\mathcal{T}_i\left(\boldsymbol{v}_i\right) := \frac{1}{2}\boldsymbol{v}_i^{\mathrm{T}}\boldsymbol{M}_i\boldsymbol{v}_i \text{ and } \quad \mathcal{V}_i\left(\boldsymbol{d}_i\right) := \frac{1}{2}\boldsymbol{d}_i^{\mathrm{T}}\boldsymbol{K}_i\boldsymbol{d}_i\cdot \tag{2.79}$$

The total energy of the i-th subdomain is given by

$$\mathcal{E}_i\left(\boldsymbol{d}_i, \boldsymbol{v}_i\right) := \mathcal{T}_i\left(\boldsymbol{v}_i\right) + \mathcal{V}_i\left(\boldsymbol{d}_i\right)\cdot \tag{2.80}$$

The total energy of all the subdomains at the n-th (system) time level can be written as

$$\mathcal{E}^{(n)} \equiv \mathcal{E}\left(\boldsymbol{d}_1^{(n)}, \cdots, \boldsymbol{d}_S^{(n)}, \boldsymbol{v}_1^{(n)}, \cdots, \boldsymbol{v}_S^{(n)}\right) := \sum_{i=1}^{S} \mathcal{E}_i\left(\boldsymbol{d}_i^{(n)}, \boldsymbol{v}_i^{(n)}\right). \tag{2.81}$$

In the remainder of this section, we shall assume that the external forces are zero (i.e., $\boldsymbol{f}_i(t) = \boldsymbol{0}\ \forall i$). For the proposed multi-time-step method, one can derive the relation

$$\mathcal{E}^{(n+1)} - \mathcal{E}^{(n)} = \mathcal{E}_{\text{algorithm}}^{(n \to n+1)} + \mathcal{E}_{\text{interface}}^{(n \to n+1)}, \tag{2.82}$$

where $\mathcal{E}_{\text{algorithm}}^{(n \to n+1)}$ and $\mathcal{E}_{\text{interface}}^{(n \to n+1)}$ are, respectively, defined as

$$\begin{aligned}\mathcal{E}_{\text{algorithm}}^{(n \to n+1)} &:= -2 \sum_{i=1}^{S} \sum_{j=0}^{\eta_i - 1} \left(\gamma_i - \frac{1}{2}\right) \mathcal{V}_i\left(\left[\boldsymbol{d}_i^{\left(n + \frac{j}{\eta_i}\right)}\right]_i\right) - \Delta t^2 \sum_{i=1}^{S} \frac{1}{\eta_i^2}\left(\beta_i - \frac{\gamma_i}{2}\right) [\![\mathcal{T}_i\left(\boldsymbol{a}_i^{(n)}\right)]\!] \\ &\quad - \Delta t^2 \sum_{i=1}^{S} \frac{1}{\eta_i^2}\left(\beta_i - \frac{\gamma_i}{2}\right)(2\gamma_i - 1)\left(\sum_{j=0}^{\eta_i - 1} \mathcal{T}_i\left(\left[\boldsymbol{a}_i^{\left(n + \frac{j}{\eta_i}\right)}\right]_i\right)\right) \text{ and}\end{aligned} \tag{2.83}$$

$$\mathcal{E}_{\text{interface}}^{(n \to n+1)} := \sum_{i=1}^{S} \sum_{j=0}^{\eta_i - 1} \left((1 - \gamma_i)\,\boldsymbol{\lambda}^{\left(n + \frac{j}{\eta_i}\right)} + \gamma_i \boldsymbol{\lambda}^{\left(n + \frac{j+1}{\eta_i}\right)}\right)^{\mathrm{T}} \boldsymbol{C}_i \left[\boldsymbol{d}_i^{\left(n + \frac{j}{\eta_i}\right)}\right]_i. \tag{2.84}$$

If there is no subcycling in all the subdomains (i.e., $\eta_i = 1\ \forall i$), the above relationship can be simplified as

$$\begin{aligned}\mathcal{E}^{(n+1)} - \mathcal{E}^{(n)} = &\underbrace{-2 \sum_{i=1}^{S}\left(\gamma_i - \frac{1}{2}\right) \mathcal{V}_i\left([\![\boldsymbol{d}_i^{(n)}]\!]\right) - \Delta t^2 \sum_{i=1}^{S} \gamma_i (2\beta_i - \gamma_i)\, \mathcal{T}_i\left([\![\boldsymbol{a}_i^{(n)}]\!]\right)}_{\mathcal{E}_{\text{algorithmic}}^{(n \to n+1)}} \\ &+ \underbrace{\sum_{i=1}^{S}\left((1 - \gamma_i)\boldsymbol{\lambda}^{(n)} + \gamma_i \boldsymbol{\lambda}^{(n+1)}\right)^{\mathrm{T}} \boldsymbol{C}_i\, [\![\boldsymbol{d}_i^{(n)}]\!]}_{\mathcal{E}_{\text{interface}}^{(n \to n+1)}}.\end{aligned} \tag{2.85}$$

Energy preserving in the first sense

We shall call that the coupling method *preserves energy in the first sense* if the coupling neither adds nor extracts energy over a system time-step in comparison to that of no coupling. By no coupling, we mean that the problem (2.15) is solved without decomposing into subdomains (i.e., $S = 1$), no subcycling (i.e., $\eta_i = 1$), and no mixed methods (i.e., $\gamma_i = \gamma$ and $\beta_i = \beta \; \forall i$). We denote the total energy at integral time levels under no coupling as

$$\mathcal{E}^{(n)}_{\text{no coupling}} := \mathcal{T}^{(n)}_{\text{no coupling}} + \mathcal{V}^{(n)}_{\text{no coupling}}, \tag{2.86}$$

where

$$\mathcal{T}^{(n)}_{\text{no coupling}} := \frac{1}{2}\boldsymbol{v}^{(n)^{\mathrm{T}}}\boldsymbol{M}\boldsymbol{v}^{(n)}, \text{ and} \tag{2.87}$$

$$\mathcal{V}^{(n)}_{\text{no coupling}} := \frac{1}{2}\boldsymbol{d}^{(n)^{\mathrm{T}}}\boldsymbol{K}\boldsymbol{d}^{(n)}. \tag{2.88}$$

Mathematically, preserving energy in the first sense implies that

$$\mathcal{E}^{(n)} = \mathcal{E}^{(n)}_{\text{no coupling}} \quad \forall n. \tag{2.89}$$

The numerical solution presented in Figure 2.6 confirms that the proposed multi-time-step coupling method, in general, does not preserve energy in the first sense.

Remark 5 *It should be noted that many stable time stepping schemes under the Newmark family are dissipative [31]. That is,*

$$\mathcal{E}^{(n+1)}_{no\ coupling} < \mathcal{E}^{(n)}_{no\ coupling} \quad \forall n. \tag{2.90}$$

Only the Newmark average acceleration scheme $(\gamma = 1/2,\ \beta = 1/4)$ *under the Newmark family conserves energy for linear problems (e.g. linear elastodynamics). That is,*

$$\mathcal{E}^{(n+1)}_{no\ coupling} = \mathcal{E}^{(n)}_{no\ coupling} \quad \forall n. \tag{2.91}$$

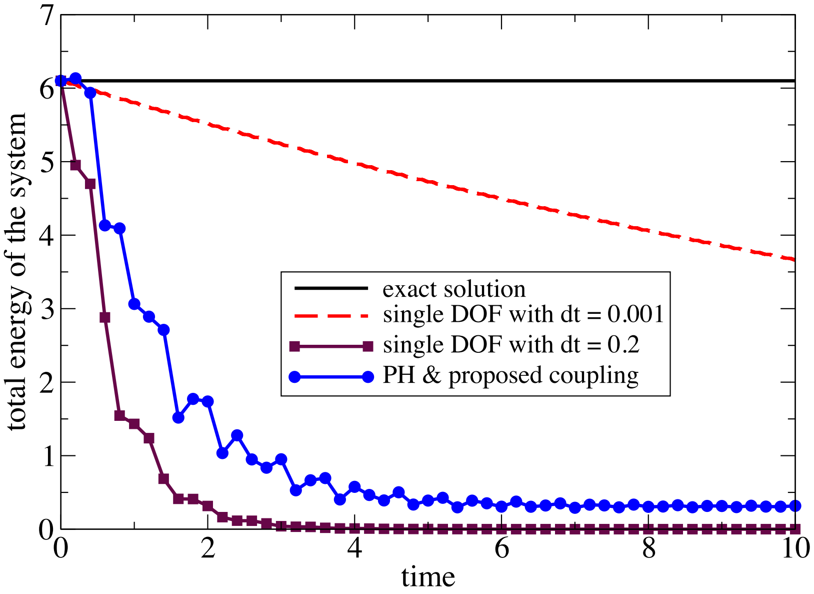


Figure 2.6: Energy preservation of multi-time-step methods: For comparison, numerical solutions for single degree of freedom (i.e., without splitting) are also presented for two different time-steps $\Delta t = 0.2$ and $\Delta t = 0.001$.

Energy preserving in the second sense

We shall call that the coupling method *preserves energy in the second sense* if the interface forces (i.e., the multipliers $\boldsymbol{\lambda}$) do not perform net work over a system time-step. That is,

$$\mathcal{E}_{\text{interface}}^{(n \to n+1)} = 0 \quad \forall n \cdot \tag{2.92}$$

In general, the proposed multi-time-step coupling method does not preserve energy even in the second sense. However, using equation (2.84), one can show that a sufficient condition for the coupling method to preserve energy in the second sense is to have $\gamma_i = \gamma \ \forall i$, $\gamma_i = 2\beta_i$, and no subcycling (i.e., $\eta_i = 1 \ \forall i$). This sufficient condition also guides one to construct a simple example that substantiates the claim that the proposed coupling method need not preserve the energy in the second sense. By choosing Newmark average acceleration scheme ($\gamma = 1/2,\ \beta = 1/4$) in all subdomains we will have $\mathcal{E}_{\text{algorithm}}^{(n\to n+1)} = 0\ \forall n$ whether the subcycling is present or not. This implies that the difference between $\mathcal{E}^{(n+1)}$ and $\mathcal{E}^{(n)}$ is solely due to $\mathcal{E}_{\text{interface}}^{(n\to n+1)}$. If there is no subcycling then the quantity $\mathcal{E}_{\text{interface}}^{(n\to n+1)}$ will also be zero. However, if there is subcycling then one can have

$$\mathcal{E}_{\text{interface}}^{(n\to n+1)} \neq 0\cdot \tag{2.93}$$

Based on the above reasoning, Figure 2.7 presents the numerical results to substantiates the above claim.

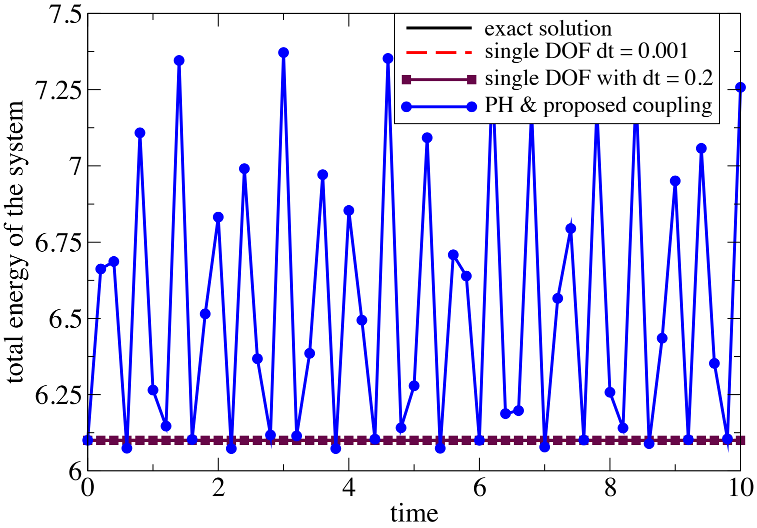


Figure 2.7: Energy preservation of multi-time-step methods: This figure illustrates that the proposed coupling method does not conserve energy if there is subcycling. The Newmark average acceleration method ($\beta = 0.25, \gamma = 0.5$) is employed in both subdomains.

Energy conserving

We shall say that the coupling method *conserves energy* exactly if

$$\mathcal{E}^{(n+1)} = \mathcal{E}^{(n)} \quad \forall n\cdot \tag{2.94}$$

Based on equation (2.85), a necessary and sufficient condition for the coupling method for conserve energy is

$$\mathcal{E}_{\text{algorithm}}^{(n\to n+1)} + \mathcal{E}_{\text{interface}}^{(n\to n+1)} = 0 \quad \forall n, \tag{2.95}$$

where $\mathcal{E}_{\text{algorithm}}^{(n\to n+1)}$ and $\mathcal{E}_{\text{interface}}^{(n\to n+1)}$ are, respectively, defined in equations (2.83) and (2.84). A sufficient condition can be written as

$$\mathcal{E}_{\text{algorithm}}^{(n\to n+1)} = 0 \quad \text{and} \quad \mathcal{E}_{\text{interface}}^{(n\to n+1)} = 0 \quad \forall n\cdot \tag{2.96}$$

The following theorem provides a way to achieve the above sufficient condition.

Theorem 2 *If all the subdomains employ the Newmark average acceleration scheme (i.e., $\gamma_i = 1/2$ and $\beta_i = 1/4$), and there is no subcycling (i.e., $\eta_i = 1 \; \forall i$), then the coupling method exactly conserves energy when $\boldsymbol{f}_i(t) = \boldsymbol{0} \; \forall i$.*

Proof 2 *This proof is a simple extension of the proof for single domain (i.e., without coupling). For Newmark average acceleration time stepping scheme using the identities*

$$\left[\!\left[\boldsymbol{d}_i^{(n)}\right]\!\right] = \Delta t \left\langle\!\left\langle \boldsymbol{v}_i^{(n)} \right\rangle\!\right\rangle \; \textit{and} \tag{2.97a}$$

$$\left[\!\left[\boldsymbol{v}_i^{(n)}\right]\!\right] = \Delta t \left\langle\!\left\langle \boldsymbol{a}_i^{(n)} \right\rangle\!\right\rangle \cdot \tag{2.97b}$$

The governing equation for i-th subdomain implies that

$$\boldsymbol{M}_i \left\langle\!\left\langle \boldsymbol{a}_i^{(n)} \right\rangle\!\right\rangle + \boldsymbol{K}_i \left\langle\!\left\langle \boldsymbol{d}_i^{(n)} \right\rangle\!\right\rangle = \boldsymbol{C}_i^{\mathrm{T}} \left\langle\!\left\langle \boldsymbol{\lambda}^{(n)} \right\rangle\!\right\rangle \cdot \tag{2.98}$$

Premultiplying by $\left\langle\!\left\langle \boldsymbol{v}_i^{(n)} \right\rangle\!\right\rangle$*, using the above relations* (2.97a)–(2.97b)*, summing over all the subdomains and using the continuity of velocities, we get*

$$\sum_{i=1}^{S} \left\langle\!\left\langle \boldsymbol{v}_i^{(n)} \right\rangle\!\right\rangle^{\mathrm{T}} \boldsymbol{M}_i \left[\!\left[\boldsymbol{v}_i^{(n)} \right]\!\right] + \sum_{i=1}^{S} \left[\!\left[\boldsymbol{d}_i^{(n)} \right]\!\right]^{\mathrm{T}} \boldsymbol{K}_i \left\langle\!\left\langle \boldsymbol{d}_i^{(n)} \right\rangle\!\right\rangle = \sum_{i=1}^{S} \left\langle\!\left\langle \boldsymbol{v}_i^{(n)} \right\rangle\!\right\rangle^{\mathrm{T}} \boldsymbol{C}_i^{\mathrm{T}} \left\langle\!\left\langle \boldsymbol{\lambda}^{(n)} \right\rangle\!\right\rangle$$

$$= \left\langle\!\left\langle \boldsymbol{\lambda}^{(n)} \right\rangle\!\right\rangle^{\mathrm{T}} \sum_{i=1}^{S} \boldsymbol{C}_i \left\langle\!\left\langle \boldsymbol{v}_i^{(n)} \right\rangle\!\right\rangle = 0\cdot \quad (2.99)$$

Using the symmetry of the matrices $\boldsymbol{M}_i$ *and* $\boldsymbol{K}_i$*, and noting the linearity of the jump operator, we have*

$$\left[\!\left[\frac{1}{2} \sum_{i=1}^{S} \boldsymbol{v}_i^{(n)\,\mathrm{T}} \boldsymbol{M}_i \boldsymbol{v}_i^{(n)} + \frac{1}{2} \sum_{i=1}^{S} \boldsymbol{d}_i^{(n)\,\mathrm{T}} \boldsymbol{K}_i \boldsymbol{d}_i^{(n)} \right]\!\right] = 0, \quad (2.100)$$

which shows that the total energy is exactly conserved over a system time-step.

It is noteworthy that if $\gamma_i = \gamma > 1/2$ and $\beta_i = \gamma_i/2$, and there is no subcycling then we have

$$\mathcal{E}^{(n+1)} - \mathcal{E}^{(n)} = -2\left(\gamma - \frac{1}{2}\right) \sum_{i=1}^{S} \mathcal{V}_i\left(\left[\!\left[\boldsymbol{d}_i^{(n)} \right]\!\right] \right) < 0, \quad (2.101)$$

which implies that the coupling method will be strictly energy decaying. As mentioned in Section 2.2, $\gamma < 1/2$ is not in the allowable range of values under the Newmark family of time integrators because of numerical stability.

Is the PH method really energy preserving?

We are now set nicely to examine the claim made in Reference [30] that the PH method preserves energy. In the absence of external forces, the proposed coupling method is the same as the PH method. Therefore, based on the earlier discussion in this section, the PH method is neither energy conserving nor energy preserving in both first and second senses. The source of error that led to the false claim is due to the use of an inappropriate definition for the work done by the interface. Using the notation introduced in this chapter, the expression considered in [30, equations (58) and (61)] for work done by the interface can be written as

$$\frac{1}{\Delta t_A} \left[\!\left[\boldsymbol{v}_A^{(n)} \right]\!\right]^{\mathrm{T}} \boldsymbol{C}_A^{\mathrm{T}} \left[\!\left[\boldsymbol{\lambda}^{(n)} \right]\!\right] + \frac{1}{\Delta t_B} \sum_{j=1}^{\eta_B} \left[\boldsymbol{v}_B^{\left(n+\frac{j-1}{\eta_B}\right)} \right]_B^{\mathrm{T}} \boldsymbol{C}_B^{\mathrm{T}} \left[\boldsymbol{\lambda}^{\left(n+\frac{j-1}{\eta_B}\right)} \right]_B \cdot \qquad (2.102)$$

But the above expression is *not* appropriate for the work done by the interface forces. A comment is also warranted on the numerical results presented in [30, Figures 8 & 11], which have been used to support their claim. For the chosen test problems, these figures report that E_{total} is constant under the PH method where

$$E_{\mathrm{total}} := \frac{1}{2} \boldsymbol{a}_A^{(n)^{\mathrm{T}}} \boldsymbol{A}_A \boldsymbol{a}_A^{(n)} + \frac{1}{2} \boldsymbol{v}_A^{(n)^{\mathrm{T}}} \boldsymbol{K}_A \boldsymbol{v}_A^{(n)} + \frac{1}{2} \boldsymbol{a}_B^{(n)^{\mathrm{T}}} \boldsymbol{A}_B \boldsymbol{a}_B^{(n)} + \frac{1}{2} \boldsymbol{v}_B^{(n)^{\mathrm{T}}} \boldsymbol{K}_B \boldsymbol{v}_B^{(n)} \cdot \qquad (2.103)$$

Recall that

$$\mathbf{A}_i := \boldsymbol{M}_i + \Delta t_i^2 \left(\beta_i - \frac{\gamma_i}{2} \right) \boldsymbol{K}_i \quad i = A, B \cdot \qquad (2.104)$$

The constant value for E_{total} has then been used to support that the PH method is energy preserving. To remove some misconceptions in the literature on the energy conserving property under multi-time-step coupling methods and to facilitate future developments, the following remarks on the nature of E_{total} are made:

(i) E_{total} is *not* equal to the physical total energy of the system (i.e., the sum of kinetic and potential energies). Hence, the preservation of E_{total} does not imply the preservation of the physical total energy of the system.

(ii) Even this quantity will not be constant under the PH method if the Newmark parameter $\gamma \neq 1/2$ even in one subdomain. The result shown in reference [30, Figures 8 & 11] used $\gamma = 1/2$ in all the subdomains.

(iii) It should be noted that E_{total} can be constant even for a non-zero constant external force, which will not be the case with the physical total energy.

(iv) If preservation of such a quantity is essential for some reason, it should be noted that the proposed coupling method will also preserve E_{total} under the same assumptions on the Newmark parameter and the external force.

On the effect of system time-step and subcycling on accuracy

In absence of external forces, the exact solution satisfies $\mathcal{E}^{(n+1)} - \mathcal{E}^{(n)} = 0$. Therefore, the quantities $\mathcal{E}_{\text{algorithm}}^{(n\to n+1)}$ and $\mathcal{E}_{\text{interface}}^{(n\to n+1)}$ can serve as error / accuracy indicators of a multi-time-stepping scheme. Note that these quantities arise, respectively, due to time-stepping scheme, and due to decomposing domain into subdomains. Of course, both these quantities are affected by subcycling.

From equation (2.83), it is easy to check that $\mathcal{E}_{\text{algorithm}}^{(n\to n+1)}$ is proportional to Δt^2 and inversely proportional to η_i^2. Therefore, algorithmic error in the subdomains can always be decreased by employing either of these two strategies:

- decreasing the system time-step by keeping the subcycling ratios fixed (i.e., keeping η_i fixed)

- decreasing the subdomain time-step (i.e., increase the values of η_i) by keeping the system time-step fixed

Equation (2.84) can be written as follows:

$$\mathcal{E}_{\text{interface}}^{(n\to n+1)} = \Delta t \sum_{i=1}^{S} \left\{ \frac{1}{\eta_i} \sum_{j=0}^{\eta_i-1} \left((1-\gamma_i)\,\boldsymbol{\lambda}^{\left(n+\frac{j}{\eta_i}\right)} + \gamma_i \boldsymbol{\lambda}^{\left(n+\frac{j+1}{\eta_i}\right)} \right)^{\mathrm{T}} \boldsymbol{C}_i \boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)} \right\} + O\left(\frac{\Delta t^2}{\eta_i^2}\right), \tag{2.105}$$

where $\mathcal{E}_{\text{interface}}^{(n\to n+1)}$ is linearly proportional to Δt, which indicates that the error due to domain decomposition can *always* be decreased with lowering the system time-step. However, for a fixed system time-step, the quantity in the parenthesis can be of $O(1)$ in magnitude. Therefore, choosing smaller subdomain time-steps while keeping the system time-step fixed need not improve the accuracy. This quantity may even grow with increase in the subcycling ratios. Hence, an appropriate quantity that can indicate the improvement or worsening of accuracy by subcycling is $\mathcal{E}_{\text{interface}}^{(n\to n+1)}$, which can be calculated on the fly during a numerical simulation. Larger values of $\mathcal{E}_{\text{interface}}^{(n\to n+1)}$ in magnitude implies that subcycling is adversely affecting the accuracy.

Summarizing, the accuracy of the numerical results under the proposed multi-time-step method can *always* be improved by decreasing the system time-step. The overall accuracy need not always improve with subcycling for a fixed system time-step. These theoretical observations are numerically verified in Section 2.9.

2.8 On Performance of Backward Difference and Runge-Kutta Schemes

In the numerical analysis literature, backward difference formulae (BDF) and implicit Runge-Kutta (IRK) schemes have been the schemes of choice for solving DAEs [24, 47, 48]. The following quote by Petzold has been a popular catch-phrase for promoting BDF schemes: "*BDF is so beautiful that it is hard to imagine something else could be better*" [47, p. 481]. This statement may be true for first-order DAEs that arise from modeling of physical systems involving dissipation. But these two families of schemes may *not* be the best choices for second-order DAEs that posses important physical invariants (e.g., conservation of energy). In the context of second-order DAEs, the time-stepping schemes from the Newmark family can perform really well, especially, with respect to invariants like the total energy. Therefore, the Newmark family of time-stepping schemes can be considered as strong alternatives to BDFs and IRKs for second-order transient systems. The Newmark family of time stepping schemes (which have been popular in Civil Engineering for solving ODEs arising in structural dynamics and earthquake engineering) did not get as much attention as they deserve to solve DAEs in both numerical analysis and engineering communities. The algebraic constraints in a DAE introduce high frequency modes, and fully implicit schemes such as Newmark family of time stepping schemes are particularly suited to avoid instabilities due to high frequency modes without introducing excessive damping.

We now show that there will be excessive numerical damping if the proposed coupling method is based on BDF or IRK schemes instead of the Newmark family of time stepping schemes. It may be argued that numerical damping is good for numerical stability, but excessive damping fails to preserve the important invariants (e.g., conservation of energy). Newmark family of time stepping schemes provide much better results under the same system time-step, especially, in the prediction of important physical invariants.

The simplest scheme under both BDF and IRK families is the backward Euler scheme (which is also referred to as the implicit Euler scheme). We rewrite the governing equations as first-order DAEs of form

$$\boldsymbol{M}_i \dot{\boldsymbol{v}}_i(t) + \boldsymbol{K}_i \boldsymbol{d}_i(t) = \boldsymbol{f}_i(t) + \boldsymbol{C}_i^{\mathrm{T}} \boldsymbol{\lambda} \quad \forall i = 1, \cdots, S, \tag{2.106a}$$

$$\dot{\boldsymbol{d}}_i(t) = \boldsymbol{v}_i(t) \text{ and} \tag{2.106b}$$

$$\sum_{i=1}^{S} \boldsymbol{C}_i \boldsymbol{v}_i(t) = \boldsymbol{0}\cdot \tag{2.106c}$$

Under the backward Euler scheme, the velocities and accelerations are approximated as

$$\boldsymbol{v}_i^{(n+1)} = \frac{\boldsymbol{d}_i^{(n+1)} - \boldsymbol{d}_i^{(n)}}{\Delta t}, \text{ and } \quad \boldsymbol{a}_i^{(n+1)} = \frac{\boldsymbol{v}_i^{(n+1)} - \boldsymbol{v}_i^{(n)}}{\Delta t}\cdot \tag{2.107}$$

In the absence of subcycling, following a similar procedure presented in the previous sections, one can arrive at the following equation for the coupling method based on the backward Euler scheme

$$[\![\mathcal{E}^{(n)}]\!] = -\sum_{i=1}^{S} \left(\mathcal{T}_i \left([\![\boldsymbol{v}_i^{(n)}]\!] \right) + \mathcal{V}_i \left([\![\boldsymbol{d}_i^{(n)}]\!] \right) \right) \quad \forall n, \tag{2.108}$$

which is *strictly negative* for any non-trivial motion of the subdomains. Figure 2.8 nicely summarizes the above discussion using the split degree-of-freedom problem. The system shown in Figure 2.3 was solved with the parameters: $m_A = 0.1$, $m_B = 0.005$, $k_A = 2.5$ and $k_B = 50$. The initial values are set to be as follows: $v_0 = 1.0$ and $d_0 = 0.1$. External forces are set to be zero. The proposed coupling method presented in this chapter is employed to solve the coupled system using Newmark average acceleration and central difference methods, with no subcycling. In addition to the aforementioned excessive numerical dissipation, the following factors make BDF and IRK schemes not particularly suitable for developing a multi-time-step coupling:

(a) High-order BDF and IRK schemes are non-self-starting.

(b) BDF and IRK schemes are developed for first-order DAEs. To solve a second-order DAE (which is the case in this chapter), auxiliary variables need to be introduced, which will increase the number of unknowns and the computational cost.

(c) IRK schemes involve multiple stages, and are generally considered difficult to implement.

2.9 Representative Numerical Results

Using several canonical problems, we illustrate that the proposed multi-time-step coupling method possesses the following desirable properties:

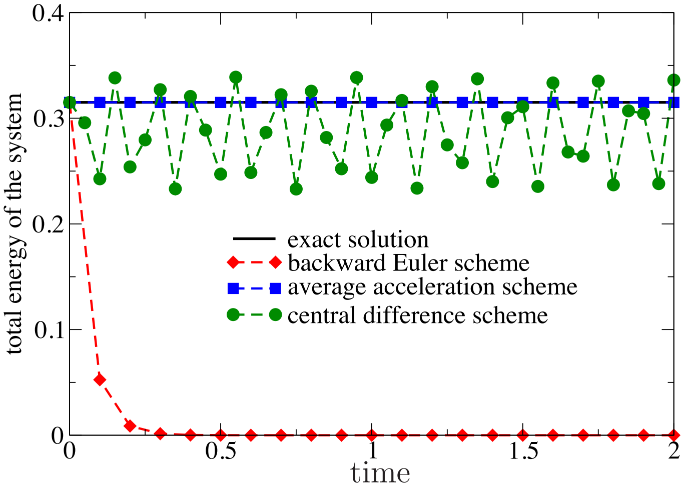


Figure 2.8: Coupling using the backward Euler scheme: The second-order differential-algebraic equations is converted to first-order differential-algebraic equations, and the resulting system is solved using the backward Euler scheme, which is the simplest member of backward difference formulae (BDF) and implicit Runge-Kutta (IRK) schemes.

(I) All subdomains can subcycle simultaneously. That is, $\Delta t_i < \Delta t \ \forall i = 1, \cdots, S$.

(II) The method can handle multiple subdomains.

(III) Drift in displacements along the subdomain interface is not significant.

(IV) Under fixed subdomain time-steps, the accuracy of numerical solutions can be improved by decreasing the system time-step.

(V) For a fixed system time-step, accuracy of the solutions *may* be improved using subcycling. We also show that monitoring $\mathcal{E}_{\mathrm{interface}}^{(n\to n+1)}$ at every system time-step can serve as a simple criterion to decide whether or not subcycling will improve the accuracy. This criterion can be calculated on the fly during a numerical simulation.

Split degree-of-freedom with three subdomains

An attractive feature of the proposed coupling method is that it can handle multiple subdomains, which is illustrated in this test problem. The single degree-of-freedom is split into three subdomains A, B and C, as shown in Figure 2.9. The problem parameters are taken as follows: $m_A = 5$, $m_B = 0.1$, $m_C = 0.01$, $k_A = 5$, $k_B = 2.5$ and $k_C = 4$. Subdomain time-steps are taken as $\Delta t_A = 0.01$, $\Delta t_B = 0.005$ and $\Delta t_C = 0.0025$. The system time-step is taken as $\Delta t = 0.01$. Newmark average acceleration scheme is employed in all the subdomains. The subdomain external forces are taken as $f_A = f_C = 0$ and $f_B = 1$. The system is subject to the initial conditions $d_0 = 1.0$ and $v_0 = 0.0$. Figure 2.10 compares analytical solution with the numerical results for the kinematic quantities. Figure 2.11 shows the Lagrange multipliers (i.e., interface forces) and the total energy of the system. The proposed coupling method performed well.

One-dimensional problem with homogeneous properties

Consider the vibration of a homogeneous one-dimensional elastic axial bar with the left end of the bar fixed and a constant tip load is applied at the right end of the

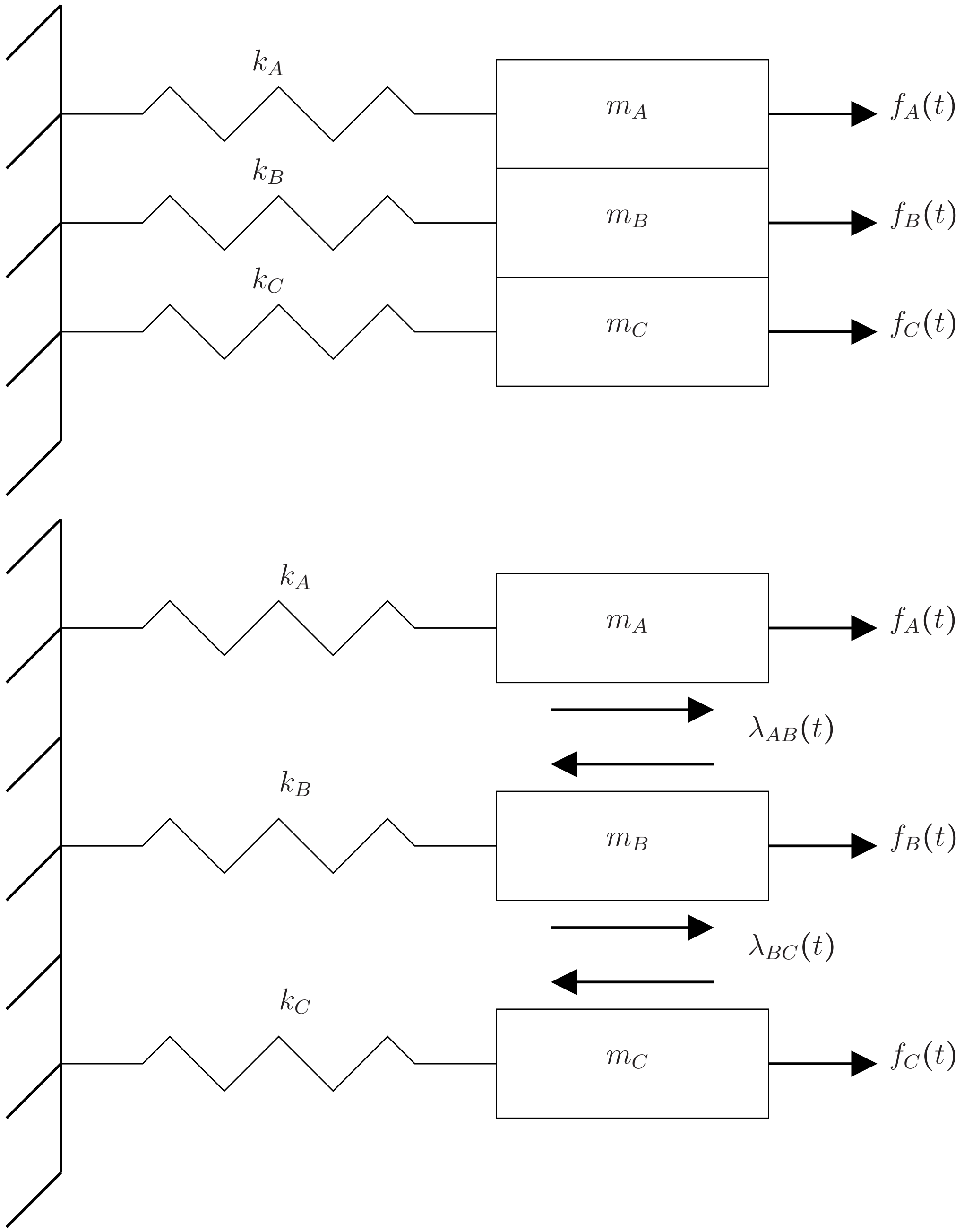


Figure 2.9: Split degree-of-freedom with three subdomains: The problem is solved using the proposed multi-time-step coupling method.

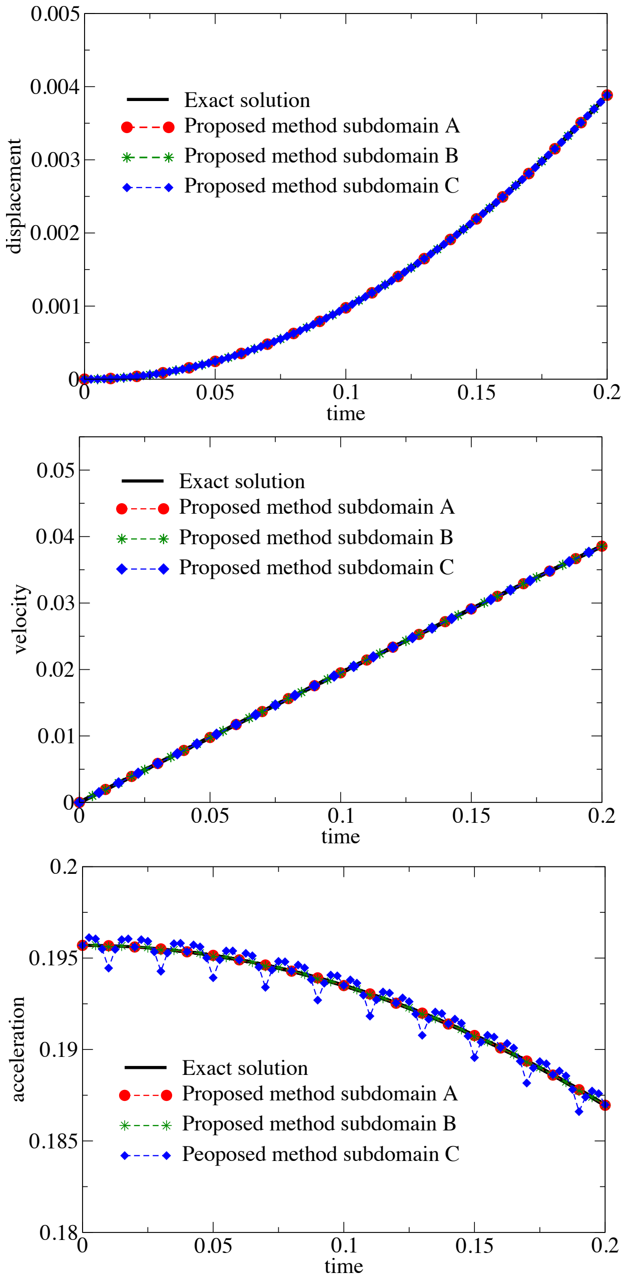


Figure 2.10: Split degree-of-freedom with three subdomains: Numerical and analytical results for displacement in problem 2 is shown in this figure. As seen here, the numerical results under the proposed coupling method matches well with the exact values.

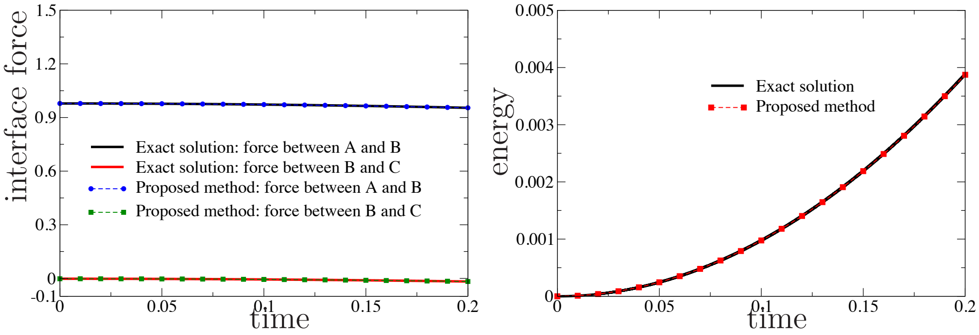


Figure 2.11: Split degree-of-freedom with three subdomains: The top figure shows the interface reaction forces λ_{AB} and λ_{BC} with respect to time. This bottom figure shows the total energy of the system with respect to time.

bar. The governing equations take the following form:

$$\rho A \frac{\partial^2 u}{\partial t^2} - \frac{\partial}{\partial x}\left(EA\frac{\partial u}{\partial x}\right) = P\delta(x = L)H(t = 0) \quad \forall x \in (0, L), \ \forall t \in (0, T], \tag{2.109a}$$

$$u(x = 0, t) = 0 \quad \forall t \in (0, T], \tag{2.109b}$$

$$E\frac{\partial u}{\partial x}(x = L, t) = 0 \quad \forall t \in (0, T], \tag{2.109c}$$

$$u(x, t = 0) = 0 \quad \forall x \in (0, L) \text{ and} \tag{2.109d}$$

$$\frac{\partial u}{\partial t}(x, t = 0) = 0 \quad \forall x \in (0, L), \tag{2.109e}$$

where $\delta(\cdot)$ is the Dirac-delta distribution, $H(\cdot)$ is the Heaviside function, and P is a constant tip loading. The analytical solution for the displacement can be written as

$$u(x, t) = \frac{Px}{EA} + \frac{8PL}{\pi^2 EA} \sum_{n=1,3,\cdots} (-1)^{\frac{n+1}{2}} \frac{1}{n^2} \sin(\beta_n x)\cos(\omega_n t), \tag{2.110}$$

where

$$\beta_n = \frac{n\pi}{2L}, \quad \omega_n = \beta_n \sqrt{\frac{E}{\rho}} = \frac{n\pi}{2L}\sqrt{\frac{E}{\rho}}\cdot \tag{2.111}$$

This test problem is the same as the one considered in Reference [30] but with different parameters. Herein, we shall use this test problem to illustrate that the proposed coupling method can handle multiple subdomains simultaneously, which is not the case with the PH method as presented in [30].

The computational domain is divided into three subdomains of equal lengths, as shown in Figure 2.12. Each subdomain is uniformly meshed using five two-node line elements. The Young's modulus is taken as $E = 10^4$, the density $\rho = 0.1$, the area of cross section $A = 1$, the total length of the bar $L = 1$, and the tip loading is taken as $P = 10$. Newmark average acceleration scheme is employed in subdomains A and C ($\beta_A = \beta_C = 1/4$ and $\gamma_A = \gamma_C = 1/2$), and the central difference scheme is employed in subdomain B ($\beta_B = 0$ and $\gamma_B = 1/2$). The critical time-step is $1.217{\times}10^{-4}$. The system time-step is taken as $\Delta t = 10^{-3}$. The subdomain time-steps for A and C are taken as $\Delta t_A = \Delta t_C = 10^{-3}$. The problem is solved using three different subdomain time-steps for B, which are defined through $\eta_B = \Delta t/\Delta t_B = 10,\ 100,\ 1000$. Figure 2.13 shows the tip displacement and the total energy obtained using the proposed coupling method. Figures 2.14 and 2.15, respectively, show drift in displacements and the interface Lagrange multipliers. *These figure clearly illustrate that, under a fixed system time-step, the accuracy can be improved by employing subcycling in the subdomains under the proposed coupling method. This implies that the time-step required for the explicit scheme need not limit the time-step in the entire computational domain under the proposed multi-time-step coupling method.*

The problem is solved again with subdomain B divided into 10 two-node linear elements. In this case the critical time-step is approximately 6.085×10^{-5}. We took the subdomain time-steps to be fixed at 10^{-5} and altered the system time-step to illustrate the effect of subcycling. The results are presented in figures 2.16, 2.17

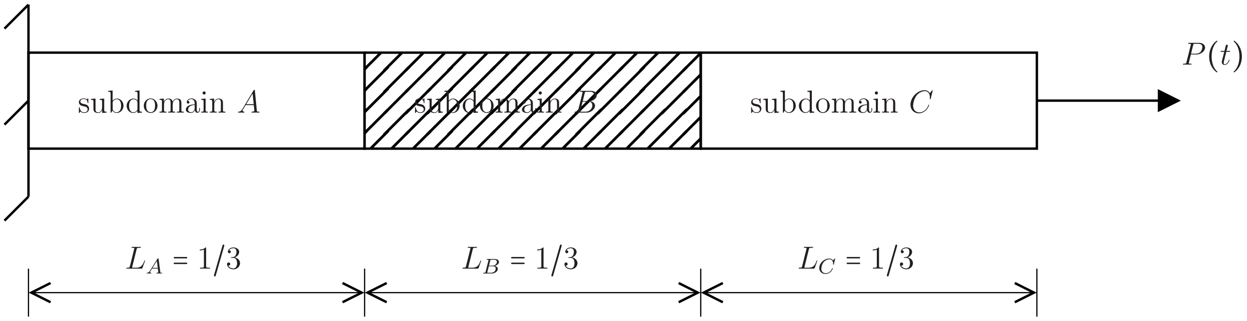


Figure 2.12: One-dimensional problem with homogeneous properties: Consider an axial elastic bar of unit length. The left end of the bar is fixed, and a constant load of $P(t) = 10$ is applied to the right end of the bar.

and 2.18. *These figures illustrate that, under the proposed multi-time-step coupling method with fixed subdomain time-steps (i.e., fixed Δt_i), the accuracy can be improved by employing smaller system time-steps.*

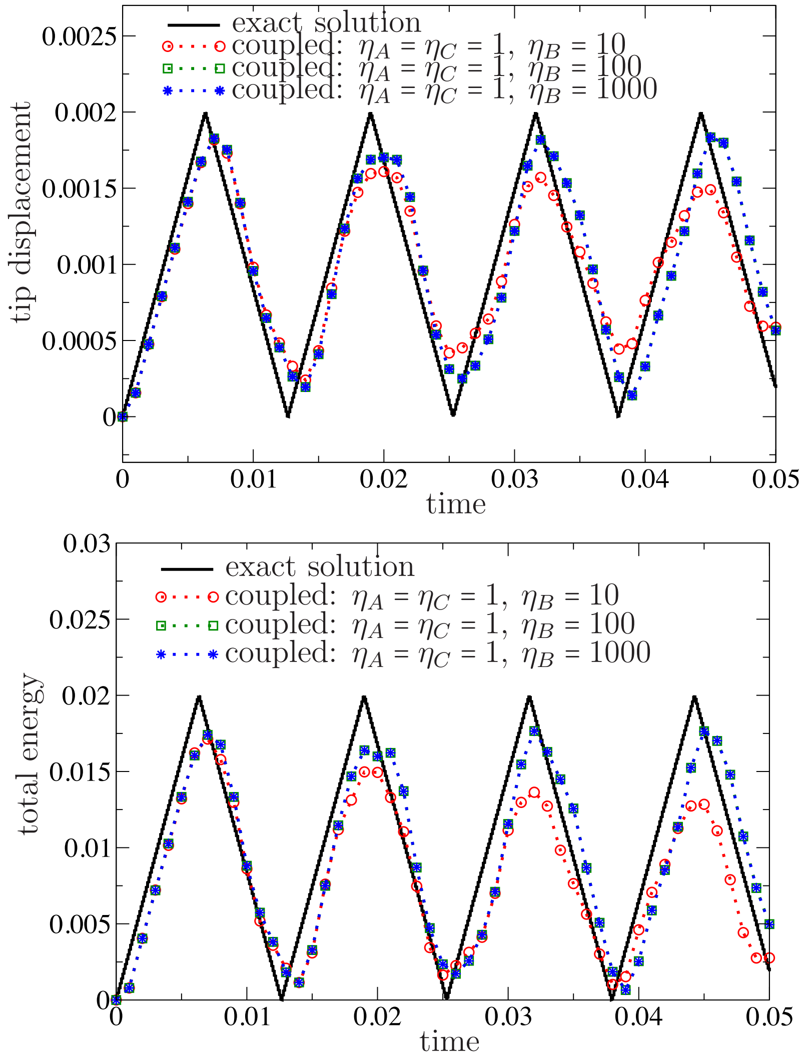


Figure 2.13: One-dimensional problem with homogeneous properties: The top and bottom figures, respectively, show the tip displacement and total energy as a function of time.

Square plate subjected to a corner force

A bi-unit square of homogeneous elastic material is fixed at the left end and a constant force with components $f_x = f_y = 1$ is applied at the right bottom corner. The Lamé parameters are taken as $\lambda = 100$ and $\mu = 100$, and the mass density is taken as $\rho = 100$. The computational domain is decomposed into four equally sized

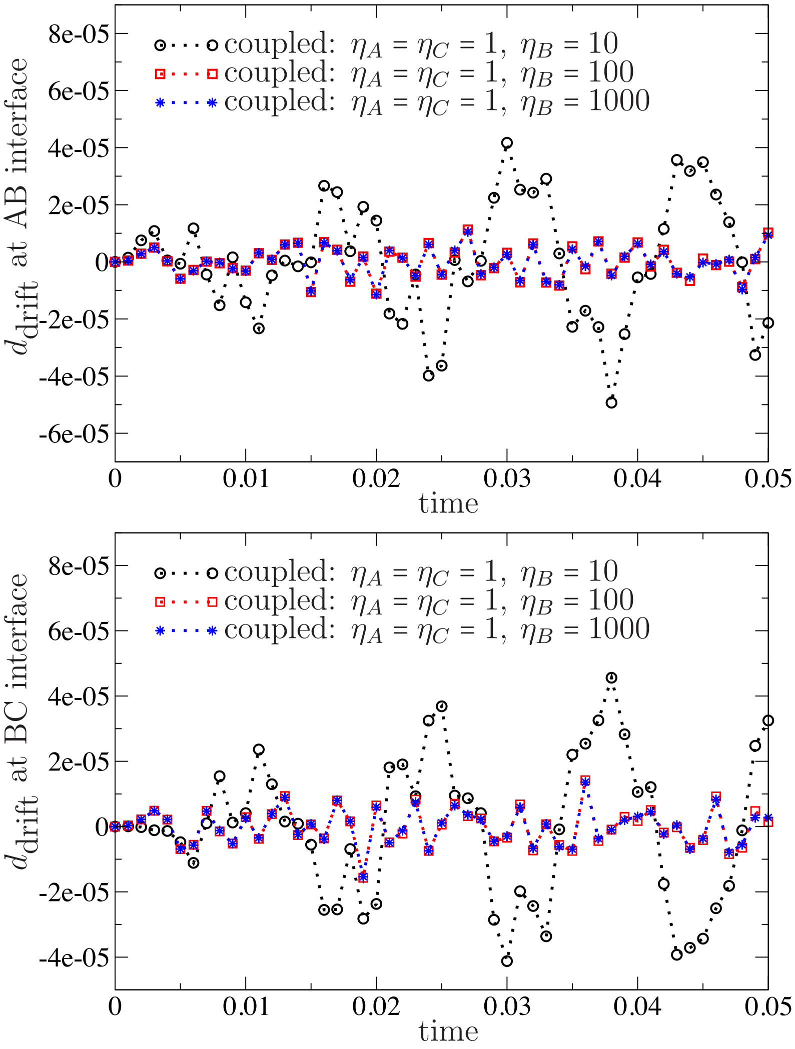


Figure 2.14: One-dimensional problem with homogeneous properties: The top figure shows the drift in the displacement at the interface of subdomains A and B. The bottom figure shows the drift in displacement at the interface of subdomains B and C.

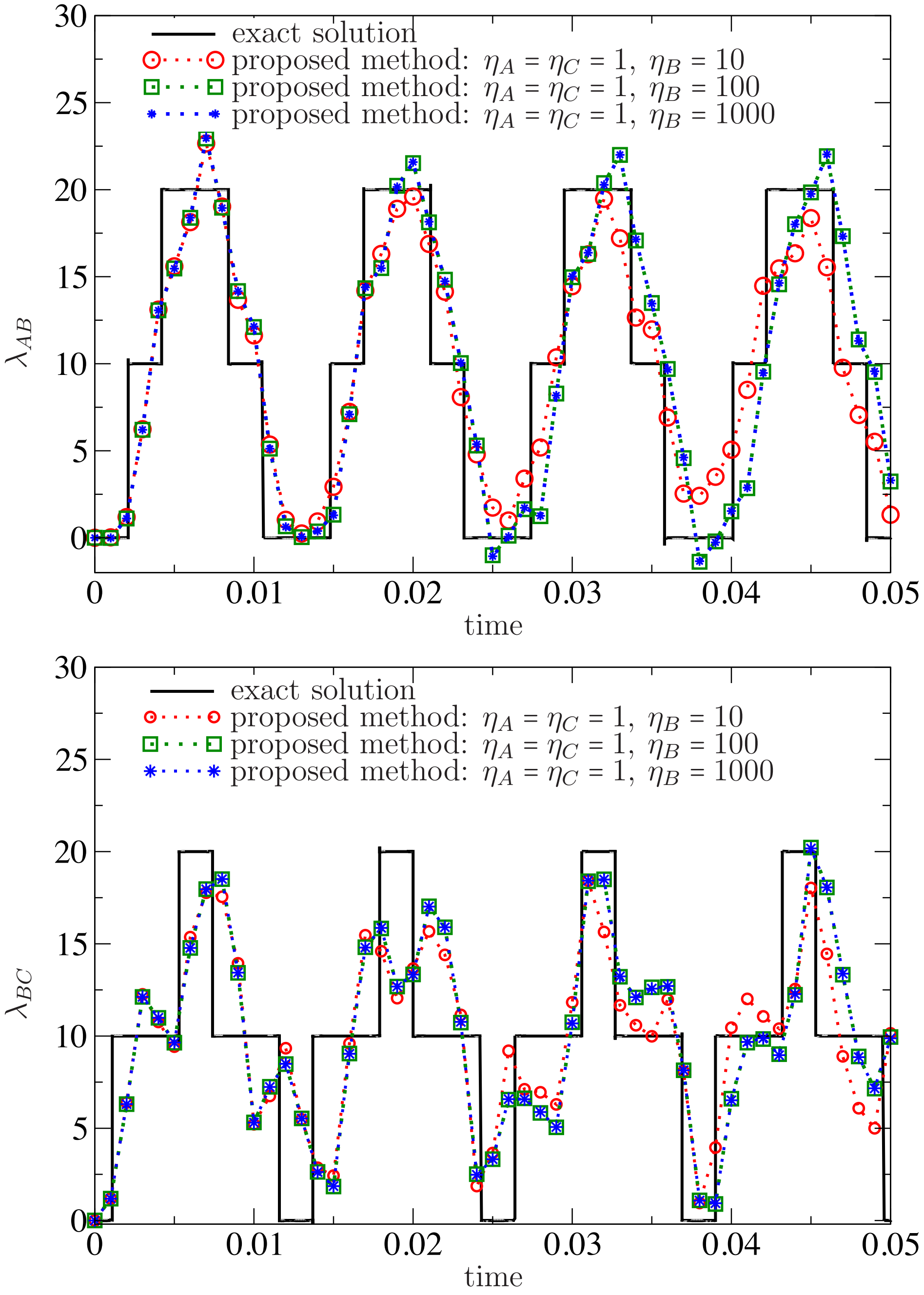


Figure 2.15: One-dimensional problem with homogeneous properties: The top figure shows the interface force between subdomains A and B. The bottom figure shows the interface force between subdomains B and C.

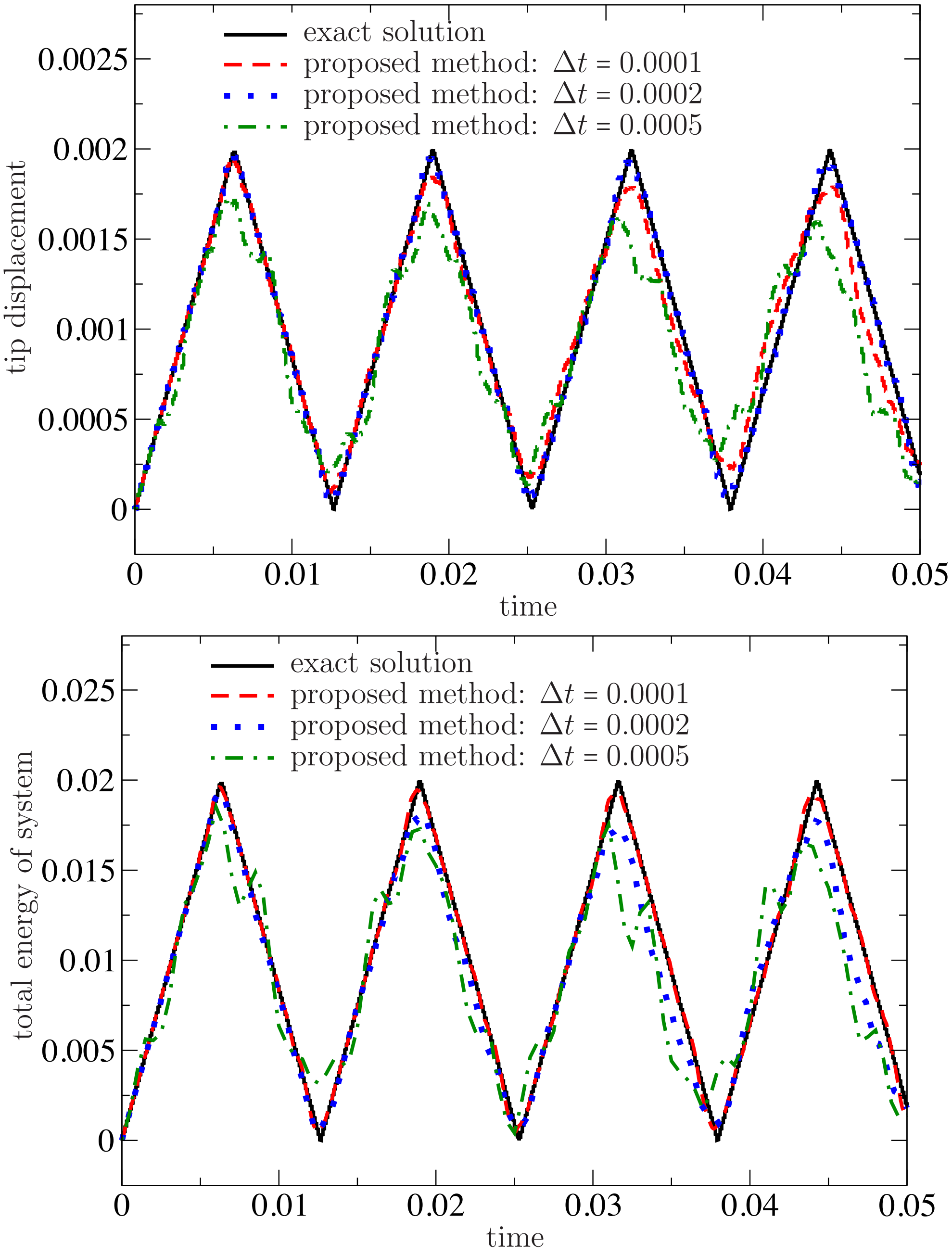


Figure 2.16: One-dimensional problem with homogeneous properties: The top and bottom figures, respectively show the tip displacement and the total energy as a function of time.

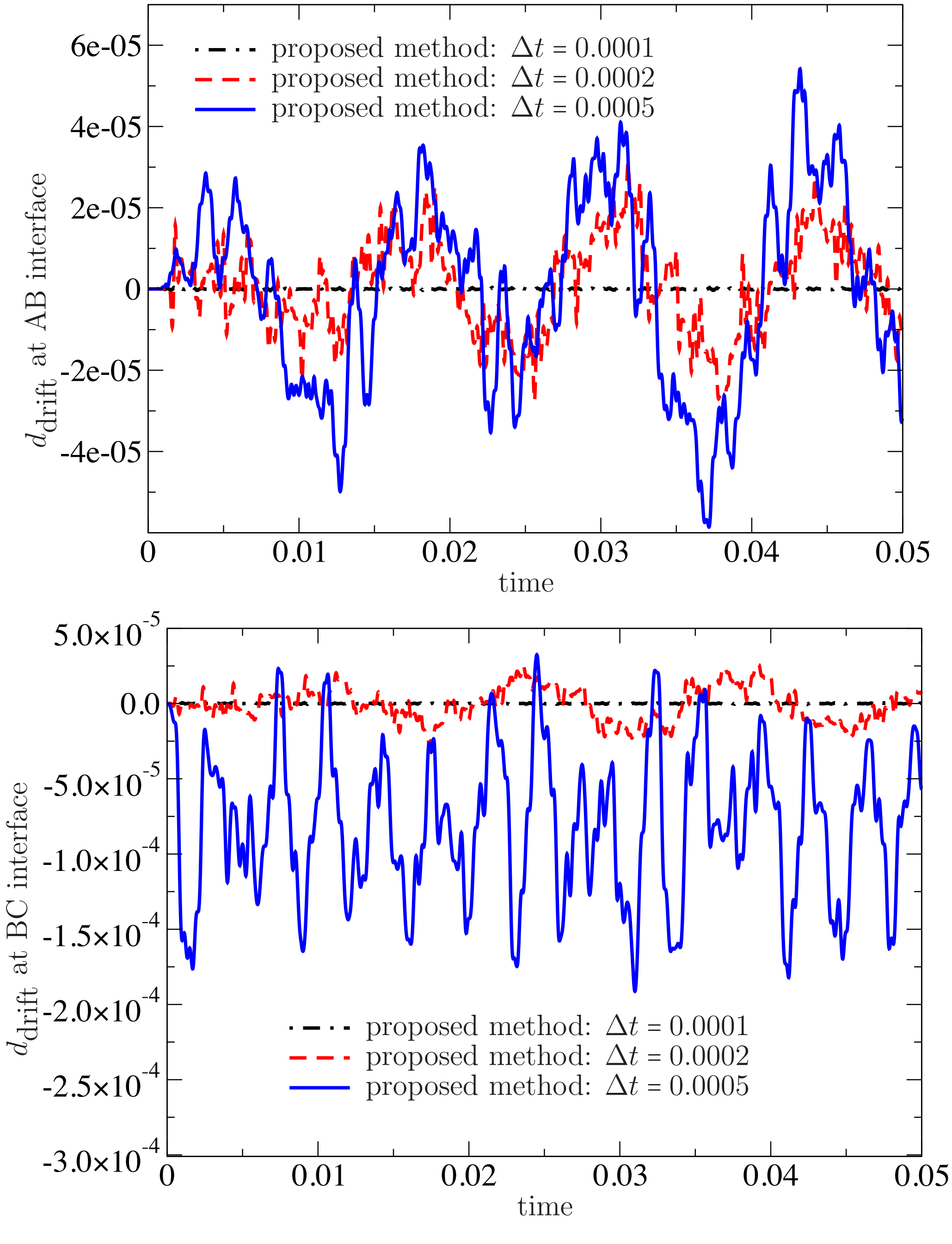


Figure 2.17: One-dimensional problem with homogeneous properties: The top figure shows the drift in the displacement at the interface of subdomains A and B. The bottom figure shows the drift in displacement at the interface of subdomains B and C.

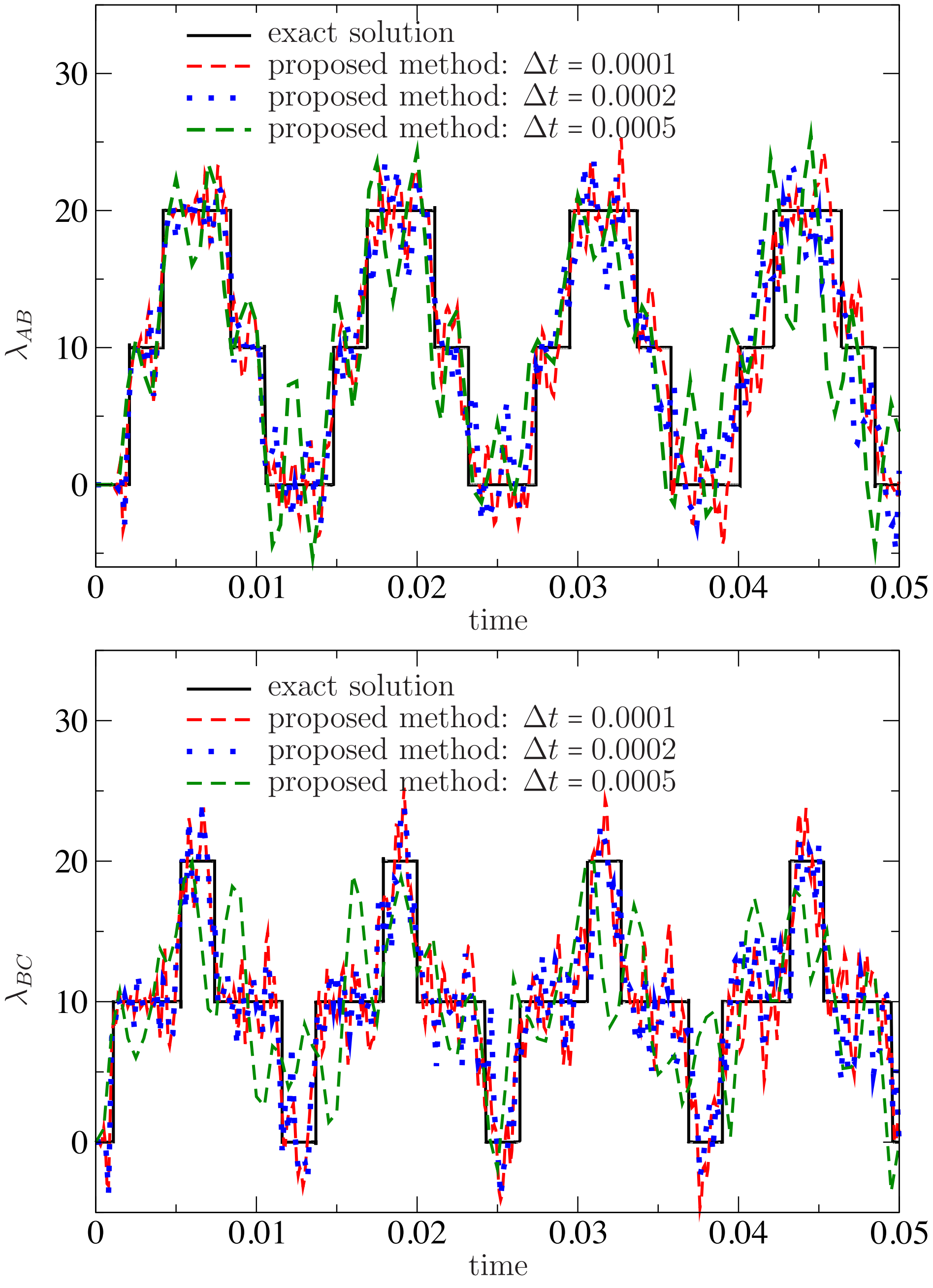


Figure 2.18: One-dimensional problem with homogeneous properties: The top figure shows the interface force between subdomains A and B. The bottom figure shows the interface force between subdomains B and C.

square subdomains. Four node quadrilateral elements are used to form a 5-element by 5-element mesh for each subdomain. Figure 2.19 provides a pictorial description of the problem. A similar problem is also considered in Reference [28], which also addressed multi-time-step coupling method for structural dynamics.

The central difference scheme $(\beta = 0, \gamma = 1/2)$ is employed for subdomains 1, 2 and 3, and Newmark average acceleration scheme $(\beta = 1/4, \gamma = 1/2)$ is employed for subdomain 4. Figure 2.20 illustrates that the accuracy can be improved by decreasing the system time-step. Figure 2.21 illustrates that the accuracy need not always improve by decreasing subdomain time-steps for a fixed system time-step. This is completely in accordance with the theoretical predictions. The numerical results shown in Figure 2.21 also illustrate that the proposed coupling method allows subcycling in all the subdomains. This is evident from the fact that all the chosen values for η_i $(i = 1, \cdots, S)$ are greater than unity. As it can be seen in Figure 2.22, subcycling can result in increase in drift. Figure 2.23 shows that there is no *appreciable* drift in displacements along the subdomain interface, and there is no drift in the velocities along the subdomain interface, as the proposed method imposes constraints on the continuity of velocities at every system time-step. Figure 2.24 shows that the theoretical bounds on the drifts in equations (2.75a)–(2.75b) match well with the numerical results. In all the numerical results, the proposed multi-time-step coupling method performed well, and behaved in accordance with the theoretical predictions derived in this chapter.

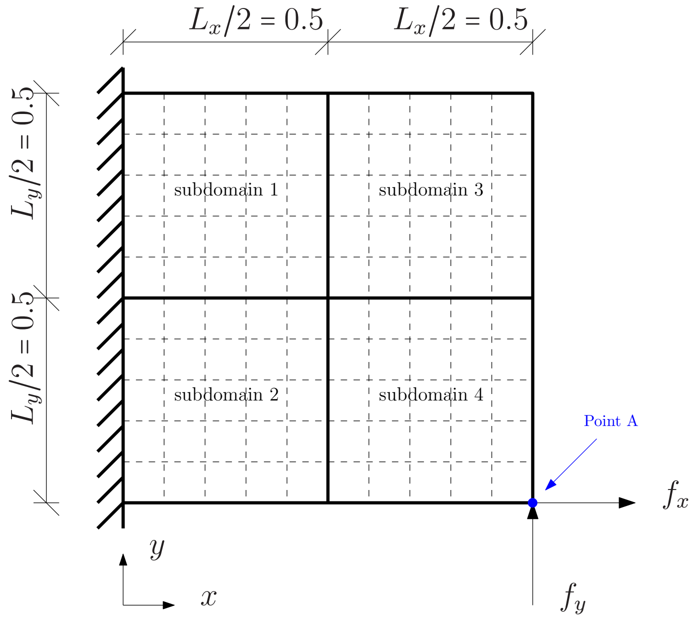


Figure 2.19: Square plate subjected to a corner force: This figure provides a pictorial description of the test problem. A bi-unit square of homogeneous elastic material is fixed at the left side, a constant force with components $f_x = f_y = 1$ is applied at Point A.

Two-dimensional wave propagation problem

Consider the transverse motion of a plate subject governed by the equations

$$\frac{1}{c_0^2}\frac{\partial^2 u}{\partial t^2} - \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = f(\mathbf{x}, t) \quad \forall\,(\mathbf{x}, t) \in \Omega \times \mathcal{I}, \tag{2.112a}$$

$$u(\mathbf{x}, t) = u^{\mathrm{p}}(\mathbf{x}, t) \quad (\mathbf{x}, t) \in \Gamma^{\mathrm{D}} \times \mathcal{I}, \tag{2.112b}$$

$$\mathrm{grad}[u] \cdot \widehat{\mathbf{n}}(\mathbf{x}) = s^{\mathrm{p}}(\mathbf{x}, t) \quad (\mathbf{x}, t) \in \Gamma^{\mathrm{N}} \times \mathcal{I}, \tag{2.112c}$$

$$u(\mathbf{x}, t = 0) = u_0(\mathbf{x}) \quad \mathbf{x} \in \Omega, \text{ and} \tag{2.112d}$$

$$\dot{u}(\mathbf{x}, t = 0) = v_0(\mathbf{x}) \quad \mathbf{x} \in \Omega, \tag{2.112e}$$

where $u(\mathbf{x}, t)$ is the transverse displacement, c_0 is the wave velocity, $f(\mathbf{x}, t)$ is the forcing function, $\widehat{\mathbf{n}}(\mathbf{x})$ is the unit outward normal to the boundary, $u^{\mathrm{p}}(\mathbf{x}, t)$ is the prescribed displacement on the boundary, $s^{\mathrm{p}}(\mathbf{x}, t)$ is the prescribed transverse traction,

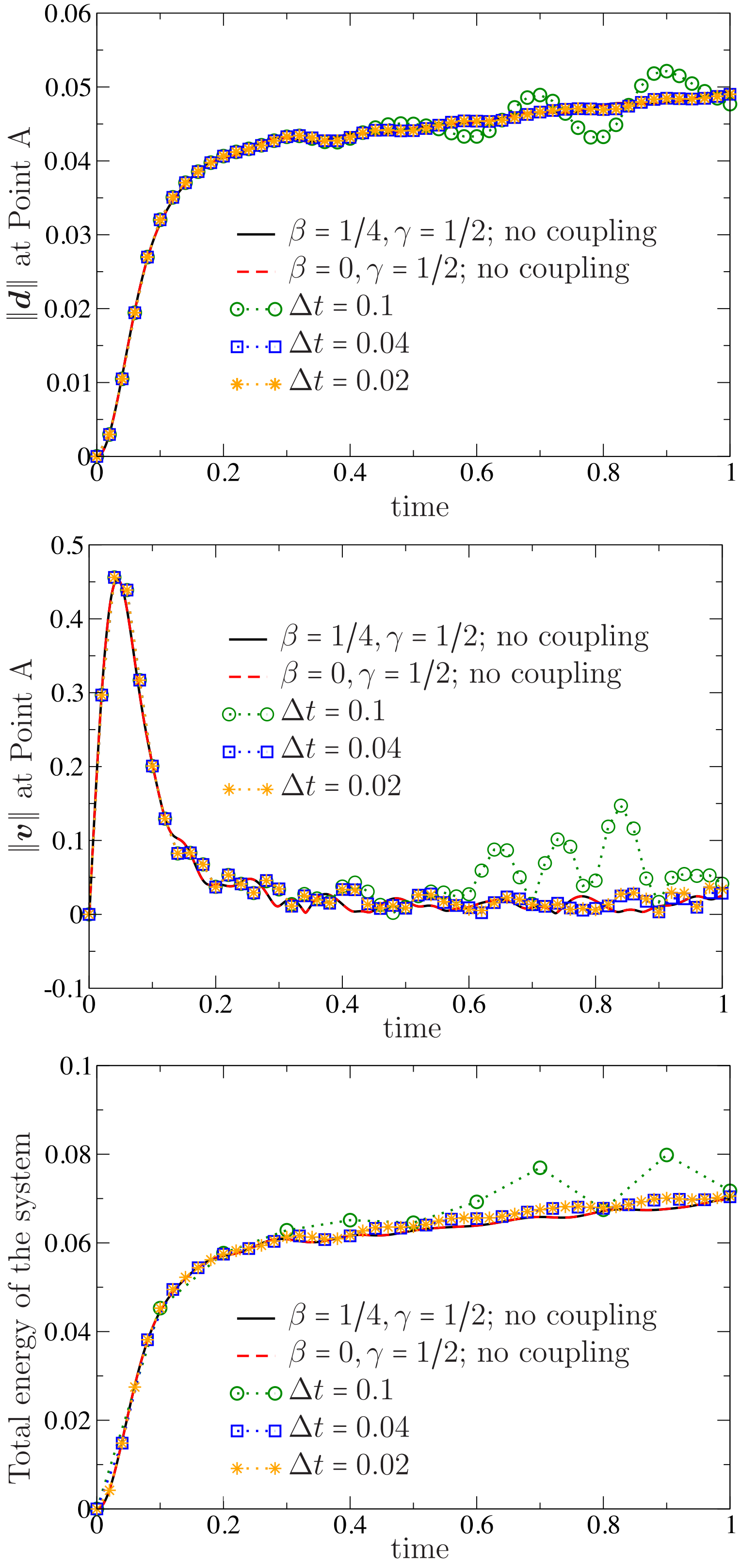


Figure 2.20: Square plate subjected to a corner force: The subdomain time-step in all subdomains is taken as 0.02.

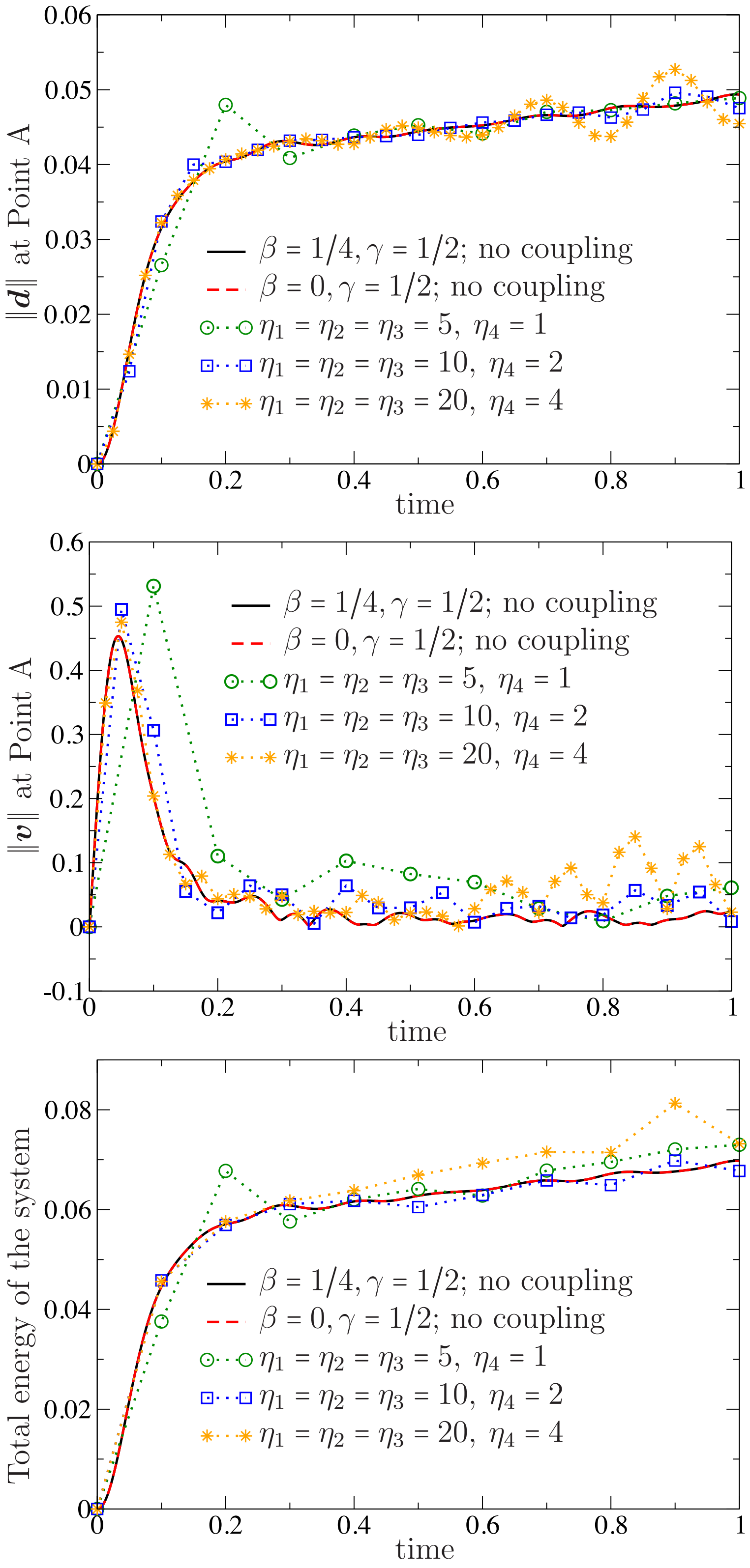


Figure 2.21: Square plate subjected to a corner force: The system time-step is taken as 0.1. The bottom figure shows the total energy of the system for values of η_i.

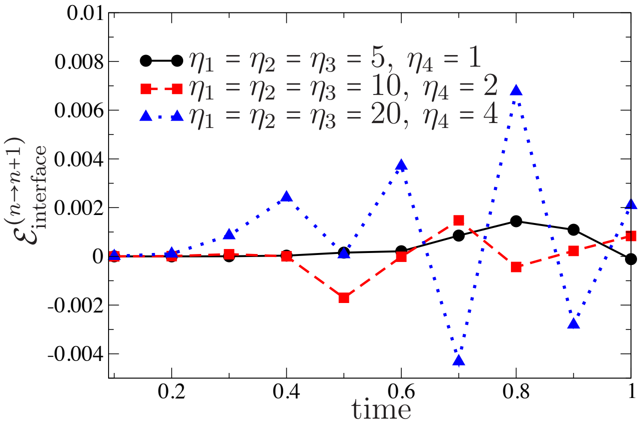


Figure 2.22: Square plate subjected to a corner force: The system time-step is taken as 0.1. The value of $\mathcal{E}_{\text{interface}}^{(n\rightarrow n+1)}$ for the problem presented in 2.21 is plotted. As seen above, sub-cycling can increase the discretization error at the interface.

and $u_0(\mathbf{x})$ and $v_0(\mathbf{x})$ are, respectively, the initial displacement and initial velocity. Computational domain is denoted by Ω. The part of the boundary on which Neumann boundary condition is denoted by Γ^{N}, and Γ^{D} is the part of the boundary on which Dirichlet boundary condition is prescribed. As usual, we assume that $\Gamma^{\mathrm{D}} \cap \Gamma^{\mathrm{N}} = \emptyset$, and $\Gamma^{\mathrm{D}} \cup \Gamma^{\mathrm{N}} = \partial\Omega$. The time interval of interest is $\mathcal{I}$.

We consider the computational domain to be a rectangle with $L_x = 2$ and $L_y = 1$. The boundary is fixed on three sides, and is excited by a sinusoidal force of the following form on the other side as

$$f(\mathbf{x},t) = \begin{cases} f_0 \sin\left(\frac{2\pi}{\tau_{\text{load}}} t\right) & t \in [0, \tau_{\text{load}}] \\ 0 & t > \tau_{\text{load}} \end{cases} \quad \text{and} \quad \mathbf{x} \in \{0\} \times [2L_y/5, 3L_y/5]\cdot \tag{2.113}$$

A pictorial description of the problem is shown in Figure 2.25. The domain is decomposed into two subdomain, as shown in Figure 2.26. In this numerical example, we have taken $u_0 = 0$, $v_0 = 0$, $c_0 = 1$, $f_0 = 5.0$, and $\tau_{\text{load}} = 0.1$. Figure 2.27 shows the result for explicit/implicit integration using the proposed coupling method. In this

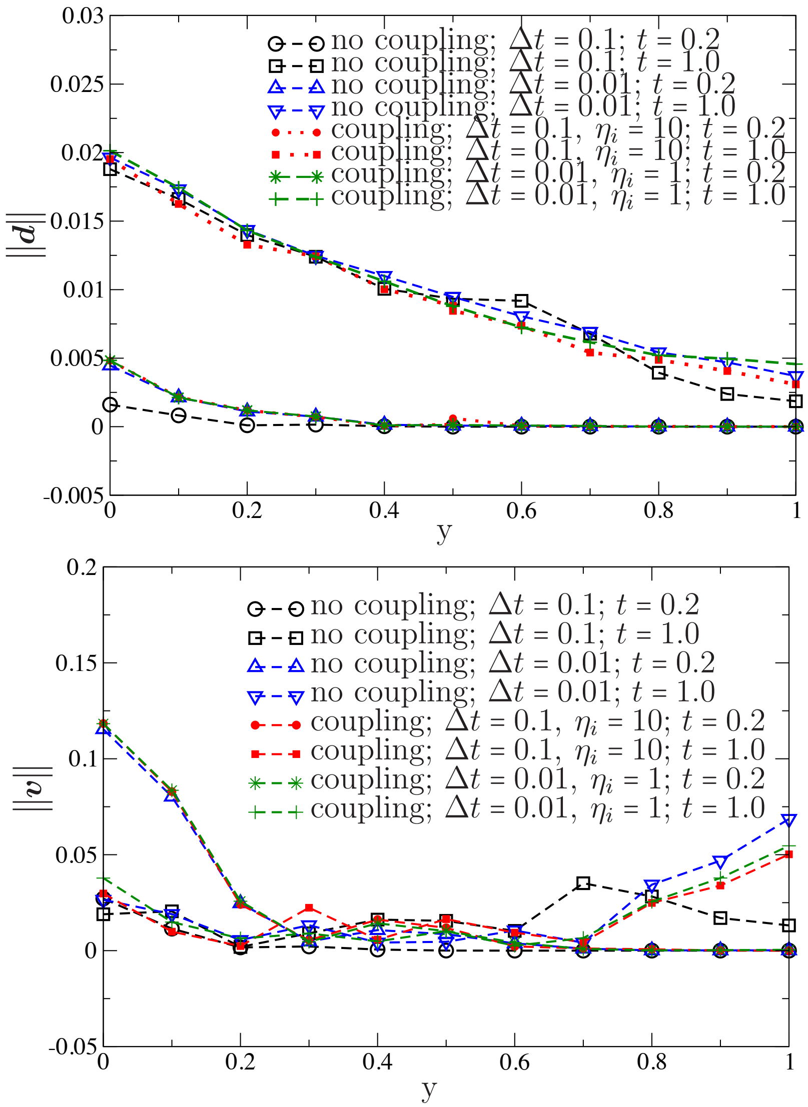


Figure 2.23: Square plate subjected to a corner force: The figure compares the numerical solutions under the proposed coupling method with that obtained without decomposing into subdomains.

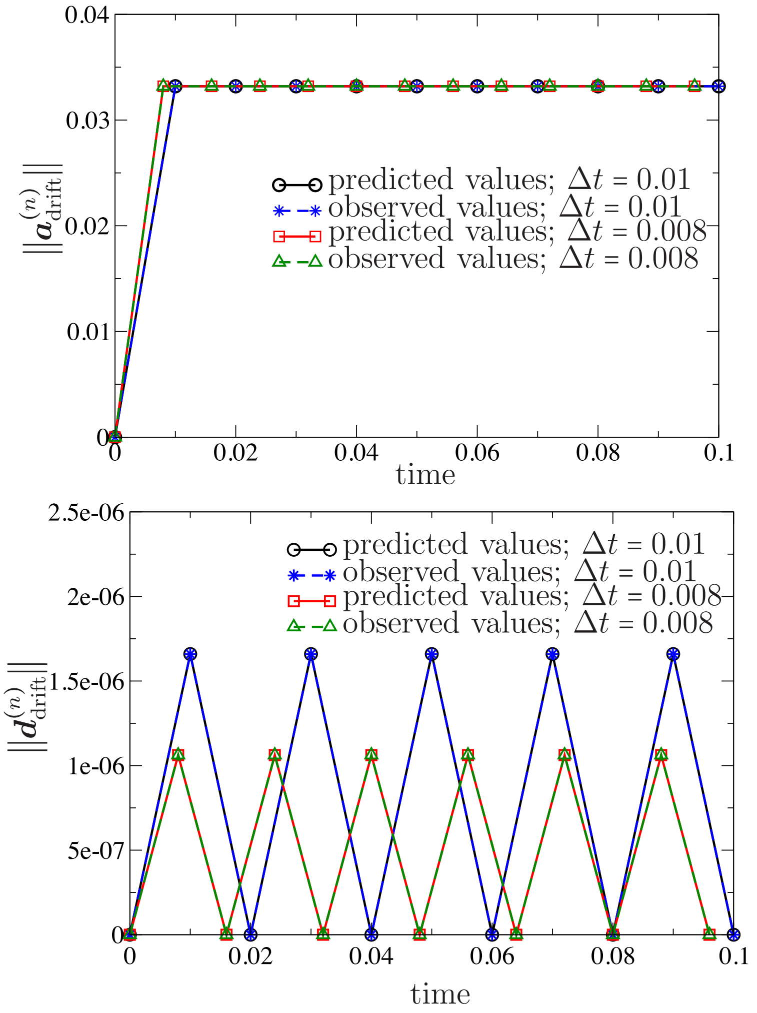


Figure 2.24: Bounds on drifts: The results in this figure substantiate the discussion presented in subsection 2.5. The L_2-norm of the drift in acceleration and and the drift in displacement at the subdomain interface are shown.

case, $(\gamma_1, \beta_1) = (1/2, 0)$, and $(\gamma_2, \beta_2) = (1/2, 1/4)$. The system time-step is $\Delta t = 10^{-4}$, subdomain time-steps are $\Delta t_1 = 10^{-5}$ and $\Delta t_2 = 10^{-4}$. As one can see from this figure, the proposed coupling method performed well. In particular, there are no spurious reflections at the subdomain interface, and there is no noticeable drift in the transverse displacement along the subdomain interface.

This problem also clearly demonstrates that the proposed multi-time-step coupling method can be attractive for wave propagation problems. The coupling method is more cost effective than mere employing either an explicit scheme or an implicit scheme in the entire domain. In wave propagation problems involving fast dynamics, small time-steps are needed, and hence explicit schemes are typically employed. This will result in taking large number of time-steps to be able to carry out the numerical simulation to a desired final time. On the other hand, under the proposed coupling method, one can use explicit methods in the regions with fast dynamics (which typically occur near the loading), and use an implicit time-stepping scheme with a larger subdomain time-step in the other regions. For the chosen problem, if one has to employ an explicit scheme in the entire domain, the time-step should be smaller than the critical time-step of 1.36×10^{-5}. Under the proposed multi-time-step coupling method, the user can employ an explicit scheme with time-steps smaller than the critical time-step near the load, and an unconditionally stable, implicit time-stepping scheme with larger time-steps in the rest of the computational domain.

2.10 Concluding Remarks

We have developed a multi-time-step coupling method that can handle *multiple subdomains* with different time-steps in different subdomains. The coupling method can couple implicit and explicit time-stepping schemes under the Newmark family even with disparate time-steps of more than two orders of magnitude in different subdomains. A systematic study on the energy preservation and energy properties of the proposed coupling method is presented, and the corresponding sufficient con-

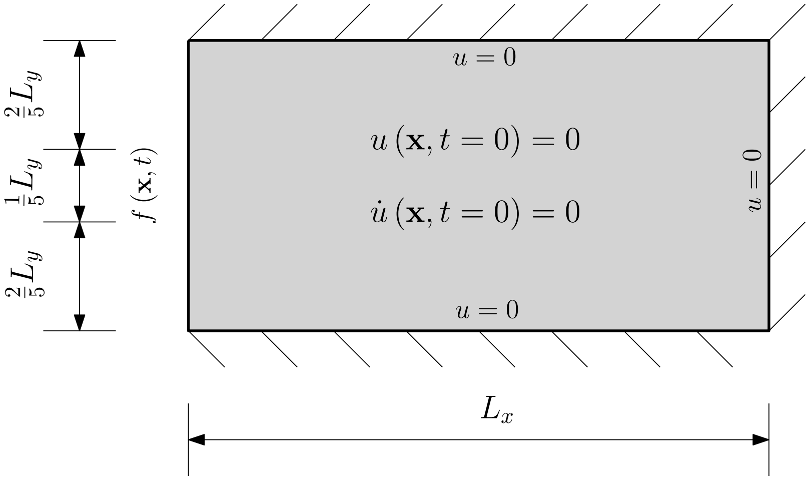


Figure 2.25: Two-dimensional wave propagation problem: A pictorial description. The elastic body is assumed to be isotropic and homogeneous. The force is applied over a length of $1/5L_y$ in the middle of the left side of the boundary.

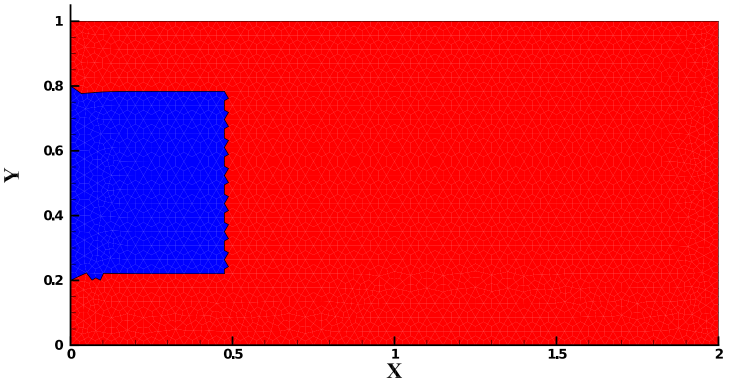
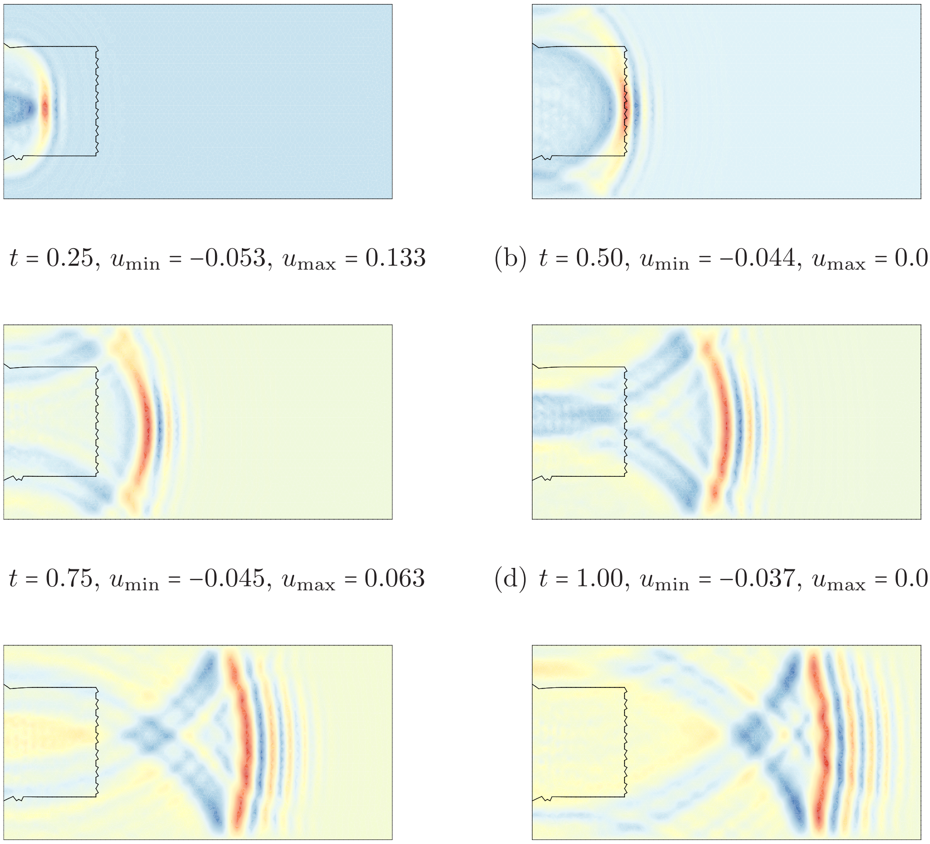


Figure 2.26: Two-dimensional wave propagation problem: The computational domain is divided into two subdomains. Subdomain 1 is shown in blue color, and subdomain 2 is shown in red color. The mesh consists of 5604 four-node quadrilateral elements.



(a) $t = 0.25$, $u_{\min} = -0.053$, $u_{\max} = 0.133$

(b) $t = 0.50$, $u_{\min} = -0.044$, $u_{\max} = 0.088$

(c) $t = 0.75$, $u_{\min} = -0.045$, $u_{\max} = 0.063$

(d) $t = 1.00$, $u_{\min} = -0.037$, $u_{\max} = 0.053$

(e) $t = 1.25$, $u_{\min} = -0.034$, $u_{\max} = 0.044$

(f) $t = 1.50$, $u_{\min} = -0.030$, $u_{\max} = 0.039$

Figure 2.27: Two-dimensional wave propagation problem: Here, $f_0 = 5.0$, $\tau_{\text{load}} = 0.1$, and $c_0 = 1$. The system time-step is $\Delta t = 10^{-4}$, and the subdomain time-steps are $\Delta t_1 = 10^{-5}$, and $\Delta t_2 = 10^{-4}$. The subdomain Newmark parameters are $(\gamma_1, \beta_1) = (1/2, 0)$, and $(\gamma_2, \beta_2) = (1/2, 1/4)$.

ditions are also derived. The proposed coupling method, in general, is not energy preserving. Despite claims in the literature, the quest for energy preserving multi-time-step coupling method is still on. One of the main conclusions of this chapter is about the effect of system time-step and subcycling on the accuracy. It has been shown that accuracy can always be improved by decreasing system time-step. It is widely believed that lowering subdomain time-step keeping the system time-step will also improve the accuracy under a multi-time-stepping scheme. Using careful mathematical analysis and numerical results, we have shown that this popular belief is not always the case. To this end, a simple criterion is also proposed, which can predict whether subcycling will improve accuracy. The criterion is to monitor $\mathcal{E}_{\mathrm{interface}}^{(n\rightarrow n+1)}$ at every system time-step, which can be calculated on the fly during a numerical simulation. Subcycling is desirable if this quantity is small.

The proposed multi-time-step coupling (which is a dual Schur domain decomposition technique) is well-suited for parallel computing. Specifically, one can utilize the advances made on the FETI method, which has shown to be scalable in a parallel setting for dual Schur domain decomposition methods [49]. There are several ways one could make advancements to the research presented in this chapter. *On the theoretical front*, a plausible future work is to perform a mathematical analysis on the numerical characteristics of the proposed multi-time-step coupling method on the lines of local error, propagation of error, and influence of perturbations. *On the computational implementation front*, one could implement the proposed coupling method in a parallel setting and do a systematic study on its parallel performance. *On the algorithmic front*, the next logical step is to extend the proposed multi-time-step coupling method to first-order transient systems, and eventually to fluid-structure interaction problems.

Chapter 3

COUPLING METHODS FOR ADVECTION-DIFFUSION SYSTEMS

3.1 Introduction and Motivation

Advection-diffusion-reaction equations can exhibit several mathematical (i.e., temporal and spatial) scales depending on the relative strengths of advection, diffusion and reaction processes, and on the strength of the volumetric source/sink. The presence of these mathematical scales is evident from the qualitative richness that the solutions of advection-diffusion-reaction equations exhibit. For example, it is well-known that solutions to advection-dominated problems typically exhibit steep gradients near the boundaries [50]. Solutions to diffusion-dominated problems tend to be diffusive and smooth [51], whereas reaction-dominated solutions typically exhibit sharp fronts and complex spatial patterns [52]. These scales can be systematically characterized using the well-known non-dimensional numbers – the Péclet number and the Damköhler numbers [53]. It needs to be emphasized that these equations, in general, are *not* amenable to analytical solutions. Therefore, one has to rely on predictive numerical simulations for solving problems of any practical relevance. Due to the presence of disparate mathematical scales in these systems, it is highly desirable to have a stable computational framework that facilitates tailored numerical formulations in different regions of the computational domain.

Several advances have been made in developing numerical formulations for advection-diffusion-reaction equations, especially in the area of stabilized formulations [54,55], and in the area of discrete maximum principles [56]. However, the main research challenge that still remains is to develop numerical methodologies for these type of problems to adequately resolve different mathematical scales in time and in space. *This chapter precisely aims at addressing this issue by developing a stable multi-time-step coupling framework for first-order transient systems that allows different time-steps, different time integrators and different numerical formulations in different regions of a computational domain.*

Most of the prior works on multi-time-step coupling methods have focused on the second-order transient systems arising in the area of structural dynamics (e.g., see the discussion in [57], and references therein). Some attempts regarding time integration of partitioned first-order systems can be found in [9,23]. In [9], a *staggered* multi-time-step coupling method is proposed. This method is considered as a staggered scheme as the Lagrange multipliers are calculated in an explicit fashion (i.e., based on the quantities known at prior time-levels). The stability and accuracy (especially, the control of drift along the subdomain interface) have been improved through the use of projection methods at appropriate time-levels. Since the method is a staggered scheme the obvious drawback is that the overall accuracy is first-order. However, it needs to be emphasized that the method proposed in [9] has better accuracy and stability properties than the previously proposed staggered schemes (e.g., [58,59]). In [23], several monolithic schemes are discussed for first-order transient systems but the treatment is restricted to transient diffusion equations (i.e., self-adjoint spatial operators) and multi-time-stepping was not addressed. Motivated by the work of Akkasale [21]; in which it has been systematically shown that many popular staggered schemes (e.g., [58, 59]) suffer from numerical instabilities for both first- and second-order transient systems; we herein choose a monolithic approach to develop coupling methods that allow multi-time-steps.

Recently, a multi-time-step monolithic coupling method for linear elastodynamics, which is a second-order transient system, has been proposed in [57]. However, developing a multi-time-step coupling method for first-order transient systems (e.g., unsteady advection-diffusion and advection-diffusion-reaction equations) will bring unique challenges. To name a few:

(i) As shown in [57], coupling explicit and implicit time-stepping schemes is *always* possible in the case of second-order transient systems. We will show later in this chapter that such coupling is *not* always possible for first-order transient systems, and can be achieved only if an appropriate stabilized form of the interface continuity constraint is employed. We will also show that this explicit/implicit coupling for first-order transient systems will come at an expense of controlled drift.

(ii) Spatial operators in advection-diffusion-reaction equations are not self-adjoint. Symmetry and positive definiteness of the discretized operators should be carefully examined to ensure the stability of multi-time-step coupling methods. For second-order transient systems, the overall stability of the coupling method can be achieved provided the stability criterion in each subdomain is satisfied (which depends on the choice of the time-stepping scheme in the subdomain and the choice of the subdomain time-step) [57]. We will show in a subsequent section that ensuring the stability of the time-stepping schemes in subdomains alone will not guarantee the overall stability of the coupling method. There is a need to place additional restrictions on the continuity constraints along the subdomain interface.

(iii) The governing equations of decomposed first-order transient systems form a system of differential/algebraic equations (DAEs) in Hessenberg form with a differential index 2. On the other hand, the governing equations for second-order transient systems form a system of DAEs with differential index 3. For more details on DAEs and associated terminology, see the brief discussion provided in subsection 3.3 or consult [47].

The current chapter builds upon the ideas presented in [23, 57]. *The central hypothesis on which the proposed multi-time-step coupling framework has been developed is two-fold*: (i) The governing equations before the domain decomposition form a system of ordinary differential equations (ODEs). On the other hand, the governing equations resulting from the decomposition of the domain form a system of differential/algebraic equations. It needs to be emphasized that many of the popular time-stepping schemes (which are developed for solving ODEs) are not appropriate for solving DAEs [60, 61]. At least, the accuracy and the stability properties will be altered considerably. The title of an influential chapter in the area of numerical solutions of DAEs by Petzold [22] clearly conveys the aforementioned sentiment:

"*Differential/algebraic equations are not ODEs.*" Therefore, we shall take a differential/algebraic equations perspective in posing the governing equations of the decomposed problems, and apply time-stepping strategies that are appropriate to solve DAEs. (ii) Development and performance of multi-time-step coupling methods for first-order transient systems is different from that of second-order transient systems.

The proposed monolithic multi-time-step coupling framework for first-order transient systems enjoys several attractive features, which will be illustrated in the subsequent sections by both theoretical analysis and numerical results. In the remainder of this chapter, we shall closely follow the notation introduced for multi-time-step coupling in [57].

3.2 Transient Advection-Diffusion-Reaction Equation

We shall consider transient advection-diffusion-reaction equation as the continuous model problem. Our choice provides an ideal setting for developing multi-time-step coupling methods for first-order transient systems, as the governing equations pose several unique challenges. *First*, the relative strengths of advection, diffusion, reaction, and volumetric source introduce multiple temporal scales, which compel a need for a multi-time-step computational framework. *Second*, the spatial operator is not self-adjoint, which adds to the complexity of obtaining stability proofs. It needs to be emphasized that the current efforts on multi-time-step coupling have focused on second-order transient systems, and the stability analyses have been restricted to the cases in which the coefficient (i.e., "stiffness") matrix is symmetric and positive definite. This will not be the case with respect to the advective-diffusive and advective-diffusive-reactive systems. *Third*, a numerical method to the chosen model problem can serve as a template for developing multi-time-step coupling methods for more complicated and important problems like transport-controlled bimolecular reactions, which exhibit complex spatial and temporal patterns. None of the prior works on multi-time-step coupling methods have undertaken such a comprehensive study, which this chapter strives to achieve.

Consider a chemical species that is transported by both advection and diffusion processes, and simultaneously undergoes a chemical reaction. Let $\Omega \subset \mathbb{R}^{nd}$ denote the spatial domain, where "*nd*" denotes the number of spatial dimensions. The boundary is denoted by $\partial\Omega$, which is assumed to be piece-wise smooth. The gradient and divergence operators with respect to $\mathbf{x} \in \Omega$ are, respectively, denoted by grad$[\cdot]$ and div$[\cdot]$. The time is denoted by $t \in \mathcal{I} := (0, T]$, where $\mathcal{I}$ is the time interval of interest. Let $c(\mathbf{x}, t)$ denote the concentration of the chemical species. As usual, the boundary is divided into two parts: Γ^{D} and Γ^{N} such that $\Gamma^{\mathrm{D}} \cup \Gamma^{\mathrm{N}} = \partial\Omega$ and $\Gamma^{\mathrm{D}} \cap \Gamma^{\mathrm{N}} = \varnothing$. Γ^{D} is the part of the boundary on which concentration is prescribed (i.e., Dirichlet boundary condition), and Γ^{N} is that part of the boundary on which flux is prescribed (i.e., Neumann boundary condition). We shall denote the advection velocity vector field by $\mathbf{v}(\mathbf{x}, t)$. The diffusivity tensor, which is a second-order tensor, is denoted by $\mathbf{D}(\mathbf{x})$, and is assumed to be symmetric and uniformly elliptic [45]. The initial boundary value problem for a transient advective-diffusive-reactive system can be written as follows:

$$\frac{\partial \mathrm{c}}{\partial t} + \mathrm{div}\left[\mathbf{v}\mathrm{c} - \mathbf{D}\left(\mathbf{x}\right)\mathrm{grad}[\mathrm{c}]\right] + \beta \mathrm{c} = f(\mathbf{x}, t) \quad \text{in } \Omega \times \mathcal{I}, \tag{3.1a}$$

$$\mathrm{c}(\mathbf{x}, t) = \mathrm{c}^{\mathrm{p}}(\mathbf{x}, t) \quad \text{on } \Gamma^{\mathrm{D}} \times \mathcal{I}, \tag{3.1b}$$

$$-\widehat{\mathbf{n}}(\mathbf{x}) \cdot \mathbf{D}(\mathbf{x})\mathrm{grad}[\mathrm{c}] = q^{\mathrm{p}}(\mathbf{x}, t) \quad \text{on } \Gamma^{\mathrm{N}} \times \mathcal{I}, \text{ and} \tag{3.1c}$$

$$\mathrm{c}(\mathbf{x}, t = 0) = \mathrm{c}_0(\mathbf{x}) \quad \text{in } \Omega, \tag{3.1d}$$

where $\widehat{\mathbf{n}}(\mathbf{x})$ denotes the unit outward normal to the boundary, $\mathrm{c}_0(\mathbf{x})$ is the prescribed initial concentration, $\mathrm{c}^{\mathrm{p}}(\mathbf{x}, t)$ is the prescribed concentration on the boundary, $q^{\mathrm{p}}(\mathbf{x}, t)$ is the prescribed diffusive flux on the boundary, $f(\mathbf{x}, t)$ is the prescribed volumetric source/sink, and $\beta \geq 0$ is the coefficient of decay due to a chemical reaction.

As mentioned earlier, the mathematical scales in advective-diffusive-reactive systems can be characterized using popular non-dimensional numbers. A non-dimensional measure to identify the relative dominance of advection is the Péclet number, which can be defined as

$$P_e(\mathbf{x}, t) := \frac{L\|\mathbf{v}(\mathbf{x}, t)\|}{D(\mathbf{x})}, \tag{3.2}$$

where L is the characteristic length, D is the characteristic diffusivity, and $\|\cdot\|$ denotes the standard 2-norm. In the case of anisotropic diffusion tensor, $D(\mathbf{x})$ can be taken as the minimum eigenvalue of the diffusivity tensor at $\mathbf{x}$ (i.e., $D(\mathbf{x}) = \min\left\{\kappa \mid \det\left(\mathbf{D}(\mathbf{x}) - \kappa\mathbf{I}\right) = 0\right\}$). Clearly, the higher the Péclet number the greater will be the relative dominance of advection. A non-dimensional quantity to measure the relative dominance of the chemical reaction is the Damköhler number, which takes the form

$$D_a := \frac{\beta L^2}{D(\mathbf{x})}. \tag{3.3}$$

In the context of numerical solutions, the characteristic length is typically associated with an appropriate measure of the mesh size. A popular choice under the finite element method is $L = h_e/2$, where h_e is the diameter of the circumscribed circle of the element and the factor $1/2$ is for convenience. This choice gives rise to what is commonly referred to as the element Péclet number (e.g., see [62])

$$P_e^h = \frac{h_e \|\mathbf{v}(\mathbf{x},t)\|}{2D}, \tag{3.4}$$

which will be used in subsequent sections, especially, in defining stabilized weak formulations. We shall employ the semi-discrete methodology [63] based on the finite element method for spatial discretization and the trapezoidal family of time-stepping schemes for the temporal discretization.

Trapezoidal family of time-stepping schemes

The time interval of interest is divided into $\mathcal{N}$ sub-intervals such that

$$\mathcal{I} = (0,T] = \bigcup_{n=1}^{\mathcal{N}} \left(t^{(n-1)}, t^{(n)}\right], \tag{3.5}$$

where $t^{(0)} = 0$ and $t^{(\mathcal{N})} = T$. To make the presentation simple, we shall assume that the sub-intervals are uniform. That is,

$$t^{(n)} - t^{(n-1)} = \Delta t \quad \forall n = 1, \cdots, \mathcal{N}, \tag{3.6}$$

where Δt will be referred to as the time-step. However, it should be noted that the methods presented in this chapter can be easily extended to variable time-steps. The primary variable (which, in our case, will be the concentration) and the corresponding time derivative at discrete time levels are denoted as

$$d^{(n)} \approx \mathrm{c}(t = t^{(n)}), \quad v^{(n)} \approx \left.\frac{\partial \mathrm{c}}{\partial t}\right|_{t=t^{(n)}} . \tag{3.7}$$

The trapezoidal family of time-stepping schemes can be compactly written as

$$d^{(n+1)} = d^{(n)} + \Delta t \left((1-\vartheta) v^{(n)} + \vartheta v^{(n+1)} \right), \tag{3.8}$$

where $\vartheta \in [0, 1]$ is a user-specified parameter. Some popular time-stepping schemes under the trapezoidal family include the explicit Euler ($\vartheta = 0$, which is also known as the forward Euler), the midpoint rule ($\vartheta = 1/2$), and the implicit Euler ($\vartheta = 1$, which is also known as the backward Euler). The forward Euler method is an explicit scheme, and the midpoint and the backward Euler schemes are implicit. The stability and accuracy properties of these time-stepping schemes in the context of *ordinary differential equations* are well-known (e.g., see [48]).

Weak formulations

We will now present several weak formulations for the initial boundary value problem given by equations (3.1a)–(5.2b), which will be used in the remainder of the chapter. Since we address advection-dominated and reaction-dominated problems, we will present two popular stabilized weak formulations in addition to the Galerkin weak formulation. Let us introduce the function spaces

$$\mathsf{C}_t := \left\{ \mathrm{c}(\mathbf{x}, \cdot) \in H^1(\Omega) \mid \mathrm{c}(\mathbf{x}, t) = \mathrm{c}^{\mathrm{p}}(\mathbf{x}, t) \text{ on } \Gamma^{\mathrm{D}} \right\}, \text{ and} \tag{3.9a}$$

$$\mathsf{W} := \left\{ \mathrm{w}(\mathbf{x}) \in H^1(\Omega) \mid \mathrm{w}(\mathbf{x}) = 0 \text{ on } \Gamma^{\mathrm{D}} \right\}, \tag{3.9b}$$

where $H^1(\Omega)$ is a standard Sobolev space [64]. For convenience, we shall denote the standard L_2 inner-product over a set K as

$$(a; b)_K \equiv \int_K a \cdot b \, \mathrm{d}K. \tag{3.10}$$

The subscript K will be dropped if the set is the entire spatial domain (i.e., $K = \Omega$).

Galerkin weak formulation

The Galerkin formulation for the initial boundary value problem (3.1a)–(5.2b) can be written as follows: Find $\mathrm{c}(\mathbf{x}, t) \in \mathsf{C}_t$ such that we have

$$(\mathrm{w}; \partial \mathrm{c}/\partial t) + (\mathrm{w}; \mathrm{div}[\mathbf{v}\mathrm{c}]) + (\mathrm{grad}[\mathrm{w}]; \mathbf{D}(\mathbf{x})\mathrm{grad}[\mathrm{c}]) + (\mathrm{w}; \beta \mathrm{c} - f) = (\mathrm{w}; q^{\mathrm{p}})_{\Gamma^{\mathrm{N}}} \quad \forall \mathrm{w}(\mathbf{x}) \in \mathsf{W}. \tag{3.11}$$

It is well-known that the Galerkin formulation may exhibit numerical instabilities (e.g., spurious node-to-node oscillations) for non-self-adjoint spatial operators like the advective-diffusive and advective-diffusive-reactive systems. The reason can be attributed to the presence of boundary layers and interior layers in the solutions of these systems when advection is more dominant than the diffusion and reaction processes. Designing stable numerical formulations for advection-diffusion and advection-diffusion-reaction problems is still an active area of research (e.g., see [50, 65, 66]). This chapter is not concerned with developing new stabilized formulations.

In order to avoid spurious oscillations and obtain accurate numerical solutions, it is sufficient to have the element Péclet number to be smaller than unity. To put it differently, if the element Péclet number is greater than unity, the computational mesh may not be adequate to resolve the steep gradients due to boundary layers and internal layers, which are typical in the solutions of advection dominated problems. One can always achieve smaller values for the element Péclet number by refining the computational mesh adequately. However, in some cases, the mesh has to be so fine that it may be computationally prohibitive to employ such a mesh. In order to alleviate the deficiencies of the Galerkin formulation for advection-dominated problems, many alternative methods have been proposed in the literature. For example, see [55] for a short description and comparison of these methods. In this chapter, we shall employ the SUPG formulation [67] and the GLS formulation [68], which are two popular approaches employed to enhance the stability of the Galerkin formulation. For completeness and future reference, we now briefly outline these two stabilized formulations.

Streamline Upwind/Petrov-Galerkin (SUPG) weak formulation

The SUPG formulation reads as follows: Find $c(\mathbf{x},t) \in \mathsf{C}_t$ such that we have

$$\begin{aligned}
&(\mathrm{w}; \partial \mathrm{c}/\partial t) + (\mathrm{w}; \mathrm{div}[\mathbf{v}\mathrm{c}]) + (\mathrm{grad}[\mathrm{w}]; \mathbf{D}(\mathbf{x})\mathrm{grad}[\mathrm{c}]) + (\mathrm{w}; \beta \mathrm{c}), \\
&\qquad + \sum_{e=1}^{Nele} \left(\tau_{\mathrm{SUPG}} \mathbf{v} \cdot \mathrm{grad}[\mathrm{w}]; \partial \mathrm{c}/\partial t + \mathrm{div}\left[\mathbf{v}\mathrm{c} - \mathbf{D}(\mathbf{x})\mathrm{grad}[\mathrm{c}]\right] + \beta \mathrm{c} - f\right)_{\Omega_e}, \text{ and} \\
&\qquad = (\mathrm{w}; f) + (\mathrm{w}; q^{\mathrm{p}})_{\Gamma^{\mathrm{N}}} \quad \forall \mathrm{w}(\mathbf{x}) \in \mathsf{W},
\end{aligned} \tag{3.12}$$

where $Nele$ is the number of elements, and τ_{SUPG} is the stabilization parameter under the SUPG formulation. We shall use the stabilization parameter proposed in [69],

$$\tau_{\mathrm{SUPG}} = \frac{h_e}{2\|\mathbf{v}\|} \xi_0 \left(P_e^h\right), \quad \xi_0(\chi) = \coth(\chi) - \frac{1}{\chi}, \tag{3.13}$$

where h_e is the element length, and ξ_0 is known as the upwind function. Recall that P_e^h is the local (element) Péclet number.

Galerkin/least-squares (GLS) weak formulation

The GLS formulation reads as follows: Find $c(\mathbf{x}, t) \in \mathsf{C}_t$ such that we have

$$
\begin{aligned}
&(\mathrm{w}; \partial \mathrm{c}/\partial t) + (\mathrm{w}; \mathrm{div}[\mathbf{v}\mathrm{c}]) + (\mathrm{grad}[\mathrm{w}]; \mathbf{D}(\mathbf{x})\mathrm{grad}[\mathrm{c}]) + (\mathrm{w}; \beta \mathrm{c}),\\
&\qquad + \sum_{e=1}^{Nele} (\mathrm{w}/\Delta t + \mathrm{div}[\mathbf{v}\mathrm{w} - \mathbf{D}(\mathbf{x})\mathrm{grad}[\mathrm{w}]] + \beta \mathrm{w}; \tau_{\mathrm{GLS}} \left(\partial \mathrm{c}/\partial t + \mathrm{div}\left[\mathbf{v}\mathrm{c} - \mathbf{D}(\mathbf{x})\mathrm{grad}[\mathrm{c}]\right] + \beta \mathrm{c} - \right.\\
&\qquad = (\mathrm{w}; f) + (\mathrm{w}; q^{\mathrm{p}})_{\Gamma^{\mathrm{N}}} \quad \forall \mathrm{w}(\mathbf{x}) \in \mathsf{W},
\end{aligned}
\tag{3.14}
$$

where τ_{GLS} is the stabilization parameter under the GLS formulation, and Δt is the time-step. In this chapter, we shall take $\tau_{\mathrm{GLS}} = \tau_{\mathrm{SUPG}}$, which is a common practice. It should be emphasized that an optimal choice of stabilization parameter for stabilized formulations in two- and three-dimensions is still an active area of research (e.g., see [55]).

3.3 Proposed Multi-Time-Step Computational Framework

The proposed multi-time-step computational framework is built upon the semi-discrete methodology [63] and the dual Schur domain decomposition method [36]. The semi-discrete methodology converts the partial differential equations into a system of ordinary differential equations. For spatial discretization of the problem at hand, one can use either the Galerkin formulation or a stabilized formulation, which could depend on the relative strengths of transport processes and the decay coefficient due to chemical reactions. The dual Schur domain decomposition is an elegant

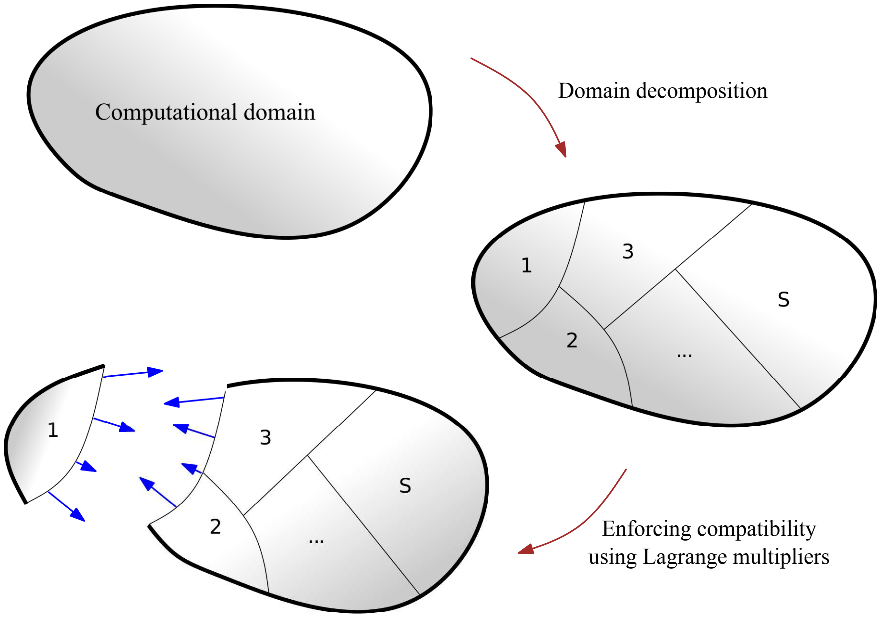


Figure 3.1: Non-overlapping domain decomposition: A pictorial description of computational domain and its decomposition into subdomains, subdomain interface, and interface interactions (i.e., Lagrange multipliers).

way to handle decomposition of the computational domain into subdomains through Lagrange multipliers.

Domain decomposition and the resulting equations

In order to facilitate multi-time-step coupling, we decompose the computational domain into $\mathcal{S}$ non-overlapping subdomains such that

$$\overline{\Omega} = \bigcup_{i=1}^{\mathcal{S}} \overline{\Omega}_i \quad \text{and} \quad \Omega_i \cap \Omega_j = \varnothing \text{ for } i \neq j, \tag{3.15}$$

where a superposed bar denotes the set closure. The meshes in all subdomains are assumed to be conforming along the subdomain interface, see Figure 3.1.

We shall use signed Boolean matrices to write the compatibility constraints along the subdomain interface, as they provide a systematic way to write the interface constraints as a system of linearly independent equations. Moreover, the mathematical structure of the resulting equations is suitable for a mathematical analysis. The entries of a signed Boolean matrix are either -1, 0, or 1, and each row has at most one non-zero entry. However, it needs to be emphasized that a signed Boolean matrix is never constructed explicitly in a computer implementation, as it is computationally not efficient to store such a matrix. It should also be noted that signed Boolean matrices can handle constraints arising from cross-points, which are the points on the subdomain interface that are connected to more than two subdomains. For more details on signed Boolean matrices see [9].

In a time-continuous setting, the governing equations after spatial discretization can be written as

$$\boldsymbol{M}_i\dot{\boldsymbol{c}}_i(t) + \boldsymbol{K}_i\boldsymbol{c}_i(t) = \boldsymbol{f}_i(t) + \boldsymbol{C}_i^{\mathrm{T}}\boldsymbol{\lambda}(t) \quad i = 1, \cdots, \mathcal{S}, \text{ and} \tag{3.16a}$$

$$\sum_{i=1}^{\mathcal{S}} \boldsymbol{C}_i\boldsymbol{c}_i(t) = \boldsymbol{0}, \tag{3.16b}$$

where a superposed dot denotes a derivative with respect to time, the subscript i denotes the subdomain number, the nodal concentration vector of the i-th subdomain is denoted by $\boldsymbol{c}_i$, the capacity matrix of the i-th subdomain is denoted by $\boldsymbol{M}_i$, the transport matrix of the i-th subdomain is denoted by $\boldsymbol{K}_i$, $\boldsymbol{f}_i(t)$ is the forcing vector of the i-th subdomain, $\boldsymbol{\lambda}$ denotes the vector of Lagrange multipliers, and $\boldsymbol{C}_i$ denotes the signed Boolean matrix for the i-th subdomain. Let the number of degrees-of-freedom in the i-th subdomain be denoted by N_i, and the number of degrees-of-freedom on the subdomain interface be denoted by N_λ. The size of $\boldsymbol{c}_i$ is $N_i \times 1$, and both the capacity and transport matrices of the i-th subdomain will be of the size $N_i \times N_i$. The size of $\boldsymbol{\lambda}$ will be $N_\lambda \times 1$, and the size of the signed Boolean matrix $\boldsymbol{C}_i$ will be

$N_\lambda \times N_i$.

It is imperative to note that the governing equations (3.16a)–(3.16b), which arise from domain decomposition, form a system of differential/algebraic equations (DAEs). For completeness and future reference we now present the necessary details about differential/algebraic equations.

Differential/algebraic equations

A differential/algebraic equation is an equation involving a set of independent variables, an unknown function of the independent variables, and derivatives of the functions with respect to the independent variables. Clearly, ordinary differential equations, and algebraic equations form subclasses of differential/algebraic equations. In this chapter, we are concerned with first-order differential/algebraic equations. Mathematically, a DAE in first-order form takes the form

$$\boldsymbol{w}\left(\dot{\boldsymbol{x}}(t), \boldsymbol{x}\left(t\right), t\right) = \boldsymbol{0} \quad t \in \mathcal{I}, \tag{3.17}$$

where t is the independent variable, and $\boldsymbol{x}(t)$ is the unknown function. It is well-known that solving a system of differential/algebraic equations numerically can be more difficult than solving a system of ordinary differential equations [22, 47]. A notion which is popularly employed to measure the difficulty of obtaining numerical solutions to a particular DAE is the *differential index*. The differential index of a DAE is the number of times one has to take derivatives of equation (3.17) in order to be able to derive an ODE by mere algebraic manipulations. It is obvious that a system of ODEs will have differential index of zero. A special form of DAEs which is of interest to us in this chapter is the Hessenberg index-2 DAE. It has the mathematical form

$$\dot{\boldsymbol{x}} = \boldsymbol{p}(\boldsymbol{x}, \boldsymbol{y}, t), \text{ and} \tag{3.18a}$$

$$\boldsymbol{0} = \boldsymbol{q}(\boldsymbol{x}), \tag{3.18b}$$

which consists of a system of ordinary differential equations along with a set of algebraic equations (i.e., constraints). This chapter concerns with differential/algebraic equations of differential index two or lower. Many of the constrained mechanical systems can be modeled using DAEs (e.g., see [39]). In the case of coupling algorithms, the compatibility of subdomains along the interfaces will appear as an algebraic constraint to the ODEs obtained from a finite element discretization. It is not possible to solve differential/algebraic equations analytically unless in some very special cases. Hence, one has to resort to numerical solutions. In this chapter, we shall restrict to time-stepping schemes from the trapezoidal family. However, the corresponding properties when applied to differential/algebraic equations can be different. For a detailed discussion on this topic see [47].

Time discretization

We now construct two multi-time-step coupling methods that can handle multiple subdomains, and can allow the use of different time-steps, different time-integrators and/or different numerical formulation in different subdomains. To this end, the time interval of interest is divided into non-overlapping intervals whose end points will be referred to as *system time-levels*. The algebraic compatibility constraints will be enforced at the system time-levels. For convenience, we shall assume that the system time-levels are uniform. The n-th system time-level will be denoted by $t^{(n)}$ and can be written as

$$t^{(n)} = n\Delta t \quad n = 0, 1, \cdots, \mathcal{N}, \tag{3.19}$$

where Δt is called the *system time-step*. The numerical time-integration of each subdomain will advance by the *subdomain time-step*. The subdomain time-step of the i-th subdomain will be denoted by Δt_i. Note that $\Delta t \geq \Delta t_i \; \forall i$. Furthermore, we shall assume that the ratio between the system and subdomain time-step is a natural number, and is denoted by η_i. That is,

$$\eta_i = \frac{\Delta t}{\Delta t_i}. \tag{3.20}$$

Figure 3.2 presents a pictorial description of the system and subdomain time-steps. In the rest of the chapter, we will use the notation to show the value of a variable at a time-level

$$x^{\left(n+\frac{j}{\eta_i}\right)} = x\left(t^{(n)} + j\Delta t_i\right), \text{ and} \tag{3.21}$$

$$t^{\left(n+\frac{j}{\eta_i}\right)} = t^{(n)} + j\Delta t_i. \tag{3.22}$$

Note that because of the enforcement of compatibility constraint at system time-levels only, the Lagrange multipliers can only be calculated at system time-levels. We shall linearly interpolate the Lagrange multipliers within system time-levels. That is,

$$\boldsymbol{\lambda}^{\left(n+\frac{j+1}{\eta_i}\right)} = \left(1 - \frac{j+1}{\eta_i}\right)\boldsymbol{\lambda}^{(n)} + \left(\frac{j+1}{\eta_i}\right)\boldsymbol{\lambda}^{(n+1)}. \tag{3.23}$$

As discussed earlier, coupling explicit and implicit time-stepping schemes is not straightforward in the case of first-order transient systems as compared with second-order systems. The proposed computational framework will employ different compatibility constraints in order to enforce continuity and to make an explicit/implicit coupling possible.

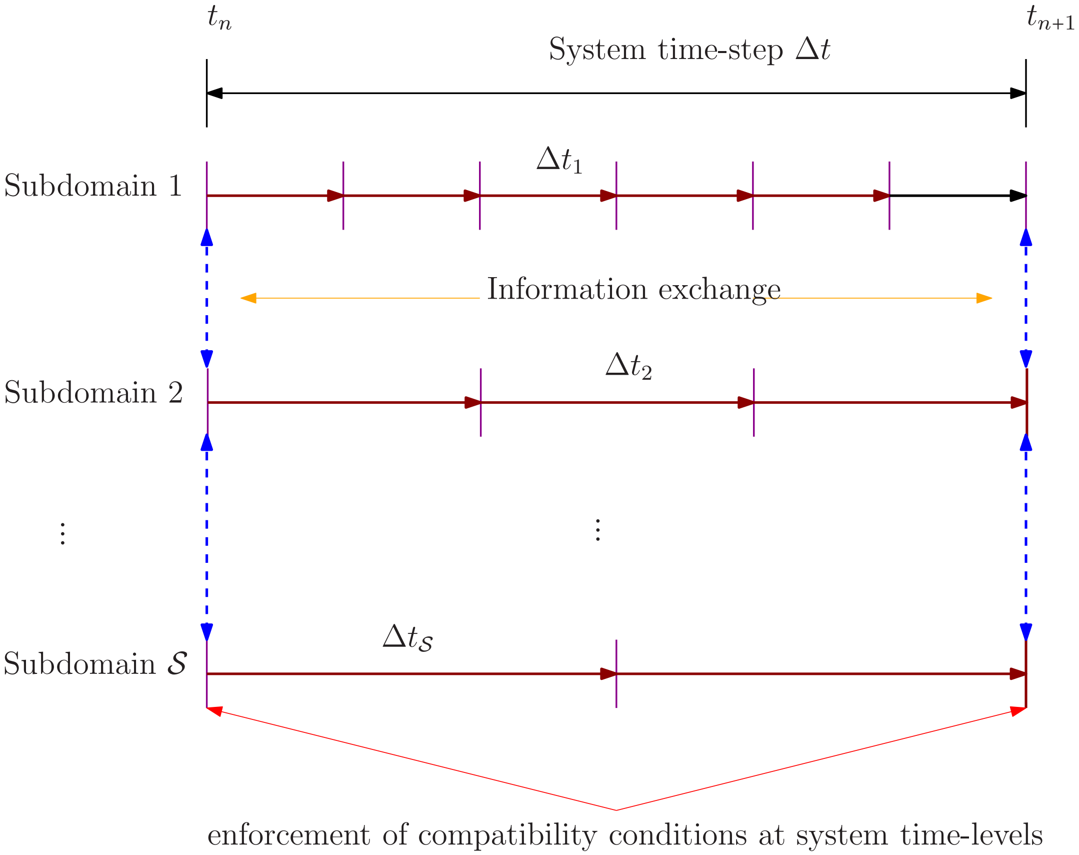


Figure 3.2: Multi-time-step integration: A pictorial description of time levels (t_n), system time-step (Δt), subdomain time-step (Δt_i), and subcycling. By subcycling in the i-th subdomain we mean that $\Delta t_i < \Delta t$.

Mathematical statements of the proposed coupling methods

The compatibility constraints along the subdomain interface will be enforced at system time-levels. Mathematically, the time discretization of compatibility constraints reads as

$$\sum_{i=1}^{\mathcal{S}} \boldsymbol{C}_i \boldsymbol{d}_i^{(n+1)} = \boldsymbol{0} \quad \forall n \qquad d\text{-continuity method and} \tag{3.24}$$

$$\sum_{i=1}^{\mathcal{S}} \boldsymbol{C}_i \left(\boldsymbol{v}_i^{(n+1)} + \frac{\alpha}{\Delta t} \boldsymbol{d}_i^{(n+1)} \right) = \boldsymbol{0} \quad \forall n \quad \text{Baumgarte stabilization,} \tag{3.25}$$

where $\alpha > 0$ is the Baumgarte stabilization parameter. The proposed coupling method based on d-continuity will read as follows: Find $\left(\boldsymbol{v}_i^{(n+(j+1)/\eta_i)}, \boldsymbol{d}_i^{(n+(j+1)/\eta_i)}, \boldsymbol{\lambda}^{(n+1)} \right)$ for $n = 1, ..., \mathcal{N}$; $j = 0, ..., \eta_i - 1$; and $i = 1, ..., \mathcal{S}$ such that we have

$$\boldsymbol{M}_i \boldsymbol{v}_i^{\left(n+\frac{j+1}{\eta_i}\right)} + \boldsymbol{K}_i \boldsymbol{d}_i^{\left(n+\frac{j+1}{\eta_i}\right)} = \boldsymbol{f}_i^{\left(n+\frac{j+1}{\eta_i}\right)} + \boldsymbol{C}_i^{\mathrm{T}} \boldsymbol{\lambda}^{\left(n+\frac{j+1}{\eta_i}\right)}, \tag{3.26a}$$

$$\boldsymbol{d}_i^{\left(n+\frac{j+1}{\eta_i}\right)} = \boldsymbol{d}_i^{\left(n+\frac{j}{\eta_i}\right)} + \Delta t_i \left(\left(1 - \vartheta_i\right) \boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)} + \vartheta_i \boldsymbol{v}_i^{\left(n+\frac{j+1}{\eta_i}\right)} \right), \tag{3.26b}$$

$$\boldsymbol{\lambda}^{\left(n+\frac{j+1}{\eta_i}\right)} = \left(1 - \frac{j+1}{\eta_i}\right) \boldsymbol{\lambda}^{(n)} + \left(\frac{j+1}{\eta_i}\right) \boldsymbol{\lambda}^{(n+1)}, \text{ and} \tag{3.26c}$$

$$\sum_{i=1}^{S} \boldsymbol{C}_i \boldsymbol{d}_i^{(n+1)} = \boldsymbol{0}\cdot \tag{3.26d}$$

The proposed coupling method based on the Baumgarte stabilization will read as follows: Find $\left(\boldsymbol{v}_i^{(n+(j+1)/\eta_i)}, \boldsymbol{d}_i^{(n+(j+1)/\eta_i)}, \boldsymbol{\lambda}^{(n+1)} \right)$ for $n = 1, ..., \mathcal{N}$; $j = 0, ..., \eta_i - 1$; and $i = 1, ..., \mathcal{S}$ such that we have

$$\boldsymbol{M}_i \boldsymbol{v}_i^{\left(n+\frac{j+1}{\eta_i}\right)} + \boldsymbol{K}_i \boldsymbol{d}_i^{\left(n+\frac{j+1}{\eta_i}\right)} = \boldsymbol{f}_i^{\left(n+\frac{j+1}{\eta_i}\right)} + \boldsymbol{C}_i^{\mathrm{T}} \boldsymbol{\lambda}^{\left(n+\frac{j+1}{\eta_i}\right)}, \tag{3.27a}$$

$$\boldsymbol{d}_i^{\left(n+\frac{j+1}{\eta_i}\right)} = \boldsymbol{d}_i^{\left(n+\frac{j}{\eta_i}\right)} + \Delta t_i \left(\left(1 - \vartheta_i\right) \boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)} + \vartheta_i \boldsymbol{v}_i^{\left(n+\frac{j+1}{\eta_i}\right)} \right), \tag{3.27b}$$

$$\boldsymbol{\lambda}^{\left(n+\frac{j+1}{\eta_i}\right)} = \left(1 - \frac{j+1}{\eta_i}\right) \boldsymbol{\lambda}^{(n)} + \left(\frac{j+1}{\eta_i}\right) \boldsymbol{\lambda}^{(n+1)}, \text{ and} \tag{3.27c}$$

$$\sum_{i=1}^{S} \boldsymbol{C}_i \left(\boldsymbol{v}_i^{(n+1)} + \frac{\alpha}{\Delta t} \boldsymbol{d}_i^{(n+1)} \right) = \boldsymbol{0}\cdot \tag{3.27d}$$

Before we perform a systematic theoretical analysis of the proposed multi-time-step coupling methods in the next section, it needs to be mentioned that the quantity

$\partial c/\partial t$ in the stabilization terms under the SUPG and GLS stabilized formulations (see equations (3.13) and (3.14)) will be evaluated at the weighted time-level $n+(j+\vartheta_i)/\eta_i$ for the i-th subdomain. This implies that this quantity in the stabilization terms for the i-th subdomain needs to be calculated as

$$\left.\frac{\partial c}{\partial t}\right|_{n+(j+\vartheta_i)/\eta_i} \approx (1-\vartheta_i)\boldsymbol{v}^{(n+j/\eta_i)} + \vartheta_i \boldsymbol{v}^{(n+(j+1)/\eta_i)} = \frac{\boldsymbol{d}^{(n+(j+1)/\eta_i)} - \boldsymbol{d}^{(n+j/\eta_i)}}{\Delta t_i}\cdot \qquad (3.28)$$

This form of discretization will be crucial in proving the stability of the proposed coupling methods. More details on the implementation of the proposed coupling methods can be be found in Appendix.

3.4 A Theoretical Study on the Proposed Methods

Notation

The jump and average operators over the i-th subdomain time-step are, respectively, defined as

$$\left[x^{\left(n+\frac{j}{\eta_i}\right)}\right]_i := x^{\left(n+\frac{j+1}{\eta_i}\right)} - x^{\left(n+\frac{j}{\eta_i}\right)}, \text{ and} \qquad (3.29a)$$

$$\left\langle x^{\left(n+\frac{j}{\eta_i}\right)}\right\rangle_i := \frac{1}{2}\left(x^{\left(n+\frac{j+1}{\eta_i}\right)} + x^{\left(n+\frac{j}{\eta_i}\right)}\right)\cdot \qquad (3.29b)$$

One can similarly define the jump and average operators over a system time-step as

$$[\![x^{(n)}]\!] := x^{(n+1)} - x^{(n)} = \sum_{j=0}^{\eta_i-1}\left[x^{\left(n+\frac{j}{\eta_i}\right)}\right]_i, \text{ and} \qquad (3.30a)$$

$$\langle\!\langle x^{(n)}\rangle\!\rangle := \frac{1}{2}\left(x^{(n+1)} + x^{(n)}\right)\cdot \qquad (3.30b)$$

Let $\boldsymbol{S}$ be a symmetric matrix, then we have

$$\langle\!\langle \boldsymbol{x}\rangle\!\rangle^{\mathrm{T}}\boldsymbol{S}[\![\boldsymbol{x}]\!] = \frac{1}{2}[\![\boldsymbol{x}^{\mathrm{T}}\boldsymbol{S}\boldsymbol{x}]\!]\cdot \qquad (3.31)$$

The trapezoidal family of time-stepping schemes applied over a subdomain time-step can be compactly written as

$$\left[\boldsymbol{d}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i = \Delta t_i \left(\left\langle \boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)} \right\rangle_i + \left(\vartheta_i - \frac{1}{2}\right)\left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i\right) \cdot \tag{3.32}$$

Stability analysis

Consistency of the proposed coupling methods is trivial by construction. Hence, for convergence, it is necessary and sufficient to show that the proposed coupling methods are stable. We now show that both the proposed coupling methods are indeed stable using the energy method [33]. For numerical stability analysis, it is common to assume that supply function is zero. Therefore, we take $\boldsymbol{f}_i(t) = \mathbf{0}$ in all the subdomains. Before we can provide stability proofs for the proposed coupling methods, we need to present an important property that the transport matrices enjoy under the three weak formulations that were outlined in the previous section. This property will play a crucial role in the stability analysis. We provide a proof for the Galerkin weak formulation.

Lemma 1 *Consider the Galerkin weak formulation given by equation* (3.11)*. If the advection velocity satisfies* $\mathrm{div}[\mathbf{v}] \geq 0$*, and the diffusivity tensor* $\mathbf{D}(\mathbf{x})$ *is symmetric and positive definite, then the symmetric part of the transport matrix resulting from the finite element discretization will be positive semi-definite.*

Proof 3 *Let us denote the spatial operator of the advective-diffusive system as*

$$\mathcal{L}[\mathrm{c}] := \mathrm{div}[\mathbf{v}\mathrm{c}] - \mathrm{div}[\mathbf{D}(\mathbf{x})\mathrm{grad}[\mathrm{c}]] \cdot \tag{3.33}$$

It is easy to show that the adjoint of the spatial operator takes the form

$$\mathcal{L}^*[\mathrm{c}] = -\mathbf{v} \cdot \mathrm{grad}[\mathrm{c}] - \mathrm{div}\left[\mathbf{D}^{\mathrm{T}}(\mathbf{x})\mathrm{grad}[\mathrm{c}]\right] \cdot \tag{3.34}$$

Noting the symmetry of diffusivity tensor, the symmetric part of the spatial operator takes the form

$$\widetilde{\mathcal{L}}[\mathrm{c}] = \frac{\mathcal{L}[\mathrm{c}] + \mathcal{L}^*[\mathrm{c}]}{2} = \frac{1}{2}\mathrm{div}[\mathbf{v}]\mathrm{c} - \mathrm{div}[\mathbf{D}(\mathbf{x})\mathrm{grad}[\mathrm{c}]] \cdot \tag{3.35}$$

The coefficient (i.e., "stiffness") matrix corresponding to the operator $\widetilde{\mathcal{L}}[\mathrm{c}]$ *over a finite element* Ω_e *can be written as*

$$\boldsymbol{K}_e = \int_{\Omega_e} \frac{1}{2}\mathrm{div}[\mathbf{v}]\boldsymbol{N}^{\mathrm{T}}(\mathbf{x})\boldsymbol{N}(\mathbf{x})\, \mathrm{d}\Omega + \int_{\Omega_e} \boldsymbol{B}(\mathbf{x})\mathbf{D}(\mathbf{x})\boldsymbol{B}^{\mathrm{T}}(\mathbf{x})\, \mathrm{d}\Omega, \tag{3.36}$$

where $\boldsymbol{N}(\mathbf{x})$ *is the row vector containing shape functions, and* $\boldsymbol{B}(\mathbf{x})$ *is the matrix containing the derivatives of shape functions with respect to* $\mathbf{x}$. *Since* $\mathrm{div}[\mathbf{v}] \geq 0$ *and* $\mathbf{D}(\mathbf{x})$ *is positive definite, the matrix* $\boldsymbol{K}_e$ *will be positive semi-definite. Since* $\mathbf{D}(\mathbf{x})$ *is symmetric, the matrix* $\boldsymbol{K}_e$ *is symmetric. The assembly procedure preserves the positive semi-definiteness when the local matrices are mapped to a global matrix.*

One can similarly show that the symmetric part of the transport matrix under the GLS formulation is also positive semi-definite. On the other hand, the symmetric part of the transport matrix under the SUPG formulation will be positive semi-definite only if the diffusivity tensor is constant, and low-order simplicial elements (e.g, two-node element, three-node triangle element, four-node tetrahedron element) are employed.

Theorem 3 (Stability of the d-continuity coupling method) *Under the proposed multi-time-step method with d-continuity, the rate variables $\boldsymbol{v}_i$ will remain bounded if $1/2 \leq \vartheta_i \leq 1 \; \forall i$.*

Proof 4 *Using the notation introduced earlier, one can write*

$$\boldsymbol{M}_i \left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i + \boldsymbol{K}_i \left[\boldsymbol{d}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i = \frac{1}{\eta_i}\boldsymbol{C}_i^{\mathrm{T}} \left[\!\left[\boldsymbol{\lambda}^{(n)}\right]\!\right], \text{ and} \tag{3.37a}$$

$$\sum_{i=1}^{\mathcal{S}} \boldsymbol{C}_i \left[\!\left[\boldsymbol{d}_i^{(n)}\right]\!\right] = \boldsymbol{0}, \tag{3.37b}$$

where interpolation of Lagrange multipliers using a first-order polynomial is used. For convenience, let us denote

$$\boldsymbol{Q}_i := \boldsymbol{M}_i + 2\left(\vartheta_i - \frac{1}{2}\right)\Delta t_i \,\mathrm{sym}\left[\boldsymbol{K}_i\right] \cdot \tag{3.38}$$

Clearly, the matrix $\boldsymbol{Q}_i$ is symmetric, as the matrix $\boldsymbol{M}_i$ is symmetric. Since the matrix $\boldsymbol{M}_i$ is positive definite, the symmetric part of $\boldsymbol{K}_i$ is positive semi-definite, $\vartheta_i \geq 1/2$, and $\Delta t_i > 0$; one can conclude that the matrix $\boldsymbol{Q}_i$ is positive definite.

Premultiplying both sides of equation (3.37a) *by* $\left[\boldsymbol{d}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i$ *and using equation* (3.32), *gives the equation*

$$\begin{aligned}\left\langle \boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right\rangle_i^{\mathrm{T}} \boldsymbol{Q}_i \left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i + \left(\vartheta_i - \frac{1}{2}\right)\left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i^{\mathrm{T}} \left(\boldsymbol{M}_i + (\vartheta_i - \frac{1}{2})\Delta t_i \,\mathrm{sym}\left[\boldsymbol{K}_i\right]\right)\left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i \\ + \Delta t_i \left\langle \boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right\rangle_i^{\mathrm{T}} \mathrm{sym}\left[\boldsymbol{K}_i\right] \left\langle \boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right\rangle_i = \frac{1}{\Delta t}\left[\!\left[\boldsymbol{\lambda}^{(n)}\right]\!\right]^{\mathrm{T}} \boldsymbol{C}_i \left[\boldsymbol{d}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i \cdot\end{aligned} \tag{3.39}$$

Since the symmetric part of $\boldsymbol{K}_i$ is positive semi-definite, and $\Delta t_i > 0$, we have the *inequality*

$$\left\langle \boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)} \right\rangle_i^{\mathrm{T}} \boldsymbol{Q}_i \left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)} \right]_i + \left(\vartheta_i - \frac{1}{2}\right) \left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)} \right]_i^{\mathrm{T}} \left(\boldsymbol{M}_i + (\vartheta_i - \frac{1}{2}) \Delta t_i \operatorname{sym}\left[\boldsymbol{K}_i \right] \right) \left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)} \right]_i \\ \leq \frac{1}{\Delta t} \left[\!\left[\boldsymbol{\lambda}^{(n)} \right]\!\right]^{\mathrm{T}} \boldsymbol{C}_i \left[\boldsymbol{d}_i^{\left(n+\frac{j}{\eta_i}\right)} \right]_i \cdot \tag{3.40}$$

The matrices $\boldsymbol{M}_i$ *and* $\operatorname{sym}\left[\boldsymbol{K}_i\right]$ *are positive definite and semidefinite respectively. In addition to that, if one has* $\vartheta_i \geq 1/2\ \forall i$*, then the following inequality can be derived*

$$\left\langle \boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)} \right\rangle_i^{\mathrm{T}} \boldsymbol{Q}_i \left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)} \right]_i \leq \frac{1}{\Delta t} \left[\!\left[\boldsymbol{\lambda}^{(n)} \right]\!\right]^{\mathrm{T}} \boldsymbol{C}_i \left[\boldsymbol{d}_i^{\left(n+\frac{j}{\eta_i}\right)} \right]_i \cdot \tag{3.41}$$

Summing over all the subdomain time levels (i.e., summing over j*), subdomains (i.e., summing over* i*), and using equation* (3.37b) *will give*

$$\sum_{i=1}^{\mathcal{S}} \sum_{j=0}^{\eta_i - 1} \left\langle \boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)} \right\rangle_i^{\mathrm{T}} \boldsymbol{Q}_i \left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)} \right]_i \leq 0 \cdot \tag{3.42}$$

Since the matrices $\boldsymbol{Q}_i$ *are symmetric, the above inequality can be rewritten as*

$$\sum_{i=1}^{\mathcal{S}} \sum_{j=0}^{\eta_i - 1} \left[\left(\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)} \right)^{\mathrm{T}} \boldsymbol{Q}_i \boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)} \right]_i \leq 0 \cdot \tag{3.43}$$

By executing the telescopic summation, we obtain

$$\sum_{i=1}^{\mathcal{S}} \left[\!\left[\left(\boldsymbol{v}_i^{(n)} \right)^{\mathrm{T}} \boldsymbol{Q}_i \boldsymbol{v}_i^{(n)} \right]\!\right] \leq 0 \cdot \tag{3.44}$$

This further implies

$$\sum_{i=1}^{\mathcal{S}} \boldsymbol{v}_i^{(n)\mathrm{T}} \boldsymbol{Q}_i \boldsymbol{v}_i^{(n)} \leq \sum_{i=1}^{\mathcal{S}} \boldsymbol{v}_i^{(n-1)\mathrm{T}} \boldsymbol{Q}_i \boldsymbol{v}_i^{(n-1)} \leq \cdots \leq \sum_{i=1}^{\mathcal{S}} \boldsymbol{v}_i^{(0)\mathrm{T}} \boldsymbol{Q}_i \boldsymbol{v}_i^{(0)} \cdot \tag{3.45}$$

Boundedness of $\boldsymbol{v}_i^{(0)}$ *and positive definiteness of matrices* $\boldsymbol{Q}_i$ *(*$i = 1, \cdots, \mathcal{S}$*) concludes the boundedness of* $\boldsymbol{v}_i^{(n)}$*, in all subdomains and at all time-levels.*

Remark 6 *One cannot relax the condition $\vartheta_i \geq 1/2$ under the coupling method based on the d-continuity method. It should be noted that one would obtain numerical instability if this condition is violated. This will be the case even if one does not employ subcycling [23]. However, the main advantage of employing the coupling method based on the d-continuity is that one can choose any system time-step and subdomain time-step, and still achieve numerical stability.*

We now assess the stability of the proposed coupling method based on the Baumgarte stabilization. We are able to construct a proof only for the case in which the matrices $\boldsymbol{K}_i$ are symmetric. This means that the proof does not hold for the case in which advection is present. However, the numerical results presented in a subsequent section show that the coupling method based on the Baumgarte stabilization provide stable solutions even in the presence of advection. It is therefore a good research problem to theoretically assess the stability of the coupling method based on the Baumgarte stabilization in the presence of advection.

Theorem 4 (Stability of the proposed method with Baumgarte stabilization) *Under the proposed multi-time-step method with Baumgarte stabilization, the rate variables $\boldsymbol{v}_i$, will remain bounded if one chooses $\Delta t_i \leq \Delta t_i^{\mathrm{critical}}$ and $\alpha \leq \alpha_{\mathrm{max}}$ where*

$$\Delta t_i^{\mathrm{critical}} := \begin{cases} \frac{2}{(1-2\vartheta_i)\omega_i} & \mathit{if} \quad 0 \leq \vartheta_i < 1/2 \\ +\infty & \mathit{if} \quad 1/2 \leq \vartheta_i \leq 1 \end{cases}, \tag{3.46a}$$

$$\alpha_{\mathrm{max}} := \begin{cases} \min\left\{\frac{2\eta_i}{1-2\vartheta_i} : 0 \leq \vartheta_i < 1/2\right\} \\ +\infty \quad \mathit{if} \quad 1/2 \leq \vartheta_i \leq 1 \ \forall i \end{cases}, \tag{3.46b}$$

and $\omega_i = \max\left\{\omega \,:\, \det\left(\omega \boldsymbol{I}_i - \boldsymbol{M}_i^{-1}\boldsymbol{K}_i\right) = 0\right\}$. *It is assumed that the matrices* $\boldsymbol{K}_i$ $(i = 1, \cdots, \mathcal{S})$ *are symmetric and positive semi-definite.*

Proof 5 *Consider the equations*

$$\boldsymbol{M}_i\left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i + \boldsymbol{K}_i\left[\boldsymbol{d}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i = \frac{1}{\eta_i}\boldsymbol{C}_i^{\mathrm{T}}\left\llbracket \boldsymbol{\lambda}^{(n)}\right\rrbracket, \text{ and} \tag{3.47a}$$

$$\sum_{i=1}^{S}\boldsymbol{C}_i\left(\left\llbracket \boldsymbol{v}_i^{(n)}\right\rrbracket + \frac{\alpha}{\Delta t}\left\llbracket \boldsymbol{d}_i^{(n)}\right\rrbracket\right) = \boldsymbol{0}\cdot \tag{3.47b}$$

Premultiplying both sides of equation (3.47a) *by* $\left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i + \frac{\alpha}{\Delta t}\left[\boldsymbol{d}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i$ *we obtain*

$$\left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i^{\mathrm{T}}\boldsymbol{M}_i\left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i + \frac{\alpha}{\Delta t}\left[\boldsymbol{d}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i^{\mathrm{T}}\boldsymbol{M}_i\left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i + \left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i^{\mathrm{T}}\boldsymbol{K}_i\left[\boldsymbol{d}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i$$
$$+ \frac{\alpha}{\Delta t}\left[\boldsymbol{d}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i^{\mathrm{T}}\boldsymbol{K}_i\left[\boldsymbol{d}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i = \frac{1}{\eta_i}\left\llbracket \boldsymbol{\lambda}^{(n)}\right\rrbracket^{\mathrm{T}}\boldsymbol{C}_i\left(\left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i + \frac{\alpha}{\Delta t}\left[\boldsymbol{d}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i\right) \tag{3.48}$$

Employing equation (3.32) *yields:*

$$\left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i^{\mathrm{T}}\left(\left(1+\alpha\left(\vartheta_i - \frac{1}{2}\right)\frac{\Delta t_i}{\Delta t}\right)\boldsymbol{M}_i + \Delta t_i\left(\vartheta_i - \frac{1}{2}\right)\left(1+\alpha\left(\vartheta_i - \frac{1}{2}\right)\frac{\Delta t_i}{\Delta t}\right)\boldsymbol{K}_i\right)\left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i$$
$$+ \left\langle \boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right\rangle_i^{\mathrm{T}}\left(\alpha\frac{\Delta t_i}{\Delta t}\boldsymbol{M}_i + \Delta t_i\left(1+2\alpha\left(\vartheta_i - \frac{1}{2}\right)\frac{\Delta t_i}{\Delta t}\right)\boldsymbol{K}_i\right)\left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i + \alpha\frac{\Delta t_i^2}{\Delta t}\left\langle \boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right\rangle_i^{\mathrm{T}}\boldsymbol{K}_i\left\langle \boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right\rangle_i$$
$$= \frac{1}{\eta_i}\left\llbracket \boldsymbol{\lambda}^{(n)}\right\rrbracket^{\mathrm{T}}\boldsymbol{C}_i\left(\left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i + \frac{\alpha}{\Delta t}\left[\boldsymbol{d}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i\right)\cdot \tag{3.49}$$

Note that the parameters α, Δt_i, *and* Δt *are strictly positive. The matrices* $\boldsymbol{K}_i$ *are assumed to be positive semi-definite. Thus, we have the inequality*

$$\left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i^{\mathrm{T}}\boldsymbol{P}_i\left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i + \left\langle \boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right\rangle_i^{\mathrm{T}}\boldsymbol{U}_i\left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i$$
$$\leq \left\llbracket \boldsymbol{\lambda}^{(n)}\right\rrbracket^{\mathrm{T}}\boldsymbol{C}_i\left(\left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i + \frac{\alpha}{\Delta t}\left[\boldsymbol{d}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i\right), \tag{3.50}$$

where

$$\boldsymbol{P}_i := \left(\eta_i + \alpha\left(\vartheta_i - \frac{1}{2}\right)\right)\boldsymbol{M_i} + \Delta t_i\left(\vartheta_i - \frac{1}{2}\right)\left(\eta_i + \alpha\left(\vartheta_i - \frac{1}{2}\right)\right)\boldsymbol{K}_i, \text{ and} \tag{3.51a}$$

$$\boldsymbol{U}_i := \alpha\boldsymbol{M}_i + \Delta t_i\left(\eta_i + 2\alpha\left(\vartheta_i - \frac{1}{2}\right)\right)\boldsymbol{K}_i\cdot \tag{3.51b}$$

Summing over all the subdomains (i.e., summing over i*) and subdomain time-levels (i.e., summing over* j*), gives the inequality*

$$\sum_{i=1}^{\mathcal{S}}\sum_{j=0}^{\eta_i-1}\left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i^{\mathrm{T}}\boldsymbol{P}_i\left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i + \sum_{i=1}^{\mathcal{S}}\sum_{j=0}^{\eta_i-1}\left\langle\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right\rangle_i^{\mathrm{T}}\boldsymbol{U}_i\left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i \le \left[\!\left[\boldsymbol{\lambda}^{(n)}\right]\!\right]^{\mathrm{T}}\sum_{i=1}^{\mathcal{S}}\boldsymbol{C}_i\left(\left[\!\left[\boldsymbol{v}_i^{(n)}\right]\!\right] + \frac{\alpha}{\Delta t}\left[\!\left[\boldsymbol{d}_i^{(n)}\right]\!\right]\right)\cdot \tag{3.52}$$

The compatibility condition along the subdomain interface in the form given by equation (3.52) *implies that*

$$\sum_{i=1}^{\mathcal{S}}\sum_{j=0}^{\eta_i-1}\left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i^{\mathrm{T}}\boldsymbol{P}_i\left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i + \sum_{i=1}^{\mathcal{S}}\sum_{j=0}^{\eta_i-1}\left\langle\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right\rangle_i^{\mathrm{T}}\boldsymbol{U}_i\left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i \le 0\cdot \tag{3.53}$$

From the hypothesis of the theorem, it is easy to show that the matrix $\boldsymbol{P}_i$ *is positive semi-definite. This implies that we have the inequality*

$$\sum_{i=1}^{\mathcal{S}}\sum_{j=0}^{\eta_i-1}\left\langle\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right\rangle_i^{\mathrm{T}}\boldsymbol{U}_i\left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i \le 0\cdot \tag{3.54}$$

It is easy to check that $\boldsymbol{U}_i$ *is symmetric, which implies*

$$\begin{aligned}\sum_{i=1}^{\mathcal{S}}\sum_{j=0}^{\eta_i-1}\left\langle\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right\rangle_i^{\mathrm{T}}\boldsymbol{U}_i\left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i &= \sum_{i=1}^{\mathcal{S}}\sum_{j=0}^{\eta_i-1}\frac{1}{2}\left[\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)^{\mathrm{T}}}\boldsymbol{U}_i\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right]_i \\ &= \frac{1}{2}\sum_{i=1}^{\mathcal{S}}\left[\!\left[\boldsymbol{v}_i^{(n)^{\mathrm{T}}}\boldsymbol{U}_i\boldsymbol{v}_i^{(n)}\right]\!\right] \le 0 \quad \forall n\cdot\end{aligned} \tag{3.55}$$

This further implies that

$$\sum_{i=1}^{\mathcal{S}}\boldsymbol{v}_i^{(n)^{\mathrm{T}}}\boldsymbol{U}_i\boldsymbol{v}_i^{(n)} \le \sum_{i=1}^{\mathcal{S}}\boldsymbol{v}_i^{(n-1)^{\mathrm{T}}}\boldsymbol{U}_i\boldsymbol{v}_i^{(n-1)} \le \cdots \le \sum_{i=1}^{\mathcal{S}}\boldsymbol{v}_i^{(0)^{\mathrm{T}}}\boldsymbol{U}_i\boldsymbol{v}_i^{(0)}\cdot \tag{3.56}$$

Since the matrices $\boldsymbol{U}_i$ $(i = 1, \cdots, \mathcal{S})$ are positive definite, and the initial rates $\boldsymbol{v}_i^{(0)}$ are bounded, one can conclude that the rate variables will remain bounded at all time-levels.

Bounds on drifts in concentrations and rate variables

A well-known phenomenon appearing in numerical solutions of DAEs is the drift in the compatibility/constraint equations [47]). In our case, the drift will manifest as discontinuity in the primary and/or rate variables along the subdomain interface. The drifts will be different for two proposed coupling methods, as they differ in handling compatibility conditions along the subdomain interface. Herein, we shall ignore subcycling (i.e., $\eta_i = 1 \ \forall i$), and assume that $\vartheta_i = \vartheta \ \forall i$. The following notation is employed:

$$\boldsymbol{d}_{\text{drift}}^{(n)} := \sum_{i=1}^{\mathcal{S}} \boldsymbol{C}_i \boldsymbol{d}_i^{(n)}, \text{ and } \quad \boldsymbol{v}_{\text{drift}}^{(n)} := \sum_{i=1}^{\mathcal{S}} \boldsymbol{C}_i \boldsymbol{v}_i^{(n)}. \tag{3.57}$$

Under the d-continuity coupling method, by construction of the method, there is no drift in the primary variable (i.e., concentration) along the subdomain interface at all system time levels. The drift in the rate satisfy the recursive relation

$$\boldsymbol{v}_{\text{drift}}^{(n+1)} = \left(1 - \frac{1}{\vartheta}\right) \boldsymbol{v}_{\text{drift}}^{(n)} \quad \forall n > 1. \tag{3.58}$$

Note that if the implicit Euler method (i.e., $\vartheta = 1$) is employed then the drifts at system time-levels will be zero in both concentrations and rate variables.

Under the proposed coupling method with Baumgarte stabilization, the recursive relations hold

$$\boldsymbol{d}_{\text{drift}}^{(n+1)} = \frac{1}{1+\alpha\vartheta}\boldsymbol{d}_{\text{drift}}^{(n)} + \frac{\Delta t\left(1-\vartheta\right)}{1+\alpha\vartheta}\boldsymbol{v}_{\text{drift}}^{(n)} \quad \forall n > 1, \text{ and} \tag{3.59a}$$

$$\boldsymbol{v}_{\text{drift}}^{(n+1)} = -\frac{\alpha}{\Delta t\left(1+\alpha\vartheta\right)}\boldsymbol{d}_{\text{drift}}^{(n)} - \frac{\alpha\left(1-\vartheta\right)}{1+\alpha\vartheta}\boldsymbol{v}_{\text{drift}}^{(n)} \quad \forall n > 1, \tag{3.59b}$$

which imply that choosing larger α will decrease drifts in concentration. It should be noted that subcycling, and mixed methods can have adverse effects on the drifts. That is, the drifts can be worse than predictions made by the above bounds. However, the above relations can be valuable to check a computer implementation, and can show a general trend of the drifts in the numerical time integration process. In a subsequent section, some numerical results are presented to corroborate the aforementioned theoretical predictions.

Influence of perturbations

In this section, we will study the propagation of perturbations over a system time-step. This analysis will help us to better understand how perturbations in input (in this case, previous time-level) will affect the solution at the next time-level. In the following theorem, we will consider application of the proposed method to non-linear DAEs of the form

$$\boldsymbol{M}_i\dot{\boldsymbol{c}}_i\left(t\right) = \boldsymbol{h}_i\left(\boldsymbol{c}_i\left(t\right),t\right) + \boldsymbol{C}_i^{\mathrm{T}}\boldsymbol{\lambda}\left(t\right) \quad \forall i, \text{ and} \tag{3.60}$$

$$\sum_{i=1}^{\mathcal{S}}\boldsymbol{C}_i\boldsymbol{c}_i\left(t\right) = \boldsymbol{0}\cdot \tag{3.61}$$

Theorem 5 *Let* $\left(\widehat{\boldsymbol{v}}_i^{(n+(j+1)/\eta_i)}, \widehat{\boldsymbol{d}}_i^{(n+(j+1)/\eta_i)}, \widehat{\lambda}^{(n+1)}\right)$ *with* $j = 1,\cdots,\eta_i - 1$ *and* $i = 1,\cdots,\mathcal{S}$

be the solution of the following system

$$\widehat{\boldsymbol{v}}_i^{\left(n+\frac{j+1}{\eta_i}\right)} = \boldsymbol{M}_i^{-1}\boldsymbol{h}_i\left(\widehat{\boldsymbol{d}}_i^{\left(n+\frac{j+1}{\eta_i}\right)}, t^{\left(n+\frac{j+1}{\eta_i}\right)}\right) + \boldsymbol{M}_i^{-1}\boldsymbol{C}_i^{\mathrm{T}}\widehat{\boldsymbol{\lambda}}^{\left(n+\frac{j+1}{\eta_i}\right)}, \tag{3.62a}$$

$$\widehat{\boldsymbol{d}}_i^{\left(n+\frac{j+1}{\eta_i}\right)} = \widehat{\boldsymbol{d}}_i^{\left(n+\frac{j+1}{\eta_i}\right)} + \Delta t_i\left(1-\vartheta_i\right)\widehat{\boldsymbol{v}}_i^{\left(n+\frac{j}{\eta_i}\right)} + \Delta t_i\vartheta_i\widehat{\boldsymbol{v}}_i^{\left(n+\frac{j+1}{\eta_i}\right)} + \Delta t_i\boldsymbol{\varepsilon}_{d_i}, \tag{3.62b}$$

$$\widehat{\boldsymbol{\lambda}}^{\left(n+\frac{j+1}{\eta_i}\right)} = \left(1-\frac{j+1}{\eta_i}\right)\widehat{\boldsymbol{\lambda}}^{(n)} + \left(\frac{j+1}{\eta_i}\right)\widehat{\boldsymbol{\lambda}}^{(n+1)} + \Delta t\boldsymbol{\Delta}_\lambda, \text{ and} \tag{3.62c}$$

$$\underbrace{\sum_{i=1}^{\mathcal{S}}\boldsymbol{C}_i\widehat{\boldsymbol{d}}_i^{(n+1)} = \boldsymbol{\varepsilon}_\lambda}_{d\text{-continuity}} \quad \text{or} \quad \underbrace{\sum_{i=1}^{\mathcal{S}}\boldsymbol{C}_i\left(\widehat{\boldsymbol{v}}_i^{(n+1)} + \frac{\alpha}{\Delta t}\widehat{\boldsymbol{d}}_i^{(n+1)}\right) = \frac{1}{\Delta t}\boldsymbol{\varepsilon}_\lambda\cdot}_{\text{Baumgarte stabilization}} \tag{3.62d}$$

in which we have assumed that

$$\boldsymbol{\Delta}_\lambda = O(\Delta t), \quad \boldsymbol{\varepsilon}_{d_i} = O(\Delta t_i), \text{ and} \quad \boldsymbol{\varepsilon}_\lambda = O(\Delta t^2)\cdot \tag{3.63}$$

Furthermore,

$$\widehat{\boldsymbol{v}}_i^{(n)} - \boldsymbol{v}_i^{(n)} = O(\Delta t_i), \quad \widehat{\boldsymbol{d}}_i^{(n)} - \boldsymbol{d}_i^{(n)} = O(\Delta t_i^2), \quad \widehat{\boldsymbol{\lambda}}^{(n)} - \boldsymbol{\lambda}^{(n)} = O(\Delta t)\cdot \tag{3.64}$$

Let the functions $\boldsymbol{M}_i^{-1}\boldsymbol{h}_i$ $(i = 1,\cdots,\mathcal{S})$ *be Lipschitz continuous, then the inequalities will hold*

$$\left\|\delta\boldsymbol{d}_i^{(n+1)}\right\| \le \mathcal{C}_d\left(\sum_{l=1}^{\mathcal{S}}\left(\left\|\delta\boldsymbol{d}_l^{(n)}\right\| + \Delta t\left\|\boldsymbol{\varepsilon}_{d_l}\right\|\right) + \Delta t\left\|\delta\boldsymbol{\lambda}^{(n)}\right\| + \phi\left\|\boldsymbol{\varepsilon}_\lambda\right\| + \Delta t^2\left\|\boldsymbol{\Delta}_\lambda\right\|\right), \tag{3.65a}$$

$$\left\|\delta\boldsymbol{v}_i^{(n+1)}\right\| \le \mathcal{C}_v\left(\sum_{l=1}^{\mathcal{S}}\left(\frac{1}{\Delta t}\left\|\delta\boldsymbol{d}_l^{(n)}\right\| + \left\|\boldsymbol{\varepsilon}_{d_l}\right\|\right) + \left\|\delta\boldsymbol{\lambda}^{(n)}\right\| + \frac{\phi}{\Delta t}\left\|\boldsymbol{\varepsilon}_\lambda\right\| + \Delta t\left\|\boldsymbol{\Delta}_\lambda\right\|\right), \text{ and} \tag{3.65b}$$

$$\left\|\delta\boldsymbol{\lambda}^{(n+1)}\right\| \le \mathcal{C}_\lambda\left(\sum_{l=1}^{\mathcal{S}}\left(\frac{1}{\Delta t}\left\|\delta\boldsymbol{d}_l^{(n)}\right\| + \left\|\boldsymbol{\varepsilon}_{d_l}\right\|\right) + \left\|\delta\boldsymbol{\lambda}^{(n)}\right\| + \frac{\phi}{\Delta t}\left\|\boldsymbol{\varepsilon}_\lambda\right\| + \Delta t\left\|\boldsymbol{\Delta}_\lambda\right\|\right), \tag{3.65c}$$

where $\mathcal{C}_d$, $\mathcal{C}_v$, $\mathcal{C}_\lambda$ *are constants,* $\delta\square = \widehat{\square} - \square$, *and*

$$\phi = \begin{cases} 1 & d-\text{continuity method} \\ \Delta t & \text{Baumgarte stabilization method} \end{cases} \tag{3.66}$$

Proof 6 *From equation* (3.62) *we can write*

$$\begin{aligned}\delta \boldsymbol{v}_i^{\left(n+\frac{j+1}{\eta_i}\right)} &= \boldsymbol{M}_i^{-1}\left(\boldsymbol{h}_i\left(\widehat{\boldsymbol{d}}_i^{\left(n+\frac{j+1}{\eta_i}\right)}, t^{\left(n+\frac{j+1}{\eta_i}\right)}\right) - \boldsymbol{h}_i\left(\boldsymbol{d}_i^{\left(n+\frac{j+1}{\eta_i}\right)}, t^{\left(n+\frac{j+1}{\eta_i}\right)}\right)\right) \\ &\quad + \boldsymbol{M}_i^{-1}\boldsymbol{C}_i^{\mathrm{T}}\left(\left(1-\frac{j+1}{\eta_i}\right)\delta\boldsymbol{\lambda}^{(n)} + \left(\frac{j+1}{\eta_i}\right)\delta\boldsymbol{\lambda}^{(n+1)} + \Delta t\boldsymbol{\Delta}_\lambda\right) \quad \forall i\cdot\end{aligned} \tag{3.67}$$

Lipschitz continuity of functions $\boldsymbol{M}_i^{-1}\boldsymbol{h}_i$ *and* $\boldsymbol{M}_i^{-1}\boldsymbol{C}_i^{\mathrm{T}}$ *can be used to obtain the inequalities*

$$\left\|\boldsymbol{M}_i^{-1}\boldsymbol{h}_i\left(\widehat{\boldsymbol{d}}_i^{\left(n+\frac{j+1}{\eta_i}\right)}, t^{\left(n+\frac{j+1}{\eta_i}\right)}\right) - \boldsymbol{M}_i^{-1}\boldsymbol{h}_i\left(\boldsymbol{d}_i^{\left(n+\frac{j+1}{\eta_i}\right)}, t^{\left(n+\frac{j+1}{\eta_i}\right)}\right)\right\| \le \mathcal{C}_i^h \left\|\delta\boldsymbol{d}_i^{\left(n+\frac{j+1}{\eta_i}\right)}\right\|, \text{ and} \tag{3.68a}$$

$$\left\|\boldsymbol{M}_i^{-1}\boldsymbol{C}_i^{\mathrm{T}}\left(\widehat{\boldsymbol{\lambda}}^{\left(n+\frac{j+1}{\eta_i}\right)} - \boldsymbol{\lambda}^{\left(n+\frac{j+1}{\eta_i}\right)}\right)\right\| \le \mathcal{C}_i^\lambda \left\|\delta\boldsymbol{\lambda}^{\left(n+\frac{j+1}{\eta_i}\right)}\right\|\cdot \tag{3.68b}$$

By taking norms of both sides of equation (3.67)*, and applying the triangle inequality, we obtain*

$$\begin{aligned}\left\|\delta\boldsymbol{v}_i^{\left(n+\frac{j+1}{\eta_i}\right)}\right\| &\le \mathcal{C}_i^h\left\|\delta\boldsymbol{d}_i^{\left(n+\frac{j+1}{\eta_i}\right)}\right\| + \left(1-\frac{j+1}{\eta_i}\right)\mathcal{C}_i^\lambda\left\|\delta\boldsymbol{\lambda}^{(n)}\right\| + \left(\frac{j+1}{\eta_i}\right)\mathcal{C}_i^\lambda\left\|\delta\boldsymbol{\lambda}^{(n+1)}\right\| + \Delta t\mathcal{C}_i^\lambda\left\|\boldsymbol{\Delta}_\lambda\right\| \\ &\le \mathcal{C}_i^h\left\|\delta\boldsymbol{d}_i^{\left(n+\frac{j+1}{\eta_i}\right)}\right\| + \mathcal{C}_i^\lambda\left\|\delta\boldsymbol{\lambda}^{(n)}\right\| + \mathcal{C}_i^\lambda\left\|\delta\boldsymbol{\lambda}^{(n+1)}\right\| + \Delta t\mathcal{C}_i^\lambda\left\|\boldsymbol{\Delta}_\lambda\right\|\cdot\end{aligned} \tag{3.69}$$

Note that $0 \le (j+1)/\eta_i \le 1\ \forall j$*. Using equation* (3.62b) *one can obtain the inequality*

$$\begin{aligned}\left\|\delta\boldsymbol{d}_i^{\left(n+\frac{j+1}{\eta_i}\right)}\right\| &\le \left\|\delta\boldsymbol{d}_i^{\left(n+\frac{j}{\eta_i}\right)}\right\| + \Delta t_i\left(1-\vartheta_i\right)\left\|\delta\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right\| + \Delta t_i\vartheta_i\left\|\delta\boldsymbol{v}_i^{\left(n+\frac{j+1}{\eta_i}\right)}\right\| + \Delta t_i\left\|\boldsymbol{\varepsilon}_{d_i}\right\| \\ &\le \left\|\delta\boldsymbol{d}_i^{\left(n+\frac{j}{\eta_i}\right)}\right\| + \Delta t_i\left\|\delta\boldsymbol{v}_i^{\left(n+\frac{j}{\eta_i}\right)}\right\| + \Delta t_i\left\|\delta\boldsymbol{v}_i^{\left(n+\frac{j+1}{\eta_i}\right)}\right\| + \Delta t_i\left\|\boldsymbol{\varepsilon}_{d_i}\right\|\cdot\end{aligned} \tag{3.70}$$

Inequalities (3.69) *and* (3.70) *imply*

$$\begin{aligned}\left(1-\Delta t_i\mathcal{C}_i^h\right)\left\|\delta\boldsymbol{d}_i^{\left(n+\frac{j+1}{\eta_i}\right)}\right\| &\le \left(1+\Delta t_i\mathcal{C}_i^h\right)\left\|\delta\boldsymbol{d}_i^{\left(n+\frac{j}{\eta_i}\right)}\right\| + 2\Delta t_i\mathcal{C}_i^\lambda\left\|\delta\boldsymbol{\lambda}^{(n)}\right\| + 2\Delta t_i\mathcal{C}_i^\lambda\left\|\delta\boldsymbol{\lambda}^{(n+1)}\right\| \\ &\quad + 2\Delta t_i\Delta t\mathcal{C}_i^\lambda\left\|\boldsymbol{\Delta}_\lambda\right\| + \Delta t_i\left\|\boldsymbol{\varepsilon}_{d_i}\right\|\cdot\end{aligned} \tag{3.71}$$

We shall assume that the subdomain time-steps Δt_i are sufficiently small such that $1 - \Delta t_i \mathcal{C}_i^h > 0$ holds. Then, the propagation of perturbations over a subdomain time-step will satisfy the inequality

$$\left\|\delta \boldsymbol{d}_i^{\left(n+\frac{j+1}{\eta_i}\right)}\right\| \leq \frac{1+\Delta t_i \mathcal{C}_i^h}{1-\Delta t_i \mathcal{C}_i^h}\left\|\delta \boldsymbol{d}_i^{\left(n+\frac{j}{\eta_i}\right)}\right\| + \frac{2\Delta t_i \mathcal{C}_i^\lambda}{1-\Delta t_i \mathcal{C}_i^h}\left\|\delta \boldsymbol{\lambda}^{(n)}\right\| + \frac{2\Delta t_i \mathcal{C}_i^\lambda}{1-\Delta t_i \mathcal{C}_i^h}\left\|\delta \boldsymbol{\lambda}^{(n+1)}\right\| \\ + \frac{2\Delta t_i \Delta t \mathcal{C}_i^\lambda}{1-\Delta t_i \mathcal{C}_i^h}\left\|\boldsymbol{\Delta}_\lambda\right\| + \frac{\Delta t_i}{1-\Delta t_i \mathcal{C}_i^h}\left\|\boldsymbol{\varepsilon}_{d_i}\right\| \cdot \qquad (3.72)$$

Applying the above inequality in a recursive manner, the following inequality can be obtained over a system time-step gives

$$\left\|\delta \boldsymbol{d}_i^{(n+1)}\right\| \leq \left(\frac{1+\Delta t_i \mathcal{C}_i^h}{1-\Delta t_i \mathcal{C}_i^h}\right)^{\eta_i}\left\|\delta \boldsymbol{d}_i^{(n)}\right\| + \left\{\sum_{k=0}^{\eta_i-1}\left(\frac{1+\Delta t_i \mathcal{C}_i^h}{1-\Delta t_i \mathcal{C}_i^h}\right)^k\right\}\left(\frac{2\Delta t_i \mathcal{C}_i^\lambda}{1-\Delta t_i \mathcal{C}_i^h}\left\|\delta \boldsymbol{\lambda}^{(n)}\right\| \right. \\ \left. + \frac{2\Delta t_i \mathcal{C}_i^\lambda}{1-\Delta t_i \mathcal{C}_i^h}\left\|\delta \boldsymbol{\lambda}^{(n+1)}\right\| + \frac{2\Delta t_i \Delta t \mathcal{C}_i^\lambda}{1-\Delta t_i \mathcal{C}_i^h}\left\|\boldsymbol{\Delta}_\lambda\right\| + \frac{\Delta t_i}{1-\Delta t_i \mathcal{C}_i^h}\left\|\boldsymbol{\varepsilon}_{d_i}\right\|\right) \cdot \qquad (3.73)$$

Similarly, one can derive other inequalities for the rate variables

$$\begin{aligned} \left\|\delta \boldsymbol{v}_i^{(n+1)}\right\| \leq{} & \mathcal{C}_i^h \left(\frac{1+\Delta t_i \mathcal{C}_i^h}{1-\Delta t_i \mathcal{C}_i^h}\right)^{\eta_i}\left\|\delta \boldsymbol{d}_i^{(n)}\right\| \\ & + \left\{\mathcal{C}_i^h \left\{\sum_{k=0}^{\eta_i-1}\left(\frac{1+\Delta t_i \mathcal{C}_i^h}{1-\Delta t_i \mathcal{C}_i^h}\right)^k\right\}\frac{2\Delta t_i \mathcal{C}_i^\lambda}{1-\Delta t_i \mathcal{C}_i^h} + \mathcal{C}_i^\lambda\right\}\left(\left\|\delta \boldsymbol{\lambda}^{(n)}\right\| + \left\|\delta \boldsymbol{\lambda}^{(n+1)}\right\|\right) \\ & + \left\{\mathcal{C}_i^h \left\{\sum_{k=0}^{\eta_i-1}\left(\frac{1+\Delta t_i \mathcal{C}_i^h}{1-\Delta t_i \mathcal{C}_i^h}\right)^k\right\}\frac{2\Delta t_i \Delta t \mathcal{C}_i^\lambda}{1-\Delta t_i \mathcal{C}_i^h} + \Delta t \mathcal{C}_i^\lambda\right\}\left\|\boldsymbol{\Delta}_\lambda\right\| \\ & + \mathcal{C}_i^h \left\{\sum_{k=0}^{\eta_i-1}\left(\frac{1+\Delta t_i \mathcal{C}_i^h}{1-\Delta t_i \mathcal{C}_i^h}\right)^k\right\}\frac{\Delta t_i}{1-\Delta t_i \mathcal{C}_i^h}\left\|\boldsymbol{\varepsilon}_{d_i}\right\| \cdot \end{aligned} \qquad (3.74)$$

From the perturbed constraint equations, we get the following inequality for the d-continuity method

$$\left\|\boldsymbol{\varepsilon}_\lambda\right\| = \left\|\sum_{i=1}^{\mathcal{S}} \boldsymbol{C}_i \boldsymbol{d}_i^{(n+1)}\right\| \leq \sum_{i=1}^{\mathcal{S}}\left\|\boldsymbol{d}_i^{(n+1)}\right\| \cdot \qquad (3.75)$$

Similarly, the following inequality can be derived for the coupling method based on the Baumgarte stabilization

$$\|\boldsymbol{\varepsilon}_\lambda\| = \left\|\sum_{i=1}^{\mathcal{S}} \boldsymbol{C}_i \left(\boldsymbol{v}_i^{(n+1)} + \frac{\alpha}{\Delta t}\boldsymbol{d}_i^{(n+1)}\right)\right\| \leq \sum_{i=1}^{\mathcal{S}} \left(\left\|\boldsymbol{v}_i^{(n+1)}\right\| + \frac{\alpha}{\Delta t}\left\|\boldsymbol{d}_i^{(n+1)}\right\|\right). \tag{3.76}$$

By substituting inequalities (3.73) *and* (3.74) *in the above inequalities, one can obtain the desired inequality for* $\left\|\delta\boldsymbol{\lambda}^{(n+1)}\right\|$. *By substituting the resulting inequality in* (3.73) *and* (3.74), *one can obtain the desired inequalities for* $\left\|\delta\boldsymbol{d}_i^{(n+1)}\right\|$ *and* $\left\|\delta\boldsymbol{v}_i^{(n+1)}\right\|$.

Remark 7 *The difference in the order of the perturbation in the algebraic constraints in* (3.62) *arises due to the difference in the differential index of the governing DAEs. That is, the d-continuity method form a system of DAEs of index 2, whereas the coupling method based on the Baumgarte stabilization form a system of DAEs of index 1. One can also decide on the order of perturbations based on dimensional analysis and consistency.*

3.5 Benchmark Problems for Verification

In this section, several benchmark problems are solved to illustrate the accuracy of the proposed coupling methods, to verify numerically the theoretical predictions, and to check the computer implementation.

Split degree-of-freedom problem

The governing equations of the coupled system that is shown in Figure 3.3 take the following form:

$$m_1 \dot{\mathrm{c}}_1(t) + k_1 \mathrm{c}_1(t) = f_1(t) + \lambda(t) \tag{3.77a}$$

$$m_2 \dot{\mathrm{c}}_2(t) + k_2 \mathrm{c}_2(t) = f_2(t) - \lambda(t), \text{ and} \tag{3.77b}$$

$$\mathrm{c}_1(t) - \mathrm{c}_2(t) = 0, \tag{3.77c}$$

where $\lambda(t)$ is the Lagrange multiplier. The following parameters have been used in this numerical simulation:

$$m_1 = 100,\ m_2 = 1,\ k_1 = 1,\ k_2 = 100,\ f_1 = f_2 = 0 \tag{3.78}$$

We shall solve the DAEs given by equations (3.77a)–(3.77c) using the proposed multi-time-step coupling methods, subject to the initial condition $\mathrm{c}_1(t = 0) = \mathrm{c}_2(t = 0) = 1$.

Performance of the d-continuity method

Figure 3.4 shows the results of numerical solution to (3.77) using the proposed coupling method with d-continuity. Implicit Euler method (i.e., $\vartheta_1 = 1$) is used to integrate the first subdomain, and the second subdomain is integrated using the midpoint rule (i.e., $\vartheta_2 = 1/2$). The results are shown for several different choices of system and subdomain time-steps (see Table 3.1). As shown earlier, the proposed method is stable under d-continuity if $\vartheta_i \geq 1/2$ in all subdomains. Enforcing d-continuity, assures the continuity of primary variable (which will be the concentration

Table 3.1: Split degree-of-freedom problem: Time-integration parameters for the d-continuity method.

Case	Δt	Δt_1	Δt_2	ϑ_1	ϑ_2
1	0.5	0.25	0.5	1	1/2
2	0.5	0.05	0.1	1	1/2
3	0.1	0.05	0.1	1	1/2

in this paper) along the interface at all system time-levels. The proposed methods shows good compatibility with the exact solution.

Performance of the Baumgarte stabilization

Baumgarte stabilization allows coupling explicit and implicit time-integrators in different subdomains. Midpoint rule is employed in the first subdomain (i.e., $\vartheta_1 = 1/2$). In this problem explicit Euler method is used in the second subdomain (i.e., $\vartheta_2 = 0$). As it can be seen in Figure 3.5, choice of system time-step Δt, and Baumgarte stabilization parameter α, influence the accuracy of the numerical result (see Table 3.2 for the values of integration parameters). The drift in the primary variables, u_1 and u_2, is nonzero. One can observe in Figure 3.5, that increasing the Baumgarte stabilization parameter α, or decreasing the system time-step Δt can improve the accuracy. Figure 3.6 shows the absolute error at time $t = 1$ vs. the system time-step. These figures show that despite subcycling (and using linear interpolation for Lagrange multipliers), the convergence rate remains close to that of the midpoint rule (which was used in all subdomains).

Table 3.2: Split degree-of-freedom problem: Time-integration parameters for the Baumgarte stabilization method.

Case	Δt	α	Δt_1	Δt_2	ϑ_1	ϑ_2
1	0.5	1.0	0.1	0.02	1/2	0
2	0.1	1.0	0.1	0.02	1/2	0
3	0.5	25.0	0.1	0.02	1/2	0

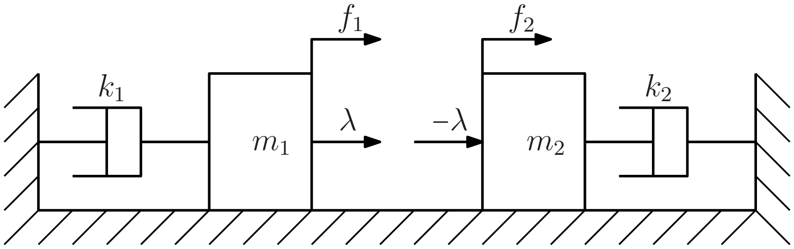


Figure 3.3: Split degree-of-freedom problem: A pictorial description.

One-dimensional problem

We will consider an unsteady diffusion with decay in one-dimension, which is an extension of the steady-state version considered in [70]. The governing equations can be written as follows:

$$\frac{\partial \mathrm{c}}{\partial t} + \mathrm{c} - \varepsilon^2 \frac{\partial^2 \mathrm{c}}{\partial \mathrm{x}^2} = 1 \qquad \mathrm{x} \in (0,1),\ t \in (0,T] \tag{3.79a}$$

$$\mathrm{c}(\mathrm{x}=0,t) = \mathrm{c}(\mathrm{x}=1,t) = 0 \quad t \in (0,T], \text{ and} \tag{3.79b}$$

$$\mathrm{c}(\mathrm{x}, t=0) = 0 \qquad \mathrm{x} \in (0,1). \tag{3.79c}$$

It is well-known that the solution of this singularly perturbed problem will exhibit boundary layers for small values of ε. Herein, we have taken $\varepsilon = 0.01$. We shall demonstrate the benefits of using the proposed multi-time-step coupling methods to problems in which the behavior of the solution can be very different in various regions of the computational domain.

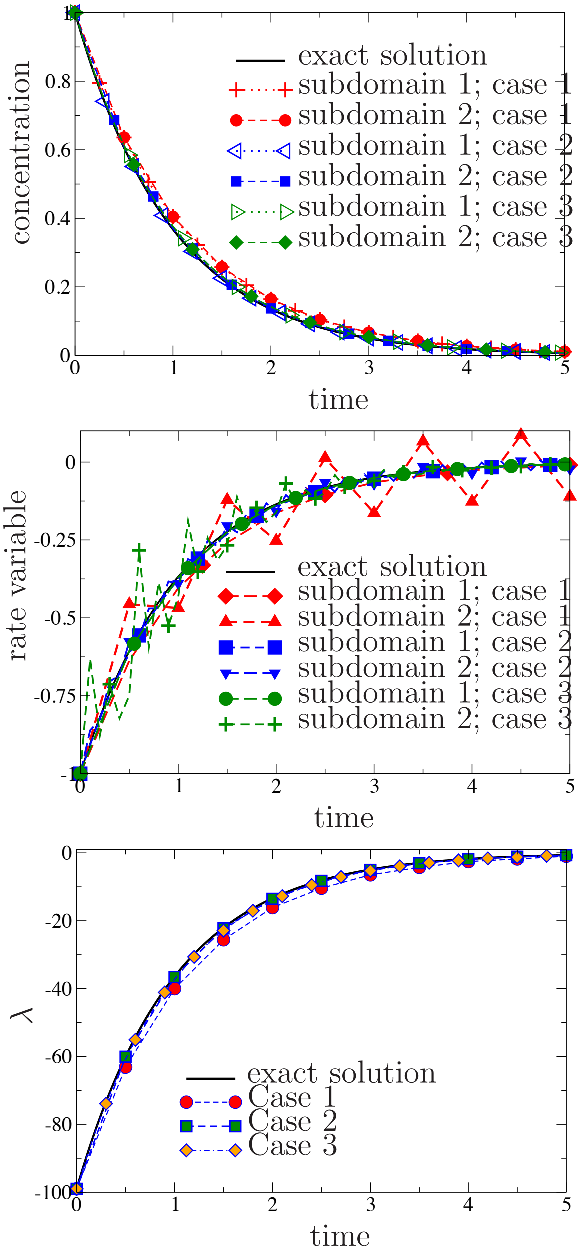


Figure 3.4: Split degree-of-freedom problem: We have employed the multi-time-step coupling method based on d-continuity method. The values of concentrations, rate variables, Lagrange multipliers, and drifts are compared with their respective exact solutions.

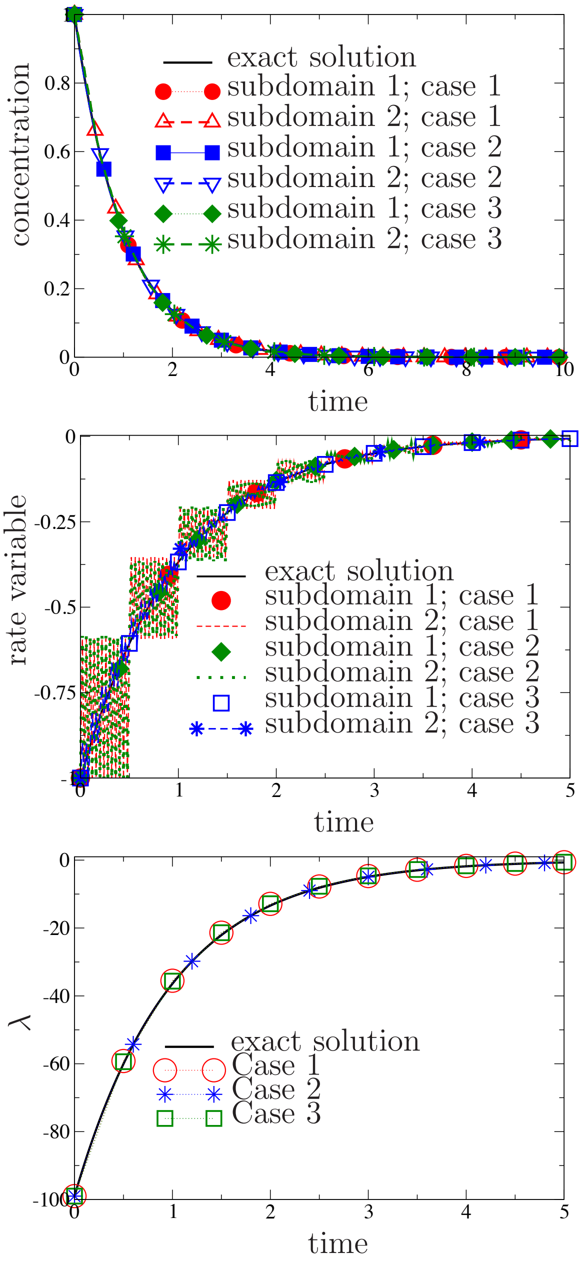
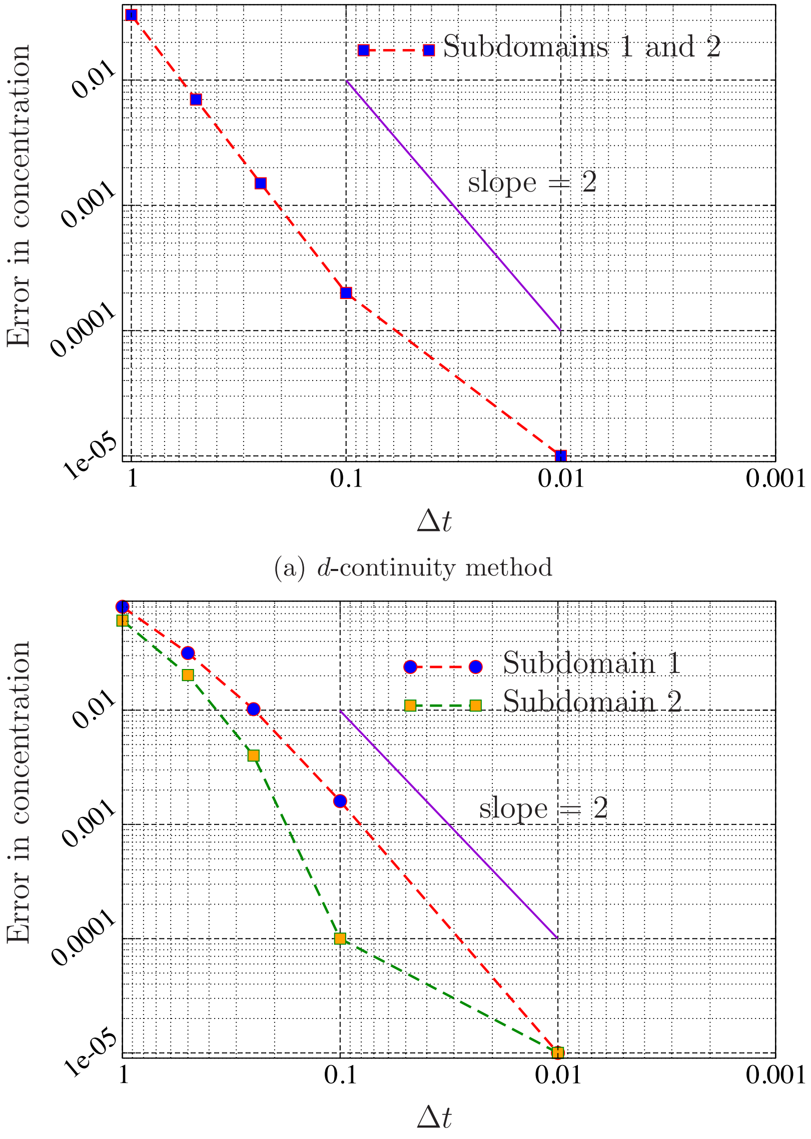


Figure 3.5: Split degree-of-freedom problem: The values of concentrations, rate variables, Lagrange multipliers, and drifts are compared with their respective exact solutions. In this problem Baumgarte stabilization is used.



(a) d-continuity method

(b) Baumgarte stabilization method

Figure 3.6: Split degree-of-freedom problem: In these figures absolute error vs. system time-step at $t = 1$ is plotted. In all cases, the subdomain time-steps are $\Delta t_i = 0.01,\ i = 1, 2$.

The domain is decomposed into three subdomains, as shown in Figure 3.7. Subdomains 1 and 3 are the regions in which the boundary layers will appear. Note that these subdomains are meshed using much finer elements than subdomain 2. For time-integration variables, see tables 3.3 and 3.4. The numerical results obtained using the proposed multi-time-step coupling methods are shown in Figures 3.8 and 3.9. Results are in good agreement with the exact solution, and the boundary layers are captured accurately by the proposed coupling methods. The drifts in concentrations and rate variables are plotted in figures 3.10 and 3.11. This numerical experiment illustrates the following attractive features of the proposed coupling methods:

(a) The system time-step can be much larger than subdomain time-steps.

(b) For fixed subdomain time-steps, smaller system time-step will result in better accuracy.

(c) Under the coupling method based on the Baumgarte stabilization and fixed subdomain time-steps, decreasing system time-step and/or increasing the Baumgarte stabilization parameter will result in improved accuracy.

(d) Utilizing smaller time-steps in individual subdomains improves the accuracy of results in the respective subdomain.

Table 3.3: One-dimensional problem: Time-integration parameters for the d-continuity method.

Case	Δt	Δt_1	Δt_2	Δt_3	ϑ_1	ϑ_2	ϑ_3
1	0.25	0.05	0.25	0.05	1/2	1	1/2
2	0.25	0.05	0.01	0.05	1/2	1	1/2
3	0.1	0.1	0.1	0.1	1/2	1/2	1/2

Table 3.4: One-dimensional problem: Time-integration parameters for the Baumgarte stabilization method.

Case	Δt	α	Δt_1	Δt_2	Δt_3	ϑ_1	ϑ_2	ϑ_3
1	0.25	1	0.125	0.25	0.125	1/2	0	1/2
2	0.25	5	0.125	0.05	0.125	1/2	0	1/2
3	0.25	5	0.00125	0.25	0.00125	0	1	0
4	0.25	1	0.0025	0.25	0.0025	0	1	0
5	0.1	1	0.1	0.1	0.1	1/2	1/2	1/2

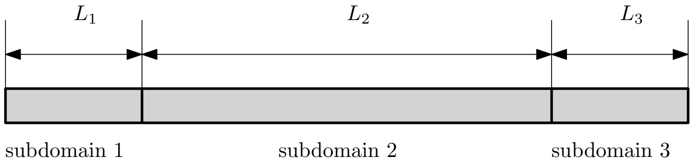
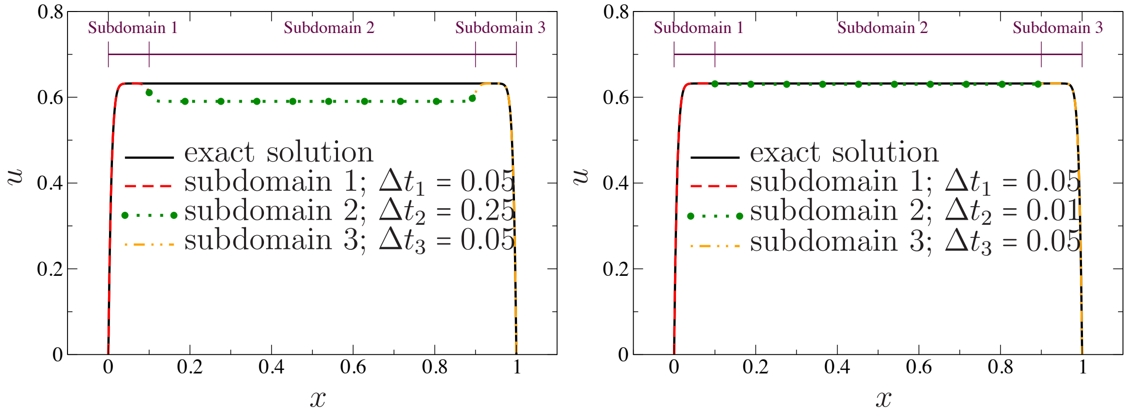


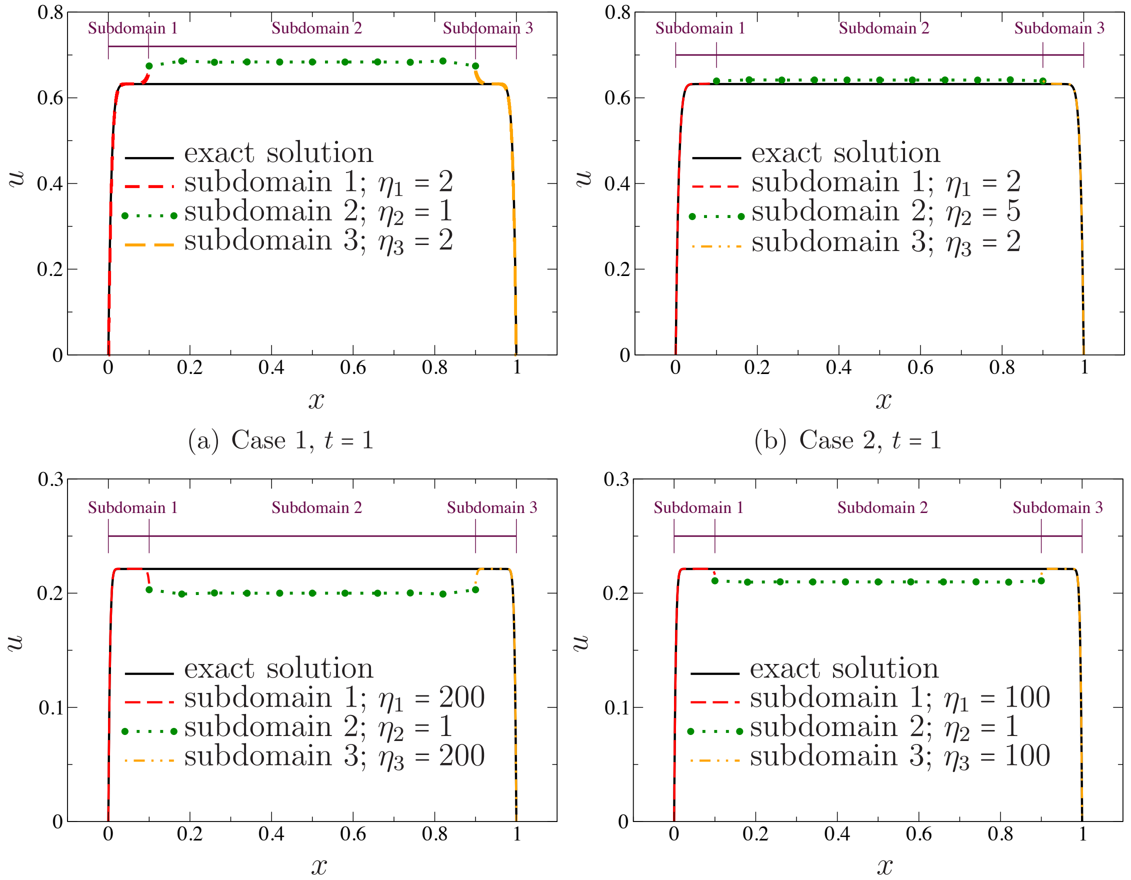
Figure 3.7: One-dimensional problem: The computational domain is divided into three subdomains of lengths $L_1 = 0.1$, $L_2 = 0.8$, and $L_3 = 0.1$. Two-node linear finite elements are used in all the subdomains.



(a) Case 1, $t = 1$

(b) Case 2, $t = 1$

Figure 3.8: One-dimensional problem: This figure compares the numerical solution obtained using the proposed d-continuity method to the exact solution. Each subdomain is meshed using 100 two-node finite elements.



(a) Case 1, $t = 1$

(b) Case 2, $t = 1$

(c) Case 3, $t = 0.25$

(d) Case 4, $t = 0.25$

Figure 3.9: One-dimensional problem: The numerical solution using the proposed coupling method with Baumgarte stabilization is shown in this figure. As it was shown in theorem 4, when conditionally stable trapezoidal schemes are used, multi-time-stepping can expand the acceptable values of α without compromising the stability of the coupling method.

Two-dimensional problem

A transient version of the well-known problem proposed by Hemker [71] will be considered. The governing equations take the following form:

$$\frac{\partial \mathrm{c}}{\partial t} + \frac{\partial \mathrm{c}}{\partial \mathrm{x}} - \varepsilon \mathrm{div}\left[\mathrm{grad}[\mathrm{c}]\right] = 0 \quad \text{in } \Omega, \tag{3.80a}$$

$$\mathrm{c}(\mathrm{x},\mathrm{y},t) = 1 \quad \text{on } \Gamma_1^{\mathrm{D}}, \tag{3.80b}$$

$$\mathrm{c}(\mathrm{x},\mathrm{y},t) = 0 \quad \text{on } \Gamma_2^{\mathrm{D}}, \tag{3.80c}$$

$$-\varepsilon \mathrm{grad}[\mathrm{c}] \cdot \widehat{\mathbf{n}}(\mathbf{x}) = 0 \quad \text{on } \Gamma^{\mathrm{N}}, \text{ and} \tag{3.80d}$$

$$\mathrm{c}(\mathrm{x},\mathrm{y},t = 0) = 0 \quad \text{in } \Omega\cdot \tag{3.80e}$$

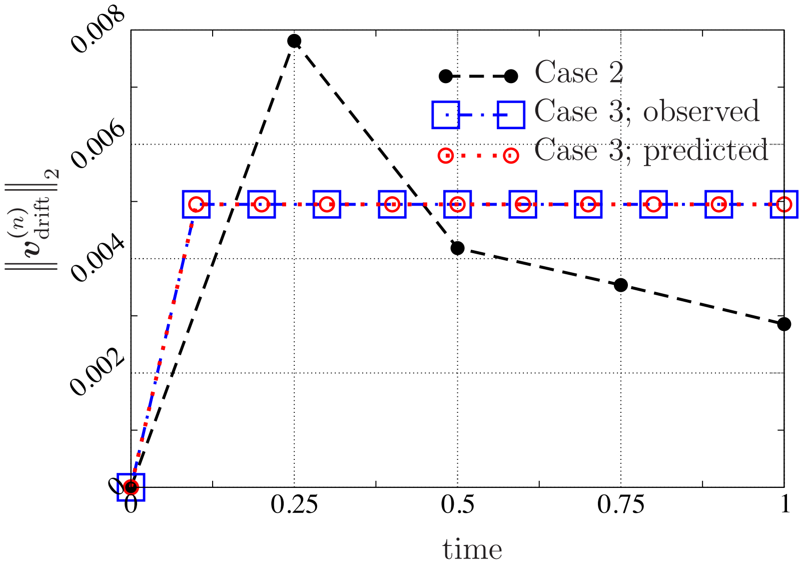
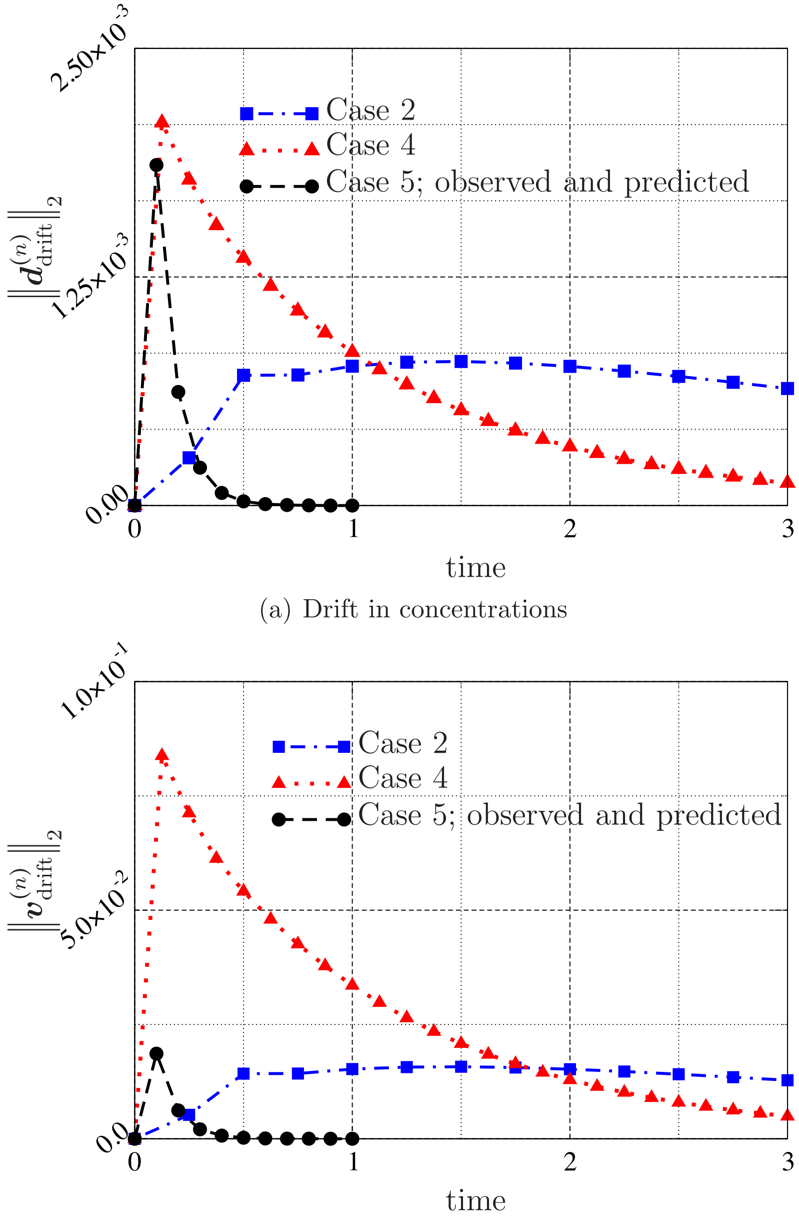


Figure 3.10: One-dimensional problem: This figure shows the 2-norm of the drift in the rate variable under the d-continuity method. (Note that, by algorithmic design, there will be no drift in the concentration along the subdomain interface at all system time levels.)

Computational domain, mesh, and domain decomposition are shown in Figures 3.12 and 3.13. In this problem, the advection velocity is $\mathbf{v} = (1, 0)$, and $\varepsilon = 0.01$. The problem at hand is a singularly perturbed equation and is known to exhibit both boundary and interior layers. Furthermore, the standard Galerkin formulation is known to produce spurious oscillations for small values of ε [50].

The numerical results obtained using the Galerkin weak formulation are shown in Figure 3.14. As expected, spurious oscillations occur at the vicinity of the circle. The minimum value observed in both cases is -0.439. The spurious oscillations and the violation of the non-negative constraint is because of using the Galerkin weak formulation, and is *not* due to the use of proposed multi-time-step coupling methods. To corroborate this claim, Figure 3.15 shows the results where tailored weak formulations are employed in different subdomains. The GLS formulation is used in



(a) Drift in concentrations

(b) Drift in rate variables

Figure 3.11: One-dimensional problem: This figure shows the drifts in the concentration and the rate variable for various cases under the Baumgarte stabilization method.

Table 3.5: Two-dimensional transient Hemker problem: Time-integration parameters for results using the standard Galerkin method.

Compatibility condition	Δt	α	Δt_1	Δt_2	Δt_3	ϑ_1	ϑ_2	ϑ_3
d-continuity method	0.1		0.001	0.01	0.1	1/2	1	1
Baumgarte stabilization	0.2	1	0.01	0.05	0.02	1/2	1	0

Table 3.6: Two-dimensional transient Hemker problem: Time-integration parameters for results using the GLS-SUPG-Galerkin formulations.

Compatibility condition	Δt	α	Δt_1	Δt_2	Δt_3	ϑ_1	ϑ_2	ϑ_3
d-continuity method	0.2		0.001	0.005	0.2	1/2	1	1
Baumgarte stabilization	0.2	1	0.001	0.005	0.02	1	1/2	0

subdomain 1, the SUPG formulation is employed in subdomain 2, and the Galerkin formulation in subdomain 3. There are no spurious oscillations and the minimum value observed is -0.062. In Figure 3.16, the ∞-norm of drift of concentrations from compatibility constraints is shown. There is no noticeable drift and in the case of Baumgarte stabilization method, the the drifts are controlled. Time integration parameters are given in tables 3.5 and 3.6. This example demonstrates choice of disparate time-steps, and different numerical formulations in different spatial regions of the computational domain.

3.6 Multi-Time-Step Transient Analysis of a Bimolecular Reaction

In this section, we shall apply the proposed multi-time-step coupling methods to a transport-controlled bimolecular reaction. This problem is of tremendous practical

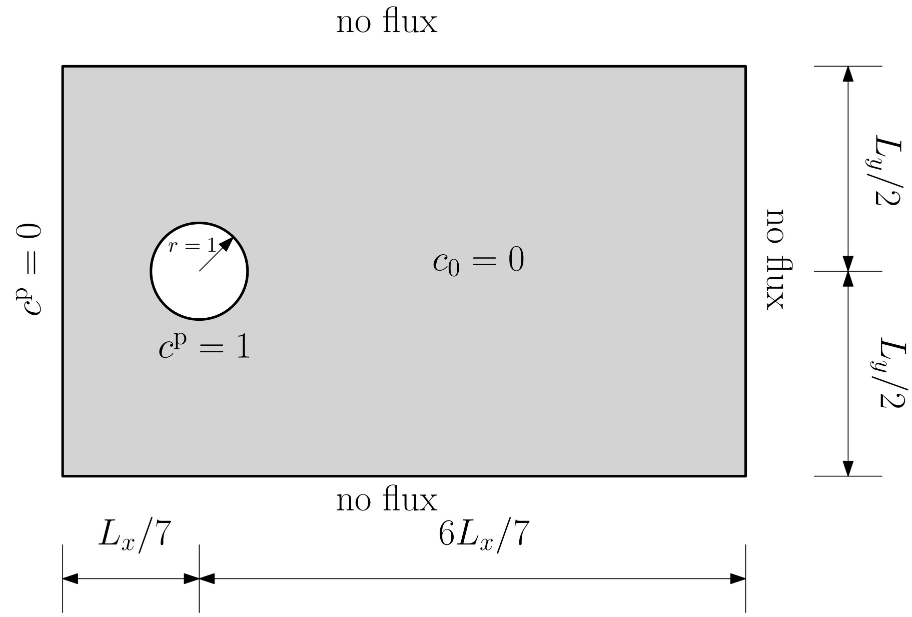


Figure 3.12: Two-dimensional transient Hemker problem: The dimensions of the computational domain are taken as $L_x = 14$ and $L_y = 8$. A circular hole is centered at the origin, and has a radius of unity.

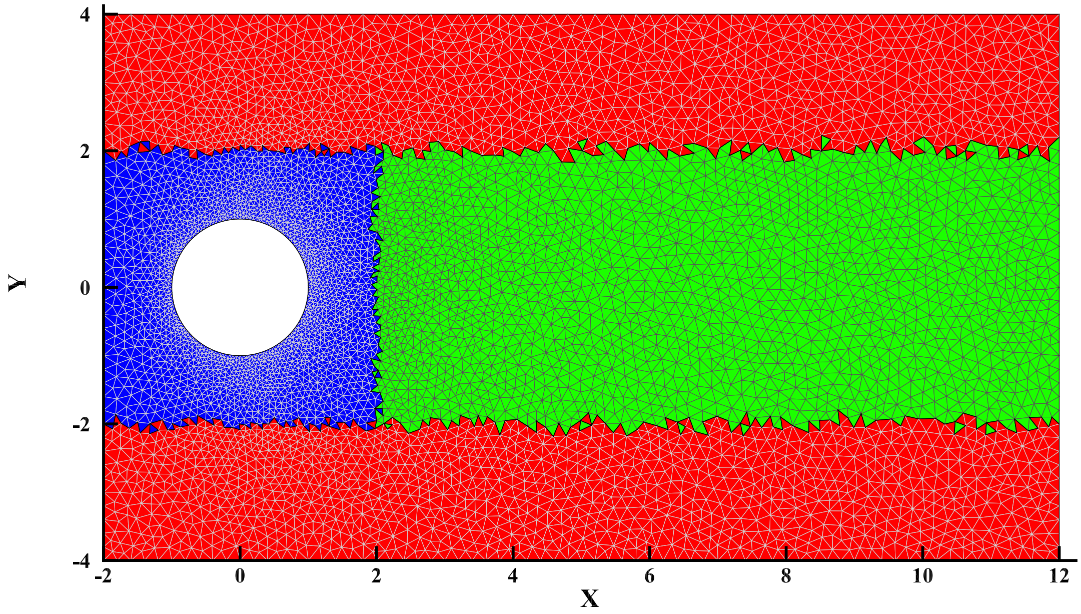
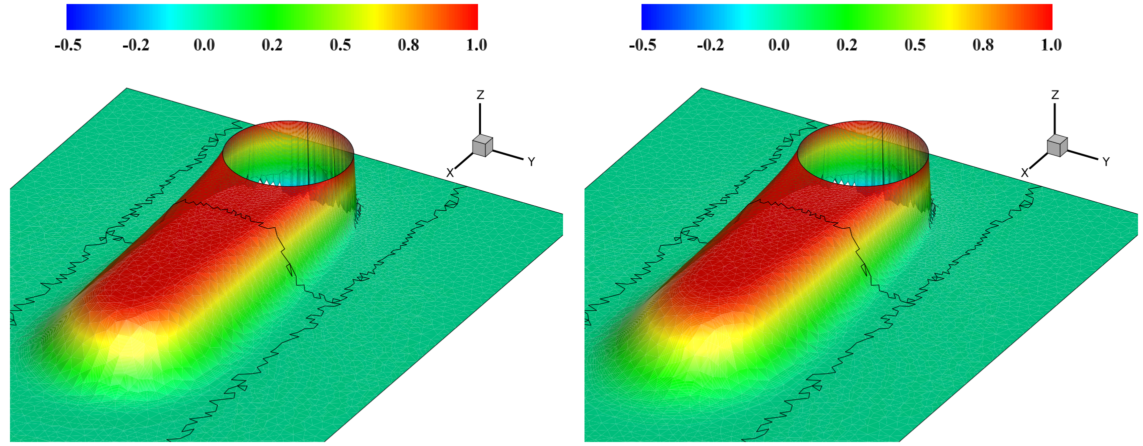


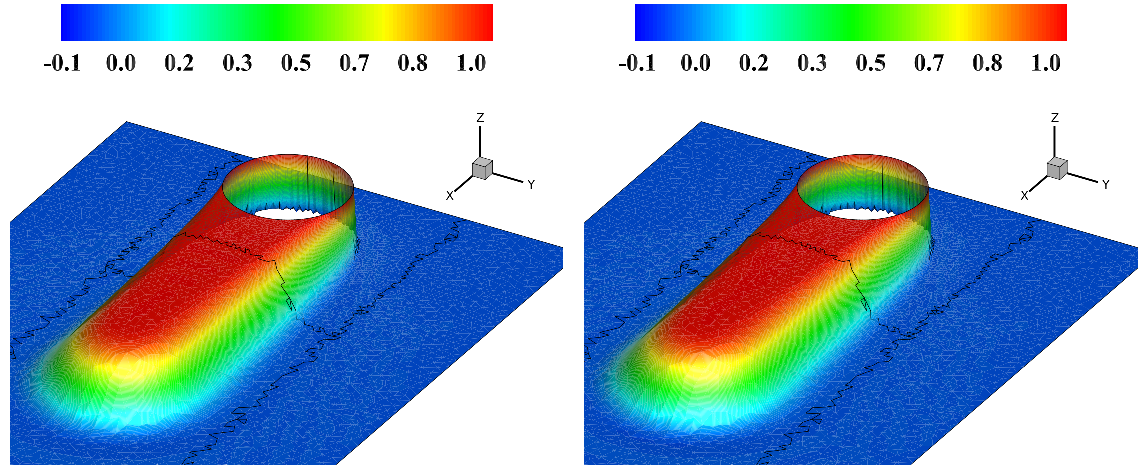
Figure 3.13: Two-dimensional transient Hemker problem: This figure shows the computational mesh, and the decomposition of the domain into subdomains. The computational domain is meshed using 11512 triangular finite elements using GMSH [1], and is partitioned into three subdomains.



(a) d-continuity method

(b) Baumgarte stabilization

Figure 3.14: Two-dimensional transient Hemker problem: The value of concentrations is shown on the domain of interest at $t = 5$. In this particular example, Galerkin weak formulation is employed. In figure (a), d-continuity method is employed to enforce continuity at the subdomain interface. Figure (b) shows the results when Baumgarte stabilization is employed to enforce continuity at the interface.



(a) d-continuity method

(b) Baumgarte stabilization

Figure 3.15: Two-dimensional transient Hemker problem: Concentrations at $t = 5$ are shown. GLS formulation is used in subdomain 1, SUPG formulation is used in subdomain 2, and the standard Galerkin formulation is used in subdomain 3.

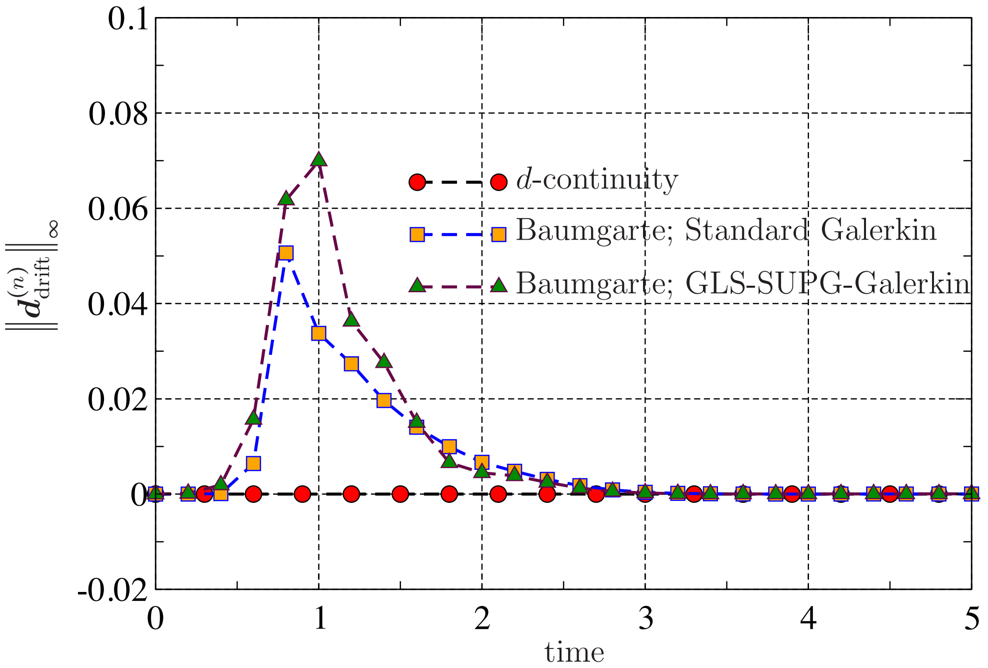


Figure 3.16: Two-dimensional transient Hemker problem: The maximum drift in concentrations is plotted against time. The time integration parameters are the same as in Figures 3.14 and 3.15.

importance in areas such as transverse mixing-limited chemical reactions in groundwater and aquifers, and mixing-controlled bioreactive transport in heterogeneous porous media arising in bioremediation. We shall now document the most important equations of the mathematical model. A more detailed discussion about the model can be found in [72], which however did not address multi-time-step coupling methods.

Mathematical model

Consider the irreversible chemical reaction

$$n_A A + n_B B \to n_C C, \tag{3.81}$$

where A, B and C are the chemical species participating in the reaction, and n_A, n_B and n_C are their respective (positive) stoichiometry coefficients. The fate of the reactants and the product are governed by coupled system of transient advection-diffusion-reaction equations. We shall assume the part of the boundary on which the Dirichlet boundary condition is enforced to be the same for the reactants and the product. Likewise is assumed for the Neumann boundary conditions. One can then find two invariants that are unaffected by the underlying reaction, which can be obtained via the linear transformations

$$\mathrm{c}_F := \mathrm{c}_A + \left(\frac{n_A}{n_C}\right)\mathrm{c}_C, \text{ and} \tag{3.82a}$$

$$\mathrm{c}_G := \mathrm{c}_B + \left(\frac{n_B}{n_C}\right)\mathrm{c}_C. \tag{3.82b}$$

The evolution of these invariants is given by the following uncoupled transient advection-diffusion equations

$$\frac{\partial \mathrm{c}_i}{\partial t} + \mathrm{div}\left[\mathbf{v}\mathrm{c}_i - \mathbf{D}(\mathbf{x})\mathrm{grad}[\mathrm{c}_i]\right] = f_i(\mathbf{x},t) \quad \text{in } \Omega \times \mathcal{I}, \tag{3.83a}$$

$$\mathrm{c}_i(\mathbf{x},t) = \mathrm{c}_i^{\mathrm{p}}(\mathbf{x},t) := \mathrm{c}_j^{\mathrm{p}}(\mathbf{x},t) + \left(\frac{n_j}{n_C}\right)\mathrm{c}_C^{\mathrm{p}}(\mathbf{x},t) \quad \text{on } \Gamma^{\mathrm{D}} \times \mathcal{I}, \tag{3.83b}$$

$$-\widehat{\mathbf{n}}(\mathbf{x}) \cdot \mathbf{D}(\mathbf{x})\mathrm{grad}[\mathrm{c}_i] = h_i^{\mathrm{p}}(\mathbf{x},t) := h_j^{\mathrm{p}}(\mathbf{x},t) + \left(\frac{n_j}{n_C}\right)h_C^{\mathrm{p}}(\mathbf{x},t) \quad \text{on } \Gamma^{\mathrm{N}} \times \mathcal{I}, \text{ and} \tag{3.83c}$$

$$\mathrm{c}_i(\mathbf{x},t=0) = \mathrm{c}_i^0(\mathbf{x}) := \mathrm{c}_j^0(\mathbf{x}) + \left(\frac{n_j}{n_C}\right)\mathrm{c}_C^0(\mathbf{x}) \quad \text{in } \Omega, \tag{3.83d}$$

where $i = F$ or G. We shall restrict to fast bimolecular reactions. That is, the time-scale of the chemical reaction is much smaller than the time-scale of the transport processes. For such situations, one can assume that the chemical species A and B cannot coexist at a spatial point and for a given instance of time. This implies that the concentrations of the reactants and the product can be obtained from the concentrations of the invariants through algebraic manipulations. To wit,

$$\mathrm{c}_A(\mathbf{x},t) = \max\left\{\mathrm{c}_F(\mathbf{x},t) - \left(\frac{n_A}{n_B}\right)\mathrm{c}_G(\mathbf{x},t), 0\right\} \tag{3.84a}$$

$$\mathrm{c}_B(\mathbf{x},t) = \left(\frac{n_B}{n_A}\right)\max\left\{-\mathrm{c}_F(\mathbf{x},t) + \left(\frac{n_A}{n_B}\right)\mathrm{c}_G(\mathbf{x},t), 0\right\}, \text{ and} \tag{3.84b}$$

$$\mathrm{c}_C(\mathbf{x},t) = \left(\frac{n_C}{n_A}\right)\left(\mathrm{c}_F(\mathbf{x},t) - \mathrm{c}_A(\mathbf{x},t)\right)\cdot \tag{3.84c}$$

Note that the solution procedure is still nonlinear, as the max{·,·} operator is nonlinear.

We shall employ the proposed multi-time-step computational framework to solve equations (3.83a)–(3.83d) to obtain concentrations of the invariants. Using the calculated values, we then find the concentrations for the reactants and the product using equations (3.84a)–(3.84c). The Galerkin formulation is employed in all subdomains. The negative values for the concentration are clipped at every subdomain time-step in the numerical simulations.

Numerical results for a diffusion-controlled bimolecular reaction

Consider a reaction chamber with $L_x = L_y = 1$, as shown in Figure 3.17. The computational domain is meshed using 5442 four-node quadrilateral elements. As shown in this figure, the domain is decomposed into four non-contiguous subdomains using METIS [2]. The diffusivity tensor is taken as

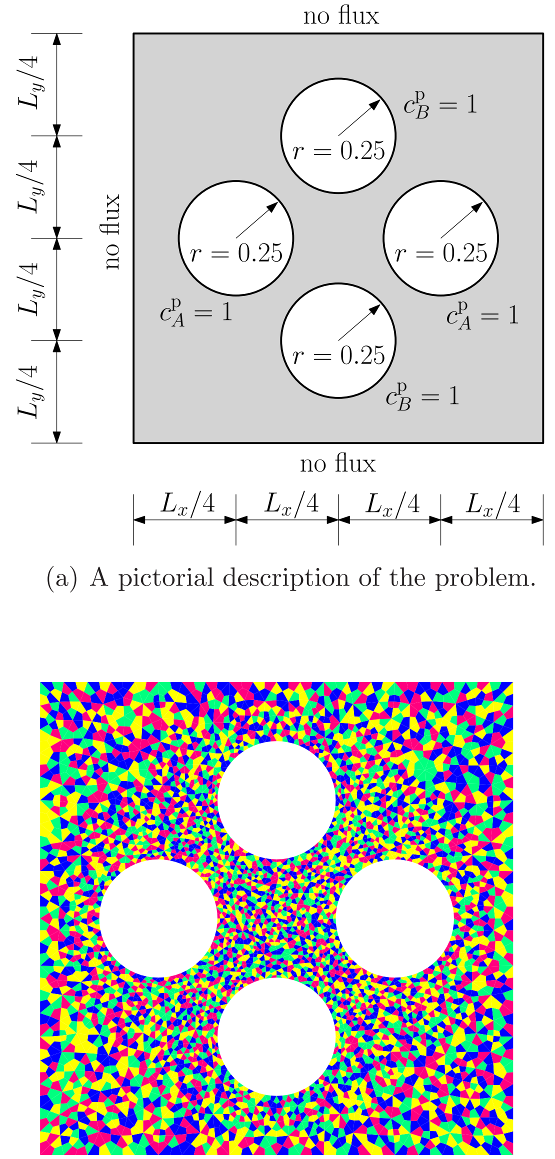
$$\mathbf{D}(\mathrm{x},\mathrm{y}) = \begin{bmatrix} \gamma \mathrm{x}^2 + \mathrm{y}^2 & -(1-\gamma)\,\mathrm{xy} \\ -(1-\gamma)\,\mathrm{xy} & \mathrm{x}^2 + \gamma \mathrm{y}^2 \end{bmatrix}, \tag{3.85}$$

where $\gamma = 0.001$. Baumgarte stabilization is employed to enforce compatibility along the subdomain interfaces with $\alpha = 100$. Implicit Euler method is employed in subdomains 1 and 3, and midpoint rule is employed in subdomains 2 and 4. The system time-step is taken as $\Delta t = 10^{-3}$, and the subdomain time-steps are taken as $\Delta t_1 = \Delta t_3 = 5 \times 10^{-4}$, and $\Delta t_2 = \Delta t_4 = 10^{-3}$.

Numerical results for the concentrations of the invariants, reactants and product are shown in Figures 3.18 and 3.19. As discussed earlier, Baumgarte stabilized coupling method can result in drift in the primary variable but it can be controlled using the stabilization parameter α. Equation (3.59) can serve as a valuable tool assessing the overall behavior of drifts with respect to system time-step, and the Baumgarte stabilization parameter α. Drifts for several choices of α and Δt are shown in Figure 3.20. Note that equation (3.59) assumes no subcycling, and no mixed time-integration.

Numerical results for a fast bimolecular reaction with advection

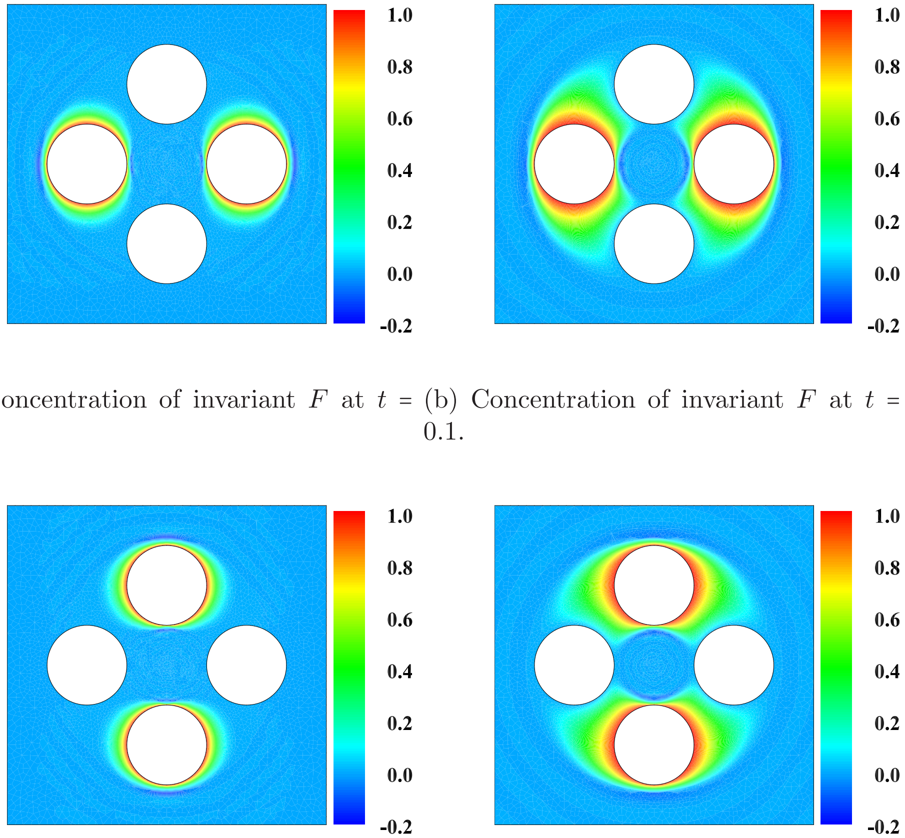
Consider a reaction chamber with $L_x = 4$ and $L_y = 1$, as shown in Figure 3.21(a). The computational domain is meshed using three-node triangular elements, and METIS [2] is employed to decomposed the domain into four non-contiguous subdomains using , as shown in Figure 3.21(b). There are 4151 interface constraints



(a) A pictorial description of the problem.

(b) Domain decomposition

Figure 3.17: Diffusion-controlled fast bimolecular reaction: The initial condition for the concentrations of all reactants is taken to be zero. The computational domain is meshed using 5442 four-node quadrilateral elements, and is divided into four subdomains using METIS [2].



(a) Concentration of invariant F at $t = 0.01$.

(b) Concentration of invariant F at $t = 0.1$.

(c) Concentration of invariant G at $t = 0.01$.

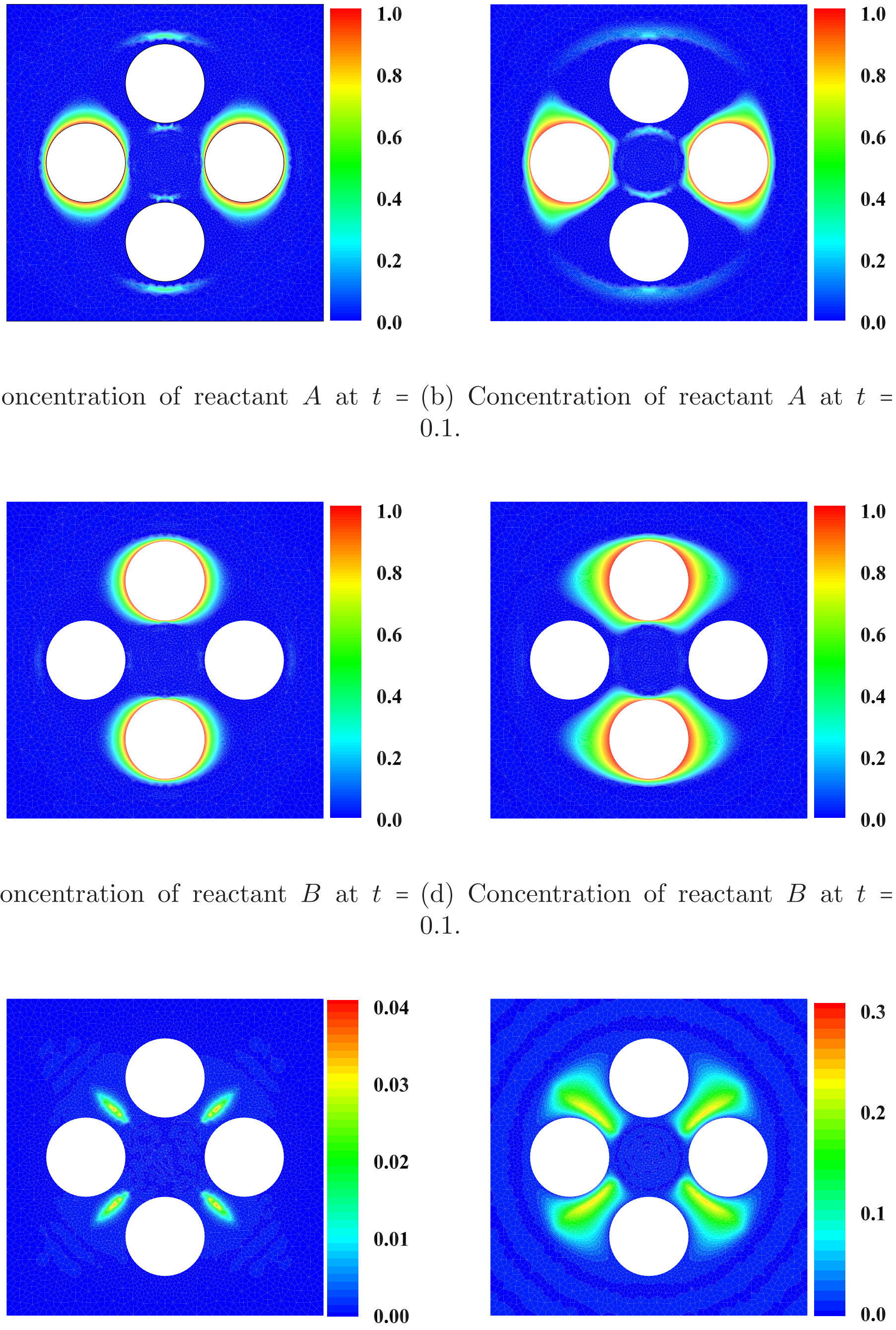
(d) Concentration of invariant G at $t = 0.1$.

Figure 3.18: Diffusion-controlled fast bimolecular reaction: This figure shows the concentrations of the invariants F and G at $t = 0.01$ and $t = 0.1$.

to ensure continuity of concentration along the subdomain interface. The diffusivity tensor is taken as

$$\mathbf{D}(\mathbf{x}) = \alpha_T \|\mathbf{v}\| \mathbf{I} + \frac{\alpha_L - \alpha_T}{\|\mathbf{v}\|} \mathbf{v} \otimes \mathbf{v}, \tag{3.86}$$

where $\mathbf{I}$ is the second-order identity tensor, $\otimes$ is the tensor product, $\|\cdot\|$ is the 2-norm, $\mathbf{v}(\mathbf{x})$ is the velocity, and α_L and α_T are, respectively, the longitudinal and transverse diffusivities. This form of diffusivity tensor is commonly employed in subsurface



(a) Concentration of reactant A at t = 0.01.

(b) Concentration of reactant A at t = 0.1.

(c) Concentration of reactant B at t = 0.01.

(d) Concentration of reactant B at t = 0.1.

(e) Concentration of product C at t = 0.01.

(f) Concentration of product C at t = 0.1.

Figure 3.19: Diffusion-controlled fast bimolecular reaction: Concentrations of the reactants and the product are shown at t = 0.01 and t = 0.1.

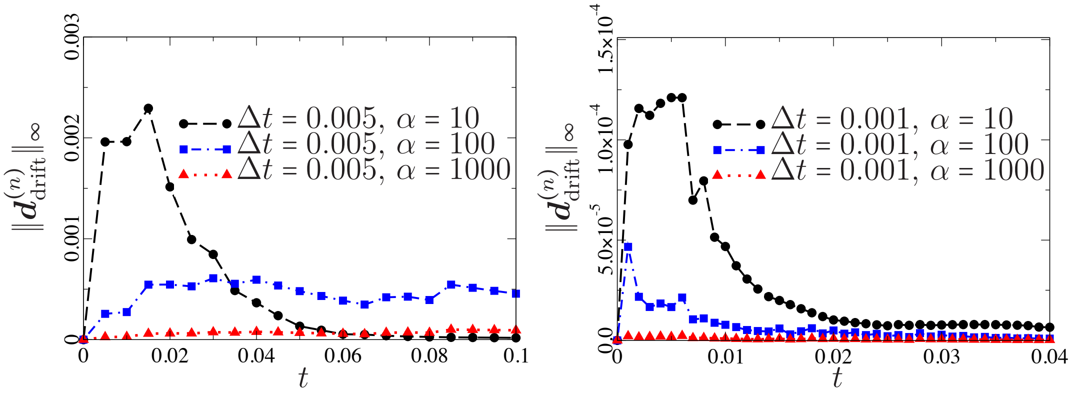


Figure 3.20: Diffusion-controlled fast bimolecular reaction: This figure shows the drift in the concentration of the chemical species C in the ∞-norm along the subdomain interface under the Baumgarte stabilization coupling method.

hydrology [73]. We define the velocity through the stream function

$$\psi(\mathrm{x},\mathrm{y}) = -\mathrm{y} - \sum_{k=1}^{3} A_k \cos\left(\frac{p_k \pi \mathrm{x}}{L_x} - \frac{\pi}{2}\right) \sin\left(\frac{q_k \pi \mathrm{y}}{L_y}\right) \cdot \tag{3.87}$$

The components of the advection velocity can then be calculated as

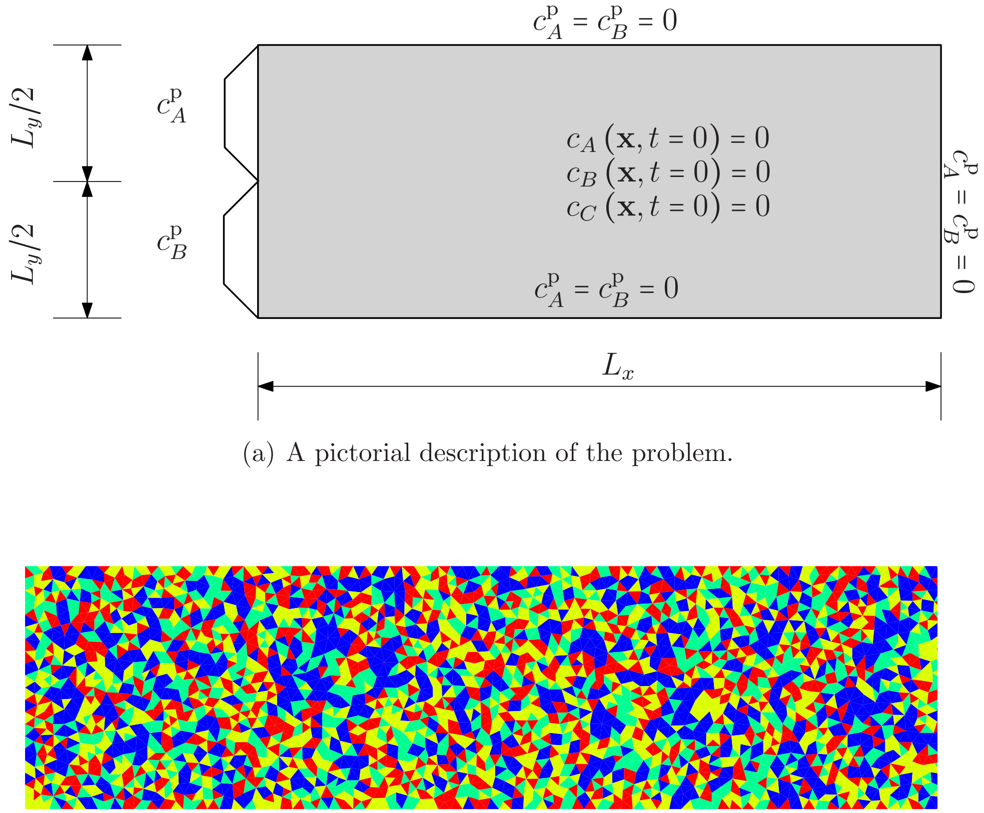
$$\mathrm{v}_x(\mathrm{x},\mathrm{y}) = -\frac{\partial \psi}{\partial \mathrm{y}}, \quad \mathrm{v}_y(\mathrm{x},\mathrm{y}) = +\frac{\partial \psi}{\partial \mathrm{x}} \cdot \tag{3.88}$$

The following parameters are used in the numerical simulation

$$p_1 = 4,\ p_2 = 5,\ p_3 = 10,\ q_1 = 1,\ q_2 = 5,\ q_3 = 10,\ \text{and}\ A_1 = 0.08,\ A_2 = 0.02,\ A_3 = 0.01 \cdot \tag{3.89}$$

The diffusivities are taken as $\alpha_L = 1$ and $\alpha_T = 10^{-4}$, and the prescribed concentrations on the boundary are taken as $c_A^{\mathrm{p}} = 1.0$ and $c_B^{\mathrm{p}} = 1.5$.

The numerical results for the concentration of the product at various time levels obtained using the d-continuity coupling method are shown in Figure 3.22, and there is no drift along the subdomain interface, which is expected under the proposed d-continuity method.

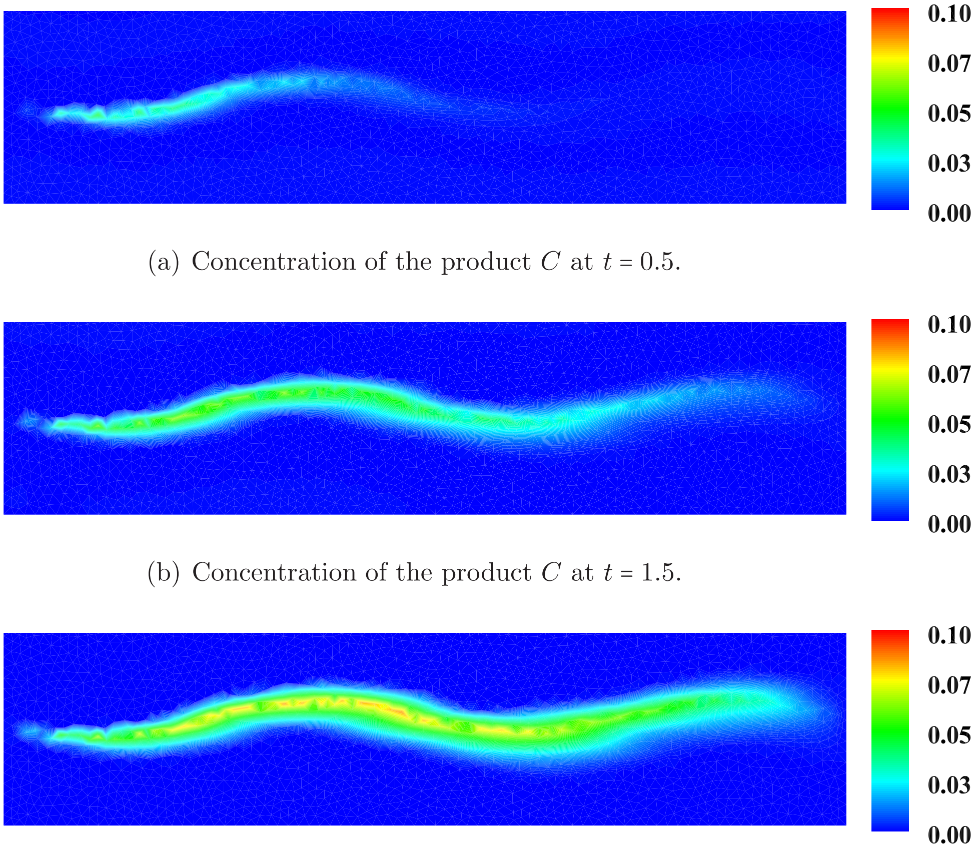

$c_A^{\mathrm{p}} = c_B^{\mathrm{p}} = 0$
$L_y/2$
$L_y/2$
c_A^{p}
c_B^{p}
$c_A\left(\mathbf{x}, t = 0\right) = 0$
$c_B\left(\mathbf{x}, t = 0\right) = 0$
$c_C\left(\mathbf{x}, t = 0\right) = 0$
$c_A^{\mathrm{p}} = c_B^{\mathrm{p}} = 0$
$c_A^{\mathrm{p}} = c_B^{\mathrm{p}} = 0$
L_x

(a) A pictorial description of the problem.

(b) Decomposition of the computational domain into subdomains.

Figure 3.21: Fast bimolecular reaction with advection: Chemical species A and B pumped into the reaction chamber from the left side and produce the product C as a result of the chemical reaction.

The above numerical examples clearly demonstrate that the proposed multi-time-step coupling methods can handle any decomposition of the computational domain: either the subdomains are contiguous or non-contiguous; whether the decomposition is based on the physics of the problem or based on numerical performance; or whether the decomposition is done manually by the user or obtained from a graph-partitioning software package.



(a) Concentration of the product C at $t = 0.5$.

(b) Concentration of the product C at $t = 1.5$.

(c) Concentration of the product C at $t = 4.0$.

Figure 3.22: Fast bimolecular reaction with advection: This figure shows the concentration of the product C at various instances of time obtained using the proposed d-continuity multi-time-step coupling method.

3.7 Concluding Remarks

We presented a stable multi-time-step computational framework for transient advective-diffusive-reactive systems. The computational domain can be divided into an arbitrary number of subdomains. Different time-stepping schemes under the trapezoidal family can be used in different subdomains. Different time-steps and different numerical formulations can be employed in different subdomains. Unlike many of the prior works on multi-time-step methods (e.g., staggered schemes proposed in [59]), no preferential treatment is given to the subdomain with the largest subdomain time-step, and thereby eliminating the associated subdomain-dependent solutions.

Under the framework, we proposed two different monolithic coupling methods, which differ in the way compatibility conditions are enforced along the subdomain interface. Under the first method (i.e., d-continuity method), the continuity of the primary variable is enforced along the subdomain interface at every system time-step. An attractive feature of the d-continuity method is that, by construction, there is no drift in the primary variable along the subdomain interface. However, one cannot couple explicit and implicit schemes under the d-continuity method. But this method has good stability characteristics. The second method is based on an extension of the classical Baumgarte stabilization [23,40] to first-order transient systems. Under this method one can couple explicit and implicit schemes. However, there can be drift in the primary variable along the subdomain interface. But this drift is bounded and small, which we have shown both theoretically and numerically. The other salient features of the proposed coupling methods are as follows: There is no limitation on the number of subdomains or on the subcycling ratios η_i. Since no preference is given to any subdomain, the numerical solutions under the proposed coupling methods will not be affected by the way the computational domain is decomposed into subdomains. This is also evident from the numerical results presented in this paper. The coupling methods are shown to be stable, which has been illustrated both mathematically and numerically.

Based on the above discussion, we shall make the following two recommendations for a multi-time-step analysis of first-order transient systems:

(i) If it is not needed to couple explicit/implicit time integrators, but one just wants to use different time-steps and different numerical formulations in different regions, then it is recommended to use the proposed *d*-continuity method. If one wants to couple explicit and implicit schemes, then one has to use the proposed coupling method based on Baumgarte stabilization.

(ii) Accuracy can be improved by decreasing the system time-step.

A possible research work can be towards the implementation of the proposed multi-time-step coupling methods in a parallel computing environment and on graphical processing units (GPUs).

Chapter 4

ON LATTICE BOLTZMANN METHOD FOR ADVECTION-DIFFUSION SYSTEMS

4.1 Introduction and Motivation

The lattice Boltzmann method (LBM) has gained remarkable popularity as a versatile numerical method for fluid dynamics simulations [74]. LBM has its roots in the kinetic theory as opposed to the continuum theory. It needs to be emphasized that LBM solves the Boltzmann equation instead of solving the continuum field equations. On the other hand, the finite element method (FEM) and the finite volume method (FVM) solve the continuum field equations directly. Some attractive features of LBM are: It can easily handle irregular domains (e.g., unstructured pores and fractures in porous media applications), easy to implement even for complicated flow models, and natural to parallelize [75]. Great advances have been made in extending LBM to simulate multi-phase flows [76], reactive flows [77], non-linear chemical reactions [78], just to name a few. In this chapter, we limit to LBM-based formulations for advection-diffusion equations.

In recent years, several key advancements have been made to extend the LBM to simulate transport phenomena. To name a few: [79–83]. Of these works, Yoshida and Nagaoka [82], and Huang and Wu [83] have proposed multiple-relaxation-time lattice Boltzmann methods to solve advection-diffusion equations with *anisotropic* diffusivity tensors.

The governing equations for transient advection-diffusion systems are parabolic partial differential equations, which possess several important mathematical properties. These properties include the maximum principle and the comparison principle [84,85], which have crucial implications in modeling physical phenomena. For example, a key consequence of the maximum principle in modeling advection-diffusion systems is the non-negative constraint of the attendant chemical species. Several factors such as the physical properties of the medium, topology of the domain, and the spatial and temporal discretization determine the performance of a numerical solution in preserving the *discrete* versions of the mentioned mathematical properties. A discussion on the influence of these factors in the context of the finite element method can be found in [86]. Violations of these mathematical properties can make a numerical solution inappropriate for scientific and engineering applications. It has been shown that many popular finite element and finite volume formulations for diffusion-type equations violate the maximum principle and the non-negative constraint [86, 87]. Recently, numerical methodologies have been proposed under the finite element method to satisfy the non-negative constraint and the maximum principle by utilizing the underlying variational structure. Since the lattice Boltzmann method does not enjoy such a variational basis, these methodologies developed for the FEM cannot be extended to the lattice Boltzmann method.

To the best of our knowledge, the performance of LBM-based formulations for unsteady diffusion and advection-diffusion equations with respect to comparison principles, maximum principles, and the non-negative constraint has not received the *attention it deserves. But such a study is of paramount importance, as LBM-based formulations are being employed in predictive numerical simulations.* We shall, therefore, put some popular LBM-based formulations to test, and particularly show that these formulations could violate all the aforementioned mathematical principles and physical constraints. The formulations of interest in this chapter are the non-thermal Single-Relaxation-Time (SRT) LBM for advection with isotropic diffusion, and the Multiple-Relaxation-Time (MRT) methods for anisotropic diffusion proposed in [82] and [83].

On the notational front, a quantity in the continuous setting will be denoted by upright font (e.g., u), and a quantity in the discrete setting will be denoted by italic font (e.g., u).

4.2 Unsteady Diffusion-Type Equations

Consider a bounded open domain Ω. We shall denote the boundary of the domain by $\Gamma = \overline{\Omega} - \Omega$, where $\overline{\Omega}$ is the set closure of Ω. We assume that the boundary Γ comprises of two parts $\Gamma^{\mathrm{N}}$and Γ^{D} such that $\Gamma^{\mathrm{N}} \cap \Gamma^{\mathrm{D}} = \varnothing$ and $\Gamma = \Gamma^{\mathrm{N}} \cup \Gamma^{\mathrm{D}}$. We denote the part of the boundary on which Dirichlet boundary condition is prescribed by Γ^{D}. Neumann boundary condition is prescribed on Γ^{N}. A spatial point is denoted by $\mathbf{x}$. The unit outward normal to the boundary is denoted by $\widehat{\mathbf{n}}(\mathbf{x})$. The time interval of interest will be denoted by $[0, \mathcal{T}]$, and the time is denoted by t. The divergence and gradient operators with respect to $\mathbf{x}$ are, respectively, denoted by $\mathrm{div}[\cdot]$ and $\mathrm{grad}[\cdot]$.

For convenience, we introduce notation

$$\mathbf{q}(\mathbf{x},\mathrm{t}) := \mathbf{v}(\mathbf{x},\mathrm{t})\mathrm{u}(\mathbf{x},\mathrm{t}) - \mathbf{D}(\mathbf{x})\mathrm{grad}[\mathrm{u}(\mathbf{x},\mathrm{t})], \text{ and} \tag{4.1}$$

$$\mathcal{L}[\mathrm{u}(\mathbf{x},\mathrm{t})] := \frac{\partial \mathrm{u}(\mathbf{x},\mathrm{t})}{\partial \mathrm{t}} + \mathrm{div}[\mathbf{q}(\mathbf{x},\mathrm{t})], \tag{4.2}$$

where $\mathrm{u}(\mathbf{x},\mathrm{t})$ is the concentration field. The diffusivity tensor is denoted by $\mathbf{D}(\mathbf{x})$, which is assumed to be symmetric, positive definite, and bounded above. The advection velocity is denoted by $\mathbf{v}(\mathbf{x},\mathrm{t})$, which is assumed to be solenoidal. The initial boundary value problem for a transient advection-diffusion system can be written as

$$\mathcal{L}[\mathrm{u}(\mathbf{x},\mathrm{t})] = \mathrm{g}(\mathbf{x},\mathrm{t}) \quad (\mathbf{x},\mathrm{t}) \in \Omega \times (0,\mathcal{T}], \tag{4.3a}$$

$$\mathrm{u}(\mathbf{x},\mathrm{t}) = \mathrm{u}^{\mathrm{p}}(\mathbf{x},\mathrm{t}) \quad (\mathbf{x},\mathrm{t}) \in \Gamma^{\mathrm{D}} \times [0,\mathcal{T}], \tag{4.3b}$$

$$\mathbf{q}(\mathbf{x},\mathrm{t}) \cdot \widehat{\mathbf{n}}(\mathbf{x}) = \mathrm{q}^{\mathrm{p}}(\mathbf{x},\mathrm{t}) \quad (\mathbf{x},\mathrm{t}) \in \Gamma^{\mathrm{N}} \times [0,\mathcal{T}], \text{ and} \tag{4.3c}$$

$$\mathrm{u}(\mathbf{x},\mathrm{t}=0) = \mathrm{u}_0(\mathbf{x}) \quad \mathbf{x} \in \Omega, \tag{4.3d}$$

where the source/sink is denoted by $\mathrm{g}(\mathbf{x},\mathrm{t})$, $\mathrm{u}^{\mathrm{p}}(\mathbf{x},\mathrm{t})$ is the prescribed concentration, $\mathrm{q}^{\mathrm{p}}(\mathbf{x},\mathrm{t})$ is the prescribed flux, and the initial concentration is denoted by $\mathrm{u}_0(\mathbf{x})$. Equation (4.3a) is a linear parabolic partial differential equation. The initial boundary value problem given by equations (4.3a)–(4.3d) satisfies several important mathematical properties and physical constraints.

Mathematical properties and physical constraints

Let $\mathsf{C}(\Omega \times (0,\mathcal{T}])$ be the set of all continuous functions on $\Omega \times (0,\mathcal{T}]$. We shall define $\mathsf{C}_1^2(\Omega \times (0,\mathcal{T}])$ to be the set of all functions in $\mathsf{C}(\Omega \times (0,\mathcal{T}])$ that have continuous first- and second-order spatial derivatives in Ω, and continuous first-order time derivative in $(0,\mathcal{T}]$.

[The maximum principle] Let $\mathrm{u}(\mathbf{x},\mathrm{t}) \in \mathsf{C}_1^2\left(\Omega \times (0,\mathcal{T}]\right) \cap \mathsf{C}\left(\overline{\Omega} \times [0,\mathcal{T}]\right)$ be a solution of the initial boundary value problem (4.3) with $\partial\Omega = \Gamma^{\mathrm{D}}$. If $\mathrm{g}(\mathbf{x},\mathrm{t}) \geq 0$ then

$$\min_{(\mathbf{x},\mathrm{t})\in\overline{\Omega}\times[0,\mathcal{T}]} \mathrm{u}(\mathbf{x},\mathrm{t}) = \min\left[\min_{(\mathbf{x},\mathrm{t})\in\Gamma\times[0,\mathcal{T}]} \mathrm{u}(\mathbf{x},\mathrm{t}),\ \min_{\mathbf{x}\in\Omega} \mathrm{u}_0(\mathbf{x})\right]\cdot$$

[The comparison principle] Let $\mathrm{u}_1(\mathbf{x},\mathrm{t})$ and $\mathrm{u}_2(\mathbf{x},\mathrm{t})$ belong to $\mathsf{C}_1^2\left(\Omega \times (0,\mathcal{T}]\right) \cap \mathsf{C}\left(\overline{\Omega} \times [0,\mathcal{T}]\right)$. If $\mathcal{L}[\mathrm{u}_1] \geq \mathcal{L}[\mathrm{u}_2]$ on $\Omega \times (0,\mathcal{T}]$, and $\mathrm{u}_1^{\mathrm{p}}(\mathbf{x},\mathrm{t}) \geq \mathrm{u}_2^{\mathrm{p}}(\mathbf{x},\mathrm{t})$ on $\Gamma \times [0,\mathcal{T}]$ then $\mathrm{u}_1(\mathbf{x},\mathrm{t}) \geq \mathrm{u}_2(\mathbf{x},\mathrm{t})$ on $\overline{\Omega} \times [0,\mathcal{T}]$. Mathematical proofs to the maximum principle and the comparison principle can be found in [45]. One can show that the non-negative constraint for the concentration can be obtained as a consequence of the maximum principle under certain assumptions on the input data.

[The non-negative constraint] If $\mathrm{g}(\mathbf{x},\mathrm{t}) \geq 0$ in Ω, $\mathrm{u}^{\mathrm{p}}(\mathbf{x},\mathrm{t}) \geq 0$ on Γ, and $\mathrm{u}_0(\mathbf{x}) \geq 0$ in Ω then $\mathrm{u}(\mathbf{x},\mathrm{t}) \geq 0$ $\forall \mathbf{x} \in \overline{\Omega}$ and $\forall \mathrm{t} \in [0,\mathcal{T}]$. The following integral will be used in the remainder of this chapter

$$\mathcal{J}_2(\mathrm{u};\Omega;\mathrm{t}) := \int_{\Omega} \mathrm{u}^2(\mathbf{x},\mathrm{t})\ \mathrm{d}\Omega\cdot \tag{4.4}$$

[The decay property] If $\mathbf{v}(\mathbf{x},t) = \mathbf{0}$, $\mathrm{u}^{\mathrm{p}}(\mathbf{x},\mathrm{t}) = 0$ on the entire Γ, and $\mathrm{g}(\mathbf{x},\mathrm{t}) = 0$ in Ω then

$$\frac{\mathrm{d}}{\mathrm{dt}}\mathcal{J}_2(\mathrm{u};\Omega;\mathrm{t}) \leq 0 \quad \forall \mathrm{t}\cdot \tag{4.5}$$

In the subsequent sections we will illustrate the performance of some popular LBM-based formulations with respect to the aforementioned mathematical properties in the discrete setting. We will also compare the performance of the lattice Boltzmann method with the finite element method in this regard.

4.3 The Lattice Boltzmann Method

The lattice Boltzmann method is a way to numerically solve the Boltzmann equation [88]

$$\partial \mathrm{f}/\partial \mathrm{t} + \mathbf{v}\cdot \mathrm{grad}[\mathrm{f}] = \left(\mathrm{f}^{\mathrm{eq.}} - \mathrm{f}\right)/\lambda, \tag{4.6}$$

where f is the distribution of particles in the phase space. The macroscopic velocity is $\mathbf{v}$. The distribution of particles at thermodynamic equilibrium is denoted by $\mathrm{f}^{\mathrm{eq.}}$ and the external force term is neglected for simplicity. Obviously, we have adopted the Bhatnagar-Gross-Krook (BGK) [89] model for the collision term on the right-hand-side of the equation (5.9). Typically, the velocities are discretized in the way of $DnQm$ lattice models, where n is the number of spatial dimensions and m is the number of discrete momenta. The lattice models of interest in this chapter are shown in Figure 4.1. These discrete velocity/momentum directions will be denoted by $\boldsymbol{e}_i,\ i = 1,\cdots,m$ and the weight associated to them is w_i, with $\sum_{i=1}^{m} w_i = 1$. The time derivative is then discretized using an explicit Euler method. The resulting lattice Boltzmann equation is

$$\begin{aligned} \left|f_i\right\rangle(\boldsymbol{x} + \boldsymbol{e}_i\Delta t, t + \Delta t) = \left|f_i\right\rangle(\boldsymbol{x},t) \\ + \boldsymbol{S}(\boldsymbol{x},t)\left(\left|f_i^{\mathrm{eq}}\right\rangle(\boldsymbol{x},t) - \left|f_i\right\rangle(\boldsymbol{x},t)\right), \end{aligned} \tag{4.7}$$

where f_i is the discrete distribution of particles with velocity $\boldsymbol{e}_i$ and $\boldsymbol{S}(\boldsymbol{x},t)$ is the $m\times m$ matrix of relaxation times. Also, we have used the notation $\big|\cdot\big\rangle$ to denote $m\times 1$ column vector of distributions. The lattice cell size is shown by Δx and time-step is Δt. In the case of Single-Relaxation-Time (SRT) LBM, a popular choice of $\boldsymbol{S}(\boldsymbol{x},t)$ is

$$\boldsymbol{S}(\boldsymbol{x},t) = \frac{1}{\tau}\mathbf{I}_m, \tag{4.8}$$

where $\mathbf{I}_m$ is the $m \times m$ identity matrix and $\tau = \mathrm{D}(\boldsymbol{x},t)/c_s^2\Delta t + 1/2$ is the relaxation time. The lattice sound velocity is shown by c_s. A choice of equilibrium distribution for advection diffusion equation can be

$$f_i^{\text{eq.}}(\boldsymbol{x},t) = w_i u(\boldsymbol{x},t)\left(1 + \frac{\boldsymbol{e}_i \cdot \mathbf{v}}{c_s^2}\right), \tag{4.9}$$

where $u(\boldsymbol{x},t)$ is the concentration at a lattice node and w_i is the weight associated with the the i-th distribution. However, SRT LBM does not provide an appropriate framework for advection with anisotropic diffusion equations. For such cases, MRT LBM creates a much more suitable structure. Some recent methods for LBM simulation of anisotropic diffusivity are [82, 83], which we will use in the rest of the chapter. For an in-depth treatment and derivation of Equation (5.12) the reader should consult references [75, 90, 91] and references therein. The lattice Boltzmann method can be shown to have a corresponding macroscopic equation as Equation (4.3) using a Chapman-Enskog expansion or asymptotic analysis. The macroscopic quantity of interest in this chapter, concentration, can be found as

$$u(\boldsymbol{x},t) = \sum_{i=1}^{m} f_i(\boldsymbol{x},t)\cdot \tag{4.10}$$

One of the main issues in numerical simulations using LBM is the implementation of boundary conditions. Since the discrete distributions f_i have more information about the state of matter at each lattice node, translating the macroscopic boundary and initial conditions of Equation (4.3) to mesoscale is not unique. Hence, there are a multitude of methods for implementing boundary conditions in LBM; e.g. see References [92–94]. Herein, we will consider the following methods for enforcing Dirichlet and Neumann boundary conditions:

(a) *Weighted splitting method for Dirichlet conditions.* In this method, we will replace all the distributions f_i at the lattice node lying near the boundary Γ^{D}. The distributions will be readjusted according to the weights w_i. That is

$$f_i(\boldsymbol{x}, t) = w_i \mathrm{u}^{\mathrm{p}}(\boldsymbol{x}, t) \quad \boldsymbol{x} \in \Gamma^{\mathrm{D}}, \tag{4.11}$$

where u^{p} is the prescribed concentration on the boundary.

(b) *Local method for Dirichlet conditions.* In this method only the unknown distributions will be replaced. Let $\mathfrak{U}_{\mathrm{D}}$ be the set of *unknown* distributions and $\mathfrak{K}_{\mathrm{D}}$ be the set of known distributions at a point $\boldsymbol{x}$ on boundary Γ^{D}. Then, the unknown distributions will be found as

$$f_i(\boldsymbol{x}, t) = \frac{w_i}{\sum_{j \in \mathfrak{U}_{\mathrm{D}}} w_j} \left(\mathrm{u}^{\mathrm{p}}(\boldsymbol{x}, t) - \sum_{k \in \mathfrak{K}_{\mathrm{D}}} f_k(\boldsymbol{x}, t) \right). \tag{4.12}$$

(c) *Bounce-back method for Neumann conditions.* The unknown distributions will be replaced by the distribution in the opposite velocity direction. Mathematically, this can be written as

$$f_i(\boldsymbol{x}, t) = f_j(\boldsymbol{x}, t) - \frac{1}{c_s} \mathrm{q}^{\mathrm{p}}(\boldsymbol{x}, t) \frac{\widehat{\mathbf{n}} \cdot \boldsymbol{e}_i}{\sum_{k \in \mathfrak{U}_{\mathrm{N}}} \widehat{\mathbf{n}} \cdot \boldsymbol{e}_k}, \tag{4.13}$$

where $\boldsymbol{e}_i = -\boldsymbol{e}_j$, $\boldsymbol{x} \in \Gamma^{\mathrm{N}}$ and $\mathfrak{U}_{\mathrm{N}}$ is the set of unknown distributions.

This concludes our overview of lattice Boltzman methods for advection-diffusion equations. In the following section we will provide ample evidence that current lattice Boltzmann methods for advection-diffusion equation may lead to violation of mathematical properties of these equations.

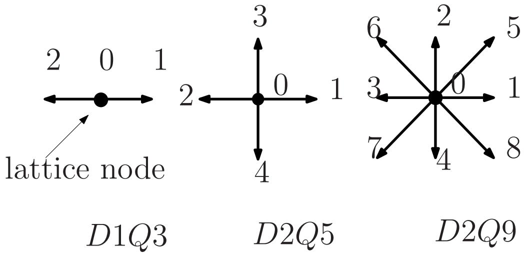


Figure 4.1: Lattice Boltzmann method: Lattice models employed in this chapter.

4.4 Representative Numerical Results

In this section, we employ the single- and multiple-relaxation-time lattice Boltzmann methods to solve representative diffusion and advection-diffusion problems. For brevity, we shall refer to the multiple-relaxation-time method proposed in [82] as the Y-N method, and to the multiple-relaxation-time method proposed in [83] as the H-W method. In the rest of this section, we shall use the notations

$$u_{\min}(t) = \min_{\boldsymbol{x} \in \overline{\Omega}} \; u(\boldsymbol{x}, t), \text{ and } \quad u_{\max}(t) = \max_{\boldsymbol{x} \in \overline{\Omega}} \; u(\boldsymbol{x}, t) \cdot \tag{4.14}$$

In all the problems that utilize MRT LBM the distributions f_i are initialized according to the Maxwell-Boltzmann equlibrium distribution. Note that all the quantities presented hereon are nondimensional.

One-Dimensional Problems

Advection and diffusion of Gaussian hill

Consider the domain $\Omega = (-2, 2)$, in which the diffusion coefficient is $\mathrm{D} = 10^{-2}$ and advection velocity is $\mathrm{v} = -1$. The source term is taken to be zero over the entire domain. The flux is set to be zero at the boundaries of the domain. The initial concentration is taken to be

$$\mathrm{u}_0(\mathrm{x}) = e^{-(\mathrm{x}-\mathrm{x}_0)^2/\alpha^2}, \tag{4.15}$$

where $\mathrm{x}_0 = 3/4$ and $\alpha^2 = 10^{-3}$ in this problem. We will use a uniform lattice with $D1Q2$ model. SRT LBM will be used to find the numerical solution. The lattice cell size is denoted by Δx and the time-step is Δt. The time-step is chosen as $\Delta t = \Delta x^2/6\mathrm{D}$ in all cases to maintain stability. The time interval of interest is $\mathcal{T} = 1/2$. Bounce-back conditions is imposed on the boundaries to satisfy the zero flux condition. The initial distributions is assigned to be equal to equilibrium distribution (i.e., $f_i(x, t = 0) = f_i^{\mathrm{eq.}}(x, t = 0),\ \forall x \in \Omega$). Since the initial concentration is non-negative throughout the domain, the non-negative constraint implies that the value of concentrations should never be negative in the course of the simulation. Herein, we will show that despite stability and convergence, the numerical solution from LBM may violate the non-negative constraint. We shall use different values of discretization parameters Δx and Δt as given in Table 4.1. The minimum observed concentration $u_{\min}(t)$ and the error in infinite norm $\mathcal{E}_\infty(t)$ are also shown in Table

Table 4.1: One-dimensional problem: Discretization of time and space domains for the one-dimensional problem. Minimum observed value for concentration, $u_{\min}(t)$ and the error in infinity norm are also provided. Violation of non-negative constraint is obvious.

Case	Δx	Δt	$u_{\min}(\mathcal{T})$	$\mathcal{E}_{\infty}(\mathcal{T})$
1	3.33×10^{-2}	1.85×10^{-2}	-3.90×10^{-3}	6.50×10^{-2}
2	2.50×10^{-2}	1.04×10^{-2}	-3.00×10^{-5}	4.06×10^{-2}
3	2.00×10^{-2}	6.70×10^{-3}	0.0	2.05×10^{-2}
4	1.25×10^{-2}	2.60×10^{-3}	0.0	7.67×10^{-3}

4.1. The error is defined as

$$\mathcal{E}_{\infty}(t) = \max_{x \in \Omega} \left| u(x,t) - u_{exact}(x,t) \right| \cdot \tag{4.16}$$

Figures 4.2-4.3 show the numerical results. The violation of the non-negative constraint can be observed. Note that by refining the discretization parameters Δx and Δt, the violations of the non-negative constraint disappear. However, satisfaction of the non-negative constraint is not inherent in the LBM in use for this problem and should not be taken for granted. A computer code for this problem is provided in the Appendix.

On comparison principle

As stated earlier in this chapter, an important mathematical property of advection-diffusion equation is the comparison principle. In the case of linear operator $\mathcal{L}[\cdot]$ in equation (4.2), maximum principle and comparison principle are mathematically equivalent. However, a numerical solution might violate one of the two principles and not the other. Herein, we provide simple numerical examples to demonstrate this issue.

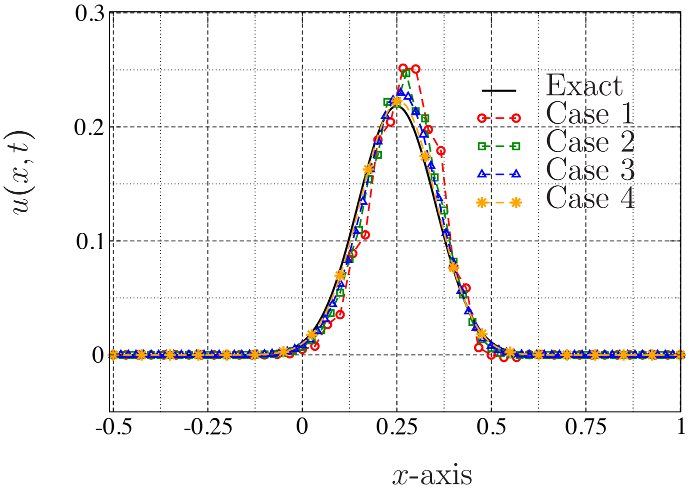


Figure 4.2: Advection and diffusion of Gaussian hill: The concentration at time $t = \mathcal{T}$ is shown. Note the negative values of concentration for Cases 1 and 2.

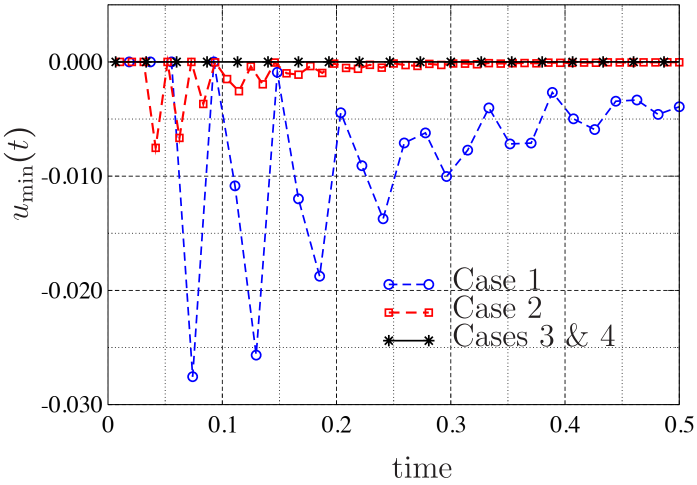


Figure 4.3: Advection and diffusion of Gaussian hill: Minimum observed concentration is shown against time.

Consider the domain $\Omega = (0, 1]$ with prescribed concentration on the boundary:

$$\mathrm{u}^{\mathrm{p}}(\mathrm{x}, \mathrm{t}) = \begin{cases} \mathrm{u}_L & \mathrm{x} = 0 \\ 0 & \mathrm{x} = 1 \end{cases} . \tag{4.17}$$

The initial concentration is set to be $\mathrm{u}_0(\mathrm{x}) = 0, \ \mathrm{x} \in \Omega$. The diffusion coefficient and the advection velocity are $\mathrm{D} = 10^{-3}$ and $\mathrm{v} = 1/2$ respectively. We use the $D1Q2$ lattice model and the time-step is $\Delta t = \Delta x^2 / 6\mathrm{D}$. Figure 4.4 shows the numerical result for

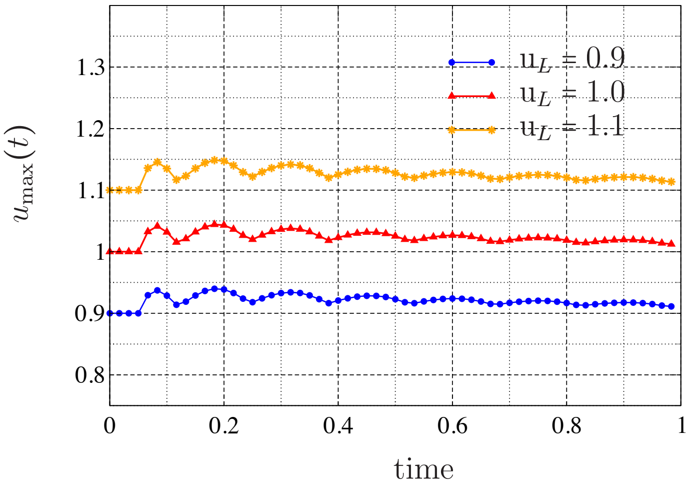


Figure 4.4: On comparison principle: This figure shows the maximum concentration observed against time. In all cases the maximum principle is violated.

different values of u_L. The lattice cell size is taken to be $\Delta x = 10^{-2}$. Obviously, the maximum principle is violated, since the values of concentration exceeds u_L. However, the comparison principle is not violated (the numerical solution in the entire domain never exceeds the one with larger u_L). Hence, one can conclude that in a discrete setting comparison principle and maximum principle should be accounted for separately.

On choice of lattice model: D1Q2 vs. D1Q3

Consider the domain $\Omega = (0, 1)$ in which the diffusion coefficient is $D = 10^{-3}$ and the advection velocity is $v = 1$. We prescribe zero-flux boundary conditions on the entire boundary. The initial condition is

$$u_0(x) = \begin{cases} 1 & x \in [0.3, 0.5] \\ 0 & \text{otherwise} \end{cases} \tag{4.18}$$

The time interval of interest is $\mathcal{T} = 10^{-1}$. Here, we shall compare the performance of two different choices of lattice models with respect to preservation of maximum principle. The lattice models of interest are $D1Q2$ and $D1Q3$ lattice models. The discrete velocities and their associated weights are

$$D1Q2: \begin{cases} \boldsymbol{e} = [1,\ -1] \\ \boldsymbol{w} = [1/2,\ 1/2] \end{cases} \quad \text{and} \tag{4.19a}$$

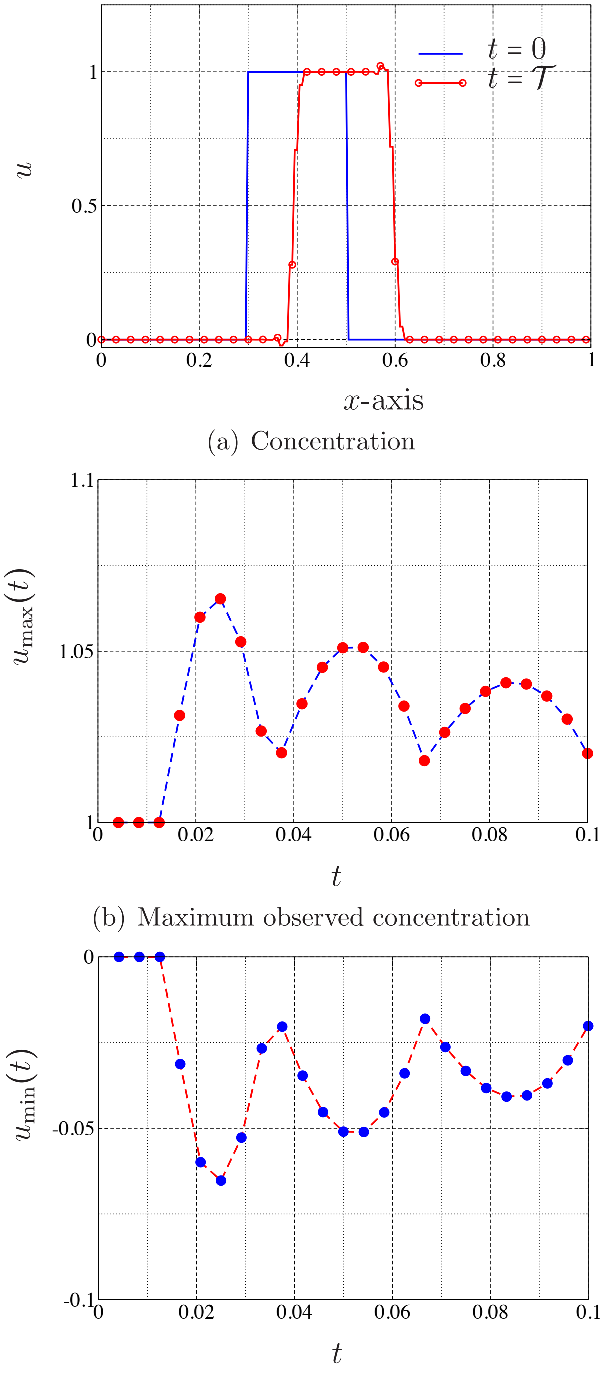
$$D1Q3: \begin{cases} \boldsymbol{e} = [0,\ 1,\ -1] \\ \boldsymbol{w} = [4/6,\ 1/6,\ 1/6] \end{cases} \quad . \tag{4.19b}$$

The lattice cell size is Δx and the time-step is Δt. The lattice sound velocity for the $D1Q2$ lattice is $c_s = \Delta x/\Delta t$ and for the $D1Q3$ this velocity is $c_s = \Delta x/\sqrt{3}\Delta t$. The distributions will be initialized according to equilibrium distribution.

Using a lattice cell size of $\Delta x = 5 \times 10^{-3}$ and a time-step of $\Delta t = 4.2 \times 10^{-3}$, we obtain the numerical results presented in Figures 4.5-4.6. According to the maximum principle, the concentration should remain in $[0, 1]$. However, It can be observed that this requirement is violated by both lattice models. Note that in this numerical experiment, concentration exceeds unity and also adopts negative values. Note that the $D1Q2$ and $D1Q3$ give different patterns in $u_{\min}(t)$ and $u_{\max}(t)$. Obviously, the maximum and minimum values of concentrator form the $D1Q3$ lattice oscillate more compared to the $D1Q2$ lattice. However, the magnitude of violations is comparable. Similar to the previous cases, these violations can be removed by refining the lattice cell size and the time-step according to the stability criteria.

Two-Dimensional Problem with Isotropic Diffusion

Consider the domain $\Omega = (0, 2) \times (0, 1)$. We will take the diffusion coefficient as $\mathrm{D} = 5 \times 10^{-3}$ and the advection velocity to be $\mathrm{v_x} = 1$ and $\mathrm{v_y} = 0$. The initial

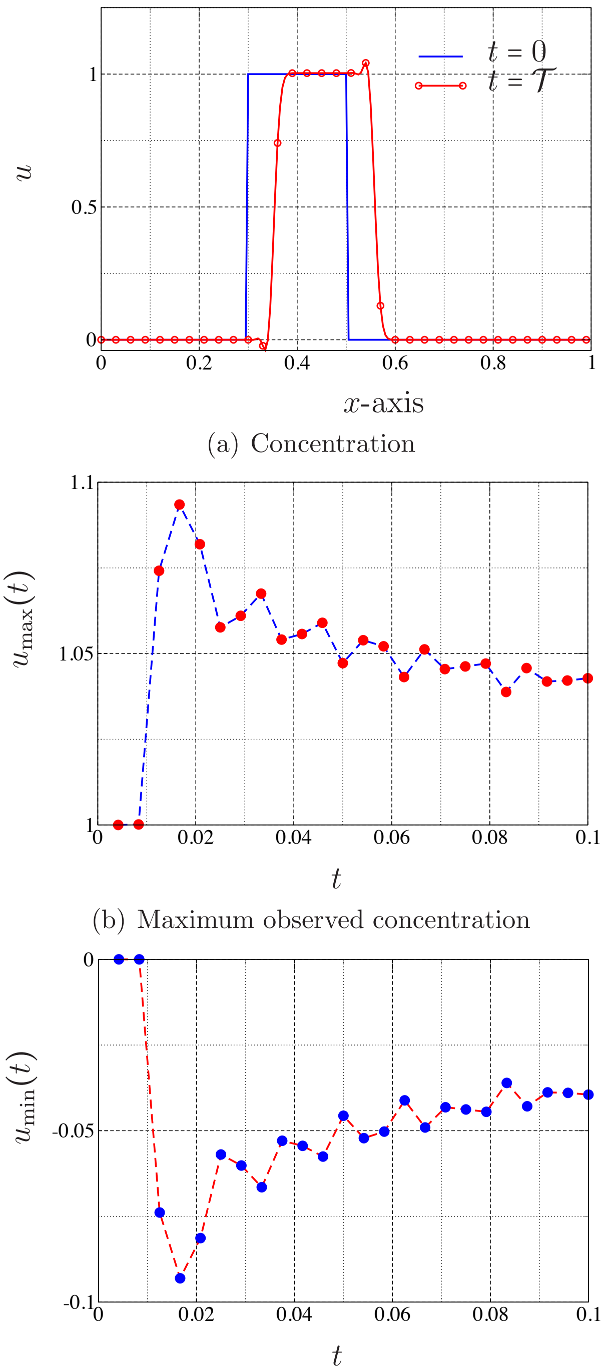


(a) Concentration

(b) Maximum observed concentration

(c) Minimum observed concentration

Figure 4.5: On choice of lattice model: In this figure, the numerical results using the $D1Q2$ lattice are presented. This figure shows that the maximum principle is violated.



(a) Concentration

(b) Maximum observed concentration

(c) Minimum observed concentration

Figure 4.6: On choice of lattice model: In this figure, the numerical results using the $D1Q3$ lattice are presented. The maximum principle is violated.

concentration is zero. The boundary conditions are as

$$\mathrm{u^p(x,y,t)} = \begin{cases} 1 & \mathrm{x = 0,\ \forall y,\ \forall t} \\ 0 & \mathrm{x = 2,\ \forall y,\ \forall t} \end{cases} \quad \text{and} \tag{4.20a}$$

$$\mathrm{q^p(x,y,t)} = 0 \quad \mathrm{y \in \{0,1\},\ \forall x,\ \forall t} \cdot \tag{4.20b}$$

The source term is set to be zero throughout the domain. We shall employ SRT LBM with a $D2Q4$ lattice model for numerical solution. The local method for enforcing the Dirichlet boundary condition at $\mathrm{x} = 0$ will be used (see equation (4.12)). The Dirichlet conditions at $\mathrm{x} = 2$ are enforced by the weighted splitting method (see Equation (4.11)). Also, bounce-back conditions are posed to enforce zero flux boundary conditions (see equation (4.13)). The initial distributions are assigned to be the same as equilibrium distributions using the given initial concentration.

Figure 4.7 shows the concentration at $t = 1$. In this case, the cell size is $\Delta x = 3.33\times10^{-2}$ and time-step is taken to be $\Delta t = 1.85\times10^{-2}$. The maximum concentration in this case is $u_{\max}(t = 1) = 1.0139$, which is greater than the maximum possible value for concentration in this problem. Hence, the maximum principle is violated. These violations, however, can be removed by refining the cell size and the time-step according to a conventional Courant-Friedrichs-Lewy (CFL) [95] condition.

If we change the Dirichlet boundary conditions as

$$\mathrm{u^p(x,y,t)} = \begin{cases} 0 & \mathrm{x = 0,\ \forall y,\ \forall t} \\ 1 & \mathrm{x = 2,\ \forall y,\ \forall t} \end{cases} . \tag{4.21}$$

The spurious oscillations in the concentration profile lead to negative values. This result is shown in Figure 4.8. The values of Δx and Δt are the same as before. The

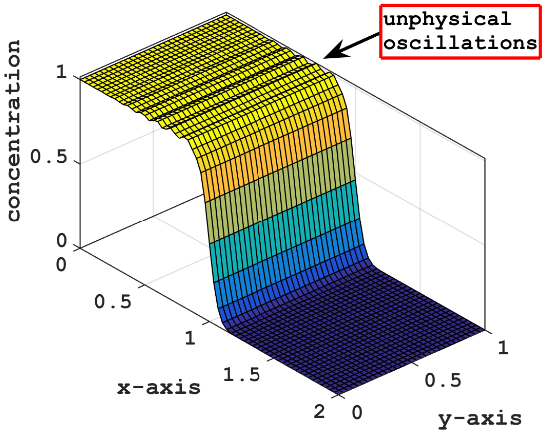


Figure 4.7: Two-dimensional problem with isotropic diffusion: This figure shows the unphysical node-to-node oscillations in the concentration. The maximum observed concentration in this figure is $u_{\max}(t = 1) = 1.0139$.

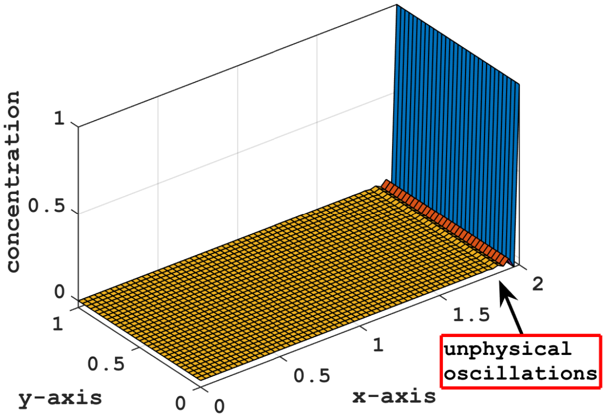


Figure 4.8: Two-dimensional problem with isotropic diffusion: This figure shows unphysical node-to-node oscillations in the concentration from the $D2Q4$ LBM. In this case, concentration adopts negative values. The minimum observed in this case is $u_{\min}(t = 1) = -0.0384$.

minimum observed value for concentration is $u_{\min}(t = 1) = -0.0382$. Similar to the previous case, these oscillations can be removed by refining the lattice cell size and time-step.

C. Two-dimensional problem with anisotropic diffusion on a non-convex domain

We now examine the Y-N multiple-relaxation-time method for anisotropic diffusion tensor. The computational domain is shown in Figure 4.9. We have taken $L = 1$ and $\Gamma^{\mathrm{D}} = \Gamma_{\mathrm{outer}} \cup \Gamma_{\mathrm{inner}}$. On the inner boundary, the concentration is prescribed to be unity (i.e., $\mathrm{u}^{\mathrm{p}}(\mathbf{x}, \mathrm{t}) = 1$ for $\mathbf{x} \in \Gamma_{\mathrm{inner}}$). The flux is prescribed to be zero on the outer boundary (i.e., $\mathrm{q}^{\mathrm{p}}(\mathbf{x}, \mathrm{t}) = 0$ for $\mathbf{x} \in \Gamma_{\mathrm{outer}}$). The anisotropic diffusion tensor is taken as

$$\mathbf{D}(\mathbf{x}) = \mathbf{R}_{\theta}^{\mathrm{T}} \mathbf{D}_0 \mathbf{R}_{\theta}, \tag{4.22}$$

where

$$\mathbf{D}_0 = \begin{bmatrix} 10 & 0 \\ 0 & 10^{-3} \end{bmatrix}. \tag{4.23}$$

The orthogonal rotation matrix is denoted by $\mathbf{R}_{\theta}$, where θ denotes the angle of rotation. Herein, we have taken $\theta = \pi/4$. We employ the $D2Q5$ lattice model. The discrete velocity directions are given by

$$\boldsymbol{e}_i^{\mathrm{T}} = \begin{cases} [0, 0] & i = 0 \\ c\left[\cos\left((i-1)\pi\right), \sin\left((i-1)\pi\right)\right] & i = 1, 2 \\ c\left[\cos\left((2i-5)\pi/2\right), \sin\left((2i-5)\pi/2\right)\right] & i = 3, 4 \end{cases}. \tag{4.24}$$

where $c = \Delta x/\Delta t$. The respective weights are taken as

$$w_i = \begin{cases} 1/3 & i = 0 \\ 1/6 & i = 1, 2, 3, 4 \end{cases}. \tag{4.25}$$

The time interval of interest is taken as $\mathcal{T} = 10^{-2}$. We employ the local method of enforcing Dirichlet boundary conditions (see equation (4.12)). Table 4.2 provides the discretization parameters employed in this chapter. Figures 4.10–4.11 show that the Y-N method violates the non-negative constraint. In fact, the obtained minimum concentration is about −0.4, which is a significant violation given the fact that the concentration should be between 0 and 1. Another noticeable feature in all the cases considered, the minimum concentration converged to a negative value as the time progressed.

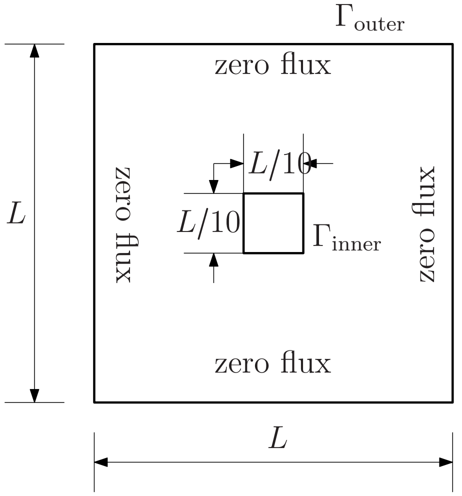
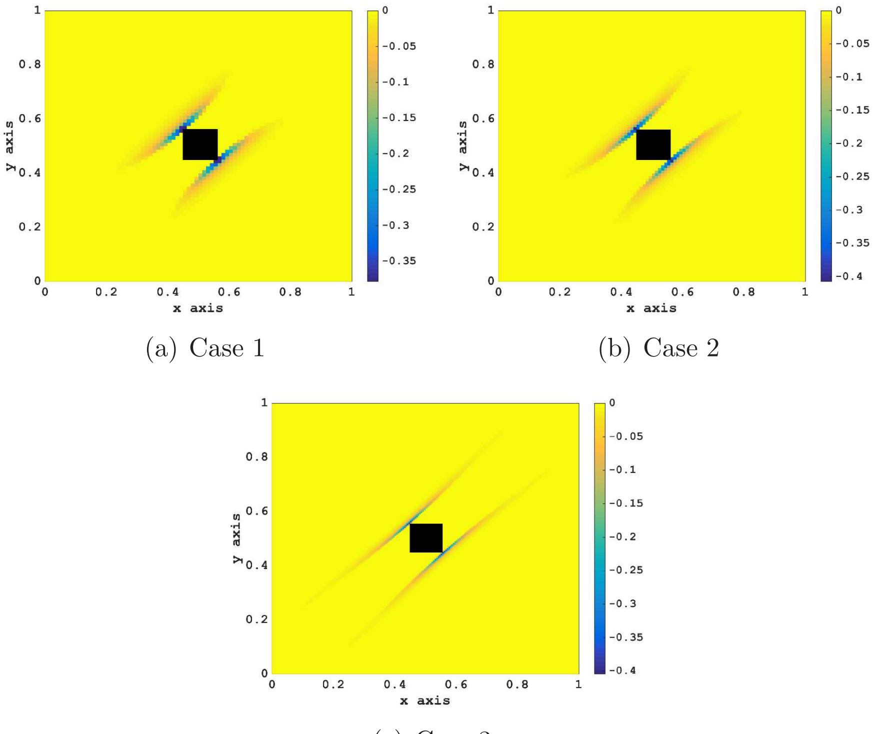


Figure 4.9: Two-dimensional problem with anisotropic diffusion in a non-convex domain: This figure provides a pictorial description of the test problem. A concentration of $\mathrm{u}^{\mathrm{p}} = 1$ is prescribed on the inner boundary Γ_{inner}.



(a) Case 1

(b) Case 2

(c) Case 3

Figure 4.10: Two-dimensional problem with anisotropic diffusion in a non-convex domain: The figure shows the regions where the non-negative constraint is violated under the Y-N method at $t = 0.01$.

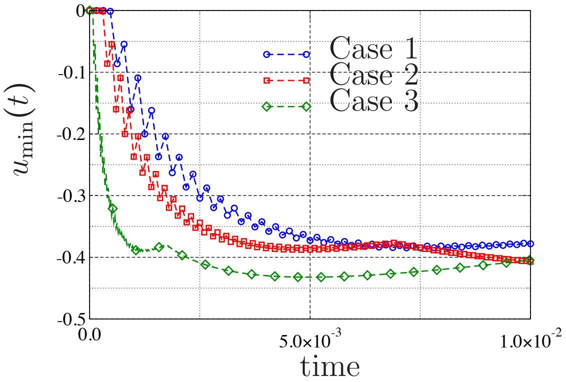


Figure 4.11: Two-dimensional problem with anisotropic diffusion in a non-convex domain: The figure shows the variation of minimum value of the concentration with respect to time under the Y-N multiple-relaxation-time lattice Boltzmann method.

Table 4.2: Two-dimensional problem with anisotropic diffusion tensor on a non-convex domain: This table provides the minimum concentrations for various discretization parameters (i.e., Δx and Δt). We have taken $\Delta x^2 = \Delta t$.

Case	Δx	Δt	$u_{\min}(\mathcal{T})$
1	1.25×10^{-2}	1.5625×10^{-4}	-0.3781
2	1.00×10^{-2}	1.0000×10^{-4}	-0.4072
3	5.00×10^{-3}	2.5000×10^{-5}	-0.4044

D. Two-dimensional problem with anisotropic and heterogeneous diffusion tensor

Consider the spatial domain to be $\Omega = (0,1) \times (0,1)$. We have taken the following anisotropic and heterogeneous diffusivity tensor as

$$\mathbf{D}(\mathrm{x},\mathrm{y}) = \epsilon' \mathbf{I}_2 + \begin{bmatrix} \epsilon \mathrm{x}^2 + \mathrm{y}^2 & -(1-\epsilon)\,\mathrm{xy} \\ -(1-\epsilon)\,\mathrm{xy} & \mathrm{x}^2 + \epsilon \mathrm{y}^2 \end{bmatrix}. \tag{4.26}$$

where $\epsilon \ll 1$ and $\epsilon' \ll 1$ are arbitrary constants, and $\mathbf{I}_2$ denotes the 2×2 identity matrix. For this numerical experiment, we have taken $\epsilon = 10^{-3}$ and $\epsilon' = 10^{-10}$. The prescribed concentration on the entire boundary is taken to be zero. The initial concentration is taken as

$$\mathrm{u}_0(\mathrm{x},\mathrm{y}) = \begin{cases} 1 & (\mathrm{x},\mathrm{y}) \in [0.4, 0.6] \times [0.4, 0.6] \\ 0 & \text{otherwise} \end{cases}. \tag{4.27}$$

The time interval of interest is taken as $\mathcal{T} = 2.5 \times 10^{-2}$. The H-W method based on the *D2Q9* lattice model is employed. The discrete velocities are taken as

$$\boldsymbol{e}_i^{\mathrm{T}} = \begin{cases} [0,0] & i = 0 \\ c\left[\cos\left((i-1)\pi/2\right), \sin\left((i-1)\pi/2\right)\right] & i = 1, \ldots, 4 \\ \sqrt{2}c\left[\cos\left((2i-9)\pi/4\right), \sin\left((2i-9)\pi/4\right)\right] & i = 5, \ldots, 8 \end{cases}$$

with the following weights

$$w_i = \begin{cases} 4/9 & i = 0 \\ 1/9 & i = 1, 2, 3, 4 \\ 1/36 & i = 5, 6, 7, 8 \end{cases} . \tag{4.28}$$

The problem is solved using different choices of Δx and Δt, which are provided in Table 4.3. This table also provides insight on the performance of the H-W method. The following conclusions can be drawn from Figures 4.13–4.14 and Table 4.3:

(a) The H-W multiple-relaxation-time method violates the non-negative constraint when the diffusion is anisotropic.

(b) As discussed earlier, the integral $\mathcal{J}_2$ should decrease monotonically with time for pure diffusion equations. But, the H-W method does not respect the decay property. However, it has been observed that refining the discretization parameters (i.e., Δx and Δt) can improve the performance of numerical solutions with respect to the decay property.

Table 4.3: Two-dimensional problem with anisotropic and heterogeneous diffusion tensor: In this table the values of discretization parameters for different cases is given. The nonnegative constraint is violated in all cases. Refining discretization parameters (Δt and Δx) does not remove the violations of the non-negative constraint.

Case	Δx	Δt	$u_{\min}(\mathcal{T})$	$u_{\max}(\mathcal{T})$
1	5.00×10^{-2}	1.00×10^{-3}	-0.0133	0.5167
2	2.50×10^{-2}	2.50×10^{-4}	-0.0077	0.5616
3	1.00×10^{-2}	4.00×10^{-5}	-0.0020	0.5977
4	5.00×10^{-3}	1.00×10^{-5}	-0.0001	0.5903

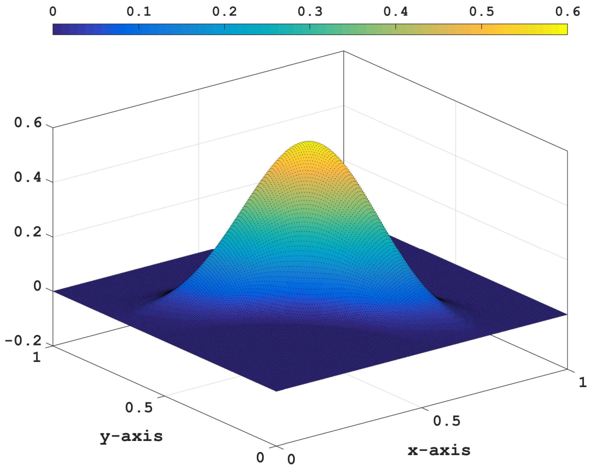
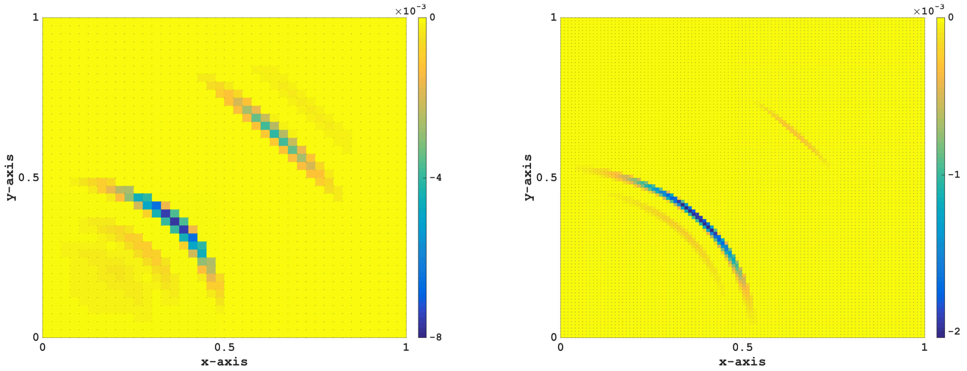


Figure 4.12: Two-dimensional problem with anisotropic and heterogeneous diffusion tensor: The concentration at time $t = 0.025$ is shown. The values of discretization parameters are given in Table 4.3, Case 4.



(a) Case 2

(b) Case 3

Figure 4.13: Two-dimensional problem with anisotropic and heterogeneous diffusion tensor: This figure shows the regions that have negative values for the concentration under the H-W method at $t = 0.025$.

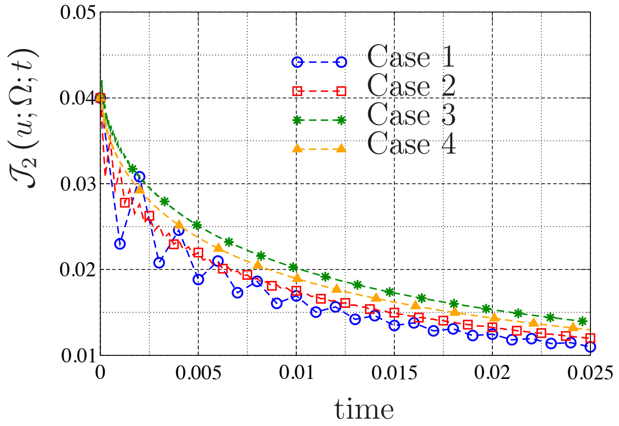


Figure 4.14: Two-dimensional problem with anisotropic and heterogeneous diffusion tensor: This figure shows the variation of the integral defined in equation (4.4) with respect to time under the H-W method.

E. Fast bimolecular reaction in an anisotropic and heterogeneous medium

Consider a simple chemical reaction of the form:

$$n_A A + n_B B \to n_C C, \tag{4.29}$$

where A, B and C are the participating chemical species; and n_A, n_B and n_C are their respective stoichiometry coefficients. We are interested in the fate of the product C when the time-scale of the chemical reaction is much faster than that of the transport processes (i.e., diffusion and advection). A detailed description of this mathematical model can be found in [72] and will not be repeated here. However, the mentioned chapter neglected advection in all their numerical examples and the entire chapter is devoted to the finite element method.

We will consider the domain given in Figure 4.15. Dimensions of the domain are

Table 4.4: Fast bimolecular reaction in anisotropic and heterogeneous medium: Different discretization parameters and violation of the non-negative constraint.

Case	Δx	Δt	$u_{\mathrm{min}}\left(\mathcal{T}\right)$	$u_{\mathrm{max}}\left(\mathcal{T}\right)$
1	5.00×10^{-2}	2.50×10^{-4}	-0.0408	0.2612
2	2.50×10^{-2}	6.25×10^{-5}	-0.0455	0.3174
3	1.25×10^{-2}	1.56×10^{-6}	-0.0593	0.3217

$L_{\mathrm{x}} = 2$ and $L_{\mathrm{y}} = 1$. The advection velocity is derived from the stream function

$$\psi\left(\mathrm{x},\mathrm{y}\right) = -\mathrm{y} - \sum_{k=1}^{3} \alpha_k \cos\left(\frac{p_k \pi \mathrm{x}}{L_{\mathrm{x}}} - \frac{\pi}{2}\right) \sin\left(\frac{q_k \pi \mathrm{y}}{L_{\mathrm{y}}}\right),$$

where $(p_1, p_2, p_3) = (4, 5, 10)$, $(q_1, q_2, q_3) = (1, 5, 10)$, and $(\alpha_1, \alpha_2, \alpha_3) = (0.08, 0.02, 0.01)$. The dispersion tensor is taken as

$$\mathbf{D}\left(\mathrm{x},\mathrm{y}\right) = 10^{-5}\mathbf{I} + \beta_{\mathrm{T}} \left\|\mathbf{v}\right\| \mathbf{I} + \left(\beta_{\mathrm{L}} - \beta_{\mathrm{T}}\right) \frac{\mathbf{v} \otimes \mathbf{v}}{\left\|\mathbf{v}\right\|}, \tag{4.30}$$

where ⊗ denotes the tensor product, $\mathbf{I}$ is the 2×2 identity tensor, $\|\cdot\|$ is the 2-norm, $\beta_{\mathrm{T}} = 10^{-4}$ and $\beta_{\mathrm{L}} = 1$. The prescribed concentrations are $\mathrm{u}_A^{\mathrm{p}} = \mathrm{u}_B^{\mathrm{p}} = 1$, and the stoichiometry coefficients are $n_A = 1$, $n_B = 2$ and $n_C = 1$. The time interval of interest is $\mathcal{T} = 2.5 \times 10^{-1}$. We employ the *D2Q9* lattice model using the H-W method, and obtain the numerical solution for the fate of the product C. Discretization parameters for various cases are given in Table 4.4. *Figures 4.16–4.18 clearly show that the H-W method violates the non-negative constraint, and the violations do not vanish either with time or with refinement of the discretization parameters.*

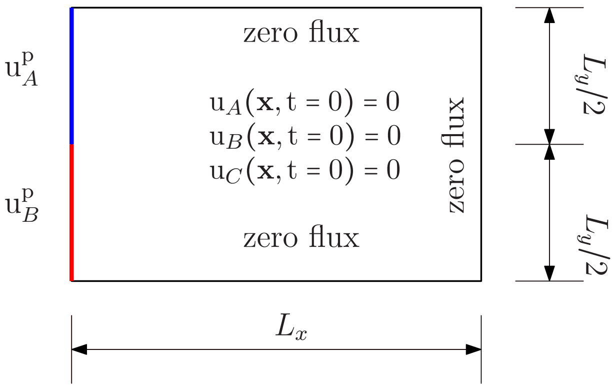

zero flux
u_A^p
u_B^p
$u_A(\mathbf{x}, t = 0) = 0$
$u_B(\mathbf{x}, t = 0) = 0$
$u_C(\mathbf{x}, t = 0) = 0$
zero flux
zero flux
$L_y/2$
$L_y/2$
L_x

Figure 4.15: Fast bimolecular reaction in anisotropic and heterogeneous medium: This figure provides a pictorial description of the test problem. The reactants A and B undergo transport (i.e., both advection and diffusion) and reacts to give product C, which in turn gets transported. We have taken $L_x = 2$ and $L_y = 1$ in the numerical experiment.

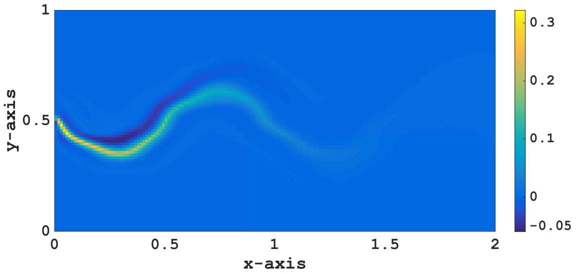


Figure 4.16: Fast bimolecular reaction in anisotropic and heterogeneous medium: The concentration of the chemical species C at time $t = 0.1$ is shown. Violation of the nonnegative constraint is obvious.

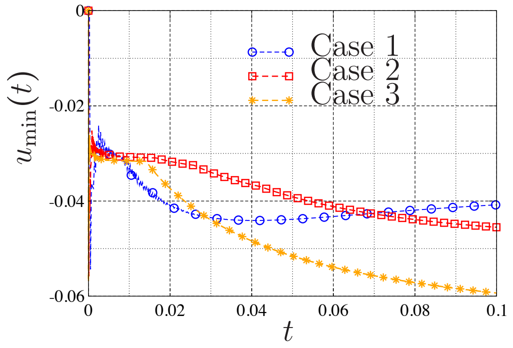
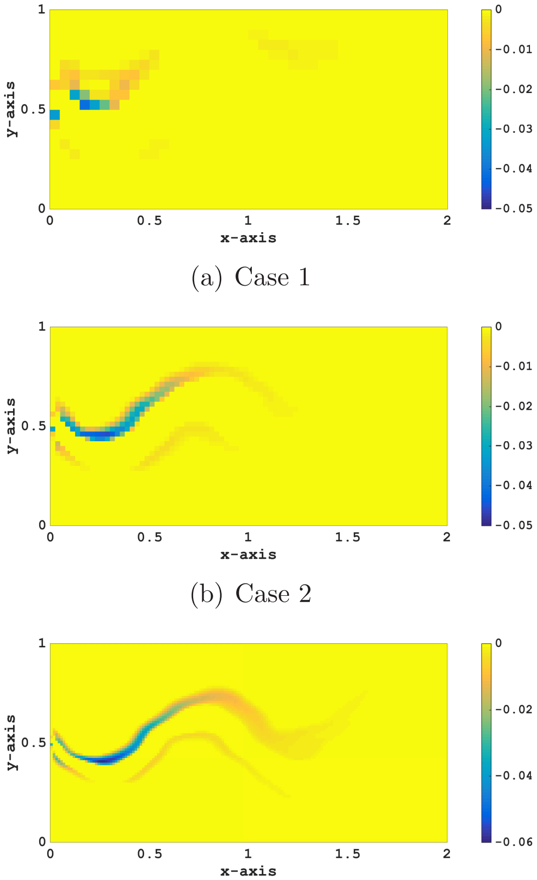


Figure 4.18: Fast bimolecular reaction in anisotropic and heterogeneous medium: The minimum concentration of the product C is plotted against time for various cases whose simulation parameters are provided in Table 4.4.



(a) Case 1

(b) Case 2

(c) Case 3

Figure 4.17: Fast bimolecular reaction in anisotropic and heterogeneous medium: This figure shows the regions where the concentration of the product C is negative at $t = \mathcal{T} = 0.1$ under the H-W method.

4.5 A Theoretical Analysis

In this section, we will provide a simple criterion in terms of the discretization parameters to satisfy the non-negative constraint for one-dimensional problems. We will also limit our scope to pure diffusion equations (i.e., $\mathbf{v}(\mathbf{x}, \mathrm{t}) = \mathbf{0}$), and $\partial\Omega = \Gamma^{\mathrm{D}}$. We will restrict the analysis to the $D1Q3$ lattice model. We initialize the discrete distributions f_i at all lattice nodes as follows:

$$f_i(x, t = 0) = w_i \mathrm{u}_0(x) \cdot \tag{4.31}$$

Since $w_i > 0$ and $\mathrm{u}_0(x) \geq 0$, we have $f_i(x, 0) \geq 0$. Furthermore, we assume that the Dirichlet boundary conditions will be discretized using the *weighted splitting method* (see equation (4.11)). The weighted splitting method guarantees the non-negativity of distributions f_i for a lattice node on the boundary provided that the prescribed concentration on Γ^{D} is non-negative. That is,

$$\mathrm{u}^{\mathrm{p}}(x \in \Gamma^{\mathrm{D}}, t) \geq 0 \Rightarrow f_i(x \in \Gamma^{\mathrm{D}}, t) \geq 0 \cdot \tag{4.32}$$

So far, we have made sure that all the distributions at the previous time-level are non-negative, and the discretization of the boundary conditions will not disrupt the non-negativity of distributions. Since the distributions at time t are non-negative, equilibrium distributions f_i^{eq} will be non-negative in the calculation of the collision step at time $t + \Delta t$. That is,

$$\begin{aligned} f_i\left(x, t\right) \geq 0 \Rightarrow u\left(\boldsymbol{x}, t\right) &= \sum_i f_i\left(x, t\right) \geq 0 \\ &\Rightarrow f_i^{\mathrm{eq}}\left(\boldsymbol{x}, t\right) = w_i u\left(\boldsymbol{x}, t\right) \geq 0 \cdot \end{aligned} \tag{4.33}$$

If all of these conditions are satisfied, restricting the value of the relaxation time τ can lead to non-negative concentrations at all lattice nodes and for all time-levels. We require that

$$1 - 1/\tau \geq 0\cdot \tag{4.34}$$

Using equation (4.8) and the above inequality, one can obtain the following condition that ensures non-negativity of populations

$$\Delta t \geq \frac{\Delta x^2}{6\mathrm{D}}\cdot \tag{4.35}$$

That is, if the discretization parameters, Δt and Δx, satisfy inequality (4.35) then all the distributions f_i will be non-negative. Non-negativity of all f_i's implies the non-negativity of the concentration $u(\boldsymbol{x}, t)$, which stems directly from equation (4.10).

Note that this result is only valid for one-dimensional pure diffusion equation (i.e., the advection velocity is zero) and for $D1Q3$ lattice model. Furthermore, the above condition does not guarantee the preservation of the comparison principles. Deriving similar conditions for more sophisticated lattice models in two and three dimensions and for multiple-relaxation-time methods will require a more rigorous analysis.

Another noteworthy point is that we have put stronger conditions on the values of distributions f_i in order to meet the non-negative constraint for the concentration. In other words, for nodal concentrations to be non-negative we made sure that all the distributions are non-negative (i.e., $f_i(\boldsymbol{x}, t) \geq 0 \ \forall i$). However, this condition can be relaxed by allowing some of the f_i to be negative, but with an additional constraint that $\sum_{i=0}^{m-1} f_i(\boldsymbol{x}, t) \geq 0$. We do not pursue such an approach here, but one can consider them in future developments of lattice Boltzmann methods.

4.6 Concluding Remarks

The maximum and comparison principles are two important mathematical properties of diffusion-type equations. The non-negative constraint is an important physical constraint on the concentration in transport and reactive-transport equations. There are other properties that the solutions to diffusion-type equations satisfy under appropriate conditions on the input data; for example, the decay property. A main challenge in designing a predictive numerical formulation is to satisfy these mathematical principles and physical constraints in the discrete setting. In this chapter, using representative numerical examples, we have systematically documented that the current LBM-based formulations do not satisfy the maximum principle, the comparison principle, the non-negative constraint, and the decay property. We have also shown that the discretization of boundary conditions has an effect on the performance of the lattice Boltzmann method in meeting these properties. To this end, we proposed a new way of discretizing Dirichlet boundary conditions – the weighted splitting method. We then derived a theoretical bound in terms of the time-step and lattice cell size that guarantees non-negative values for the concentration under the weighted splitting method for one-dimension problems.

It needs to be emphasized that stability conditions for the lattice Boltzmann method (i.e., Courant-Fredrichs-Lewy conditions) have been satisfied in all our numerical experiments. *This implies that meeting stability conditions alone does not guarantee the preservation of the mentioned mathematical principles in the discrete setting.* The main findings of the chapter about LBM-based formulations can be summarized as follows:

(a) One-dimensional problems: For a given time-step, one can eliminate the violation of the non-negative constraint and the maximum principle by refining the lattice cell size. For a given lattice cell size, the violation of the non-negative constraint and the maximum principle cannot be eliminated by decreasing the time-step. Both these trends are similar to FEM.

(b) Critical time-step: Based on a simple theoretical analysis, we found that $\Delta t \geq \Delta x^2/(6D)$ in order to meet the non-negative constraint under LBM for 1D problems. One can obtain exactly the same bound under the single-field Galerkin finite element method based on the backward Euler time-stepping scheme for 1D problems . This is an interesting result given the fact that the underlying basis of the lattice Boltzmann method (which solves Boltzmann equation to obtain distributions at lattice nodes) is completely different from that of the finite element method (which is based on a weak formulation).

(c) Isotropic vs. anisotropic diffusion: The violations of the non-negative constraint and the maximum principle are smaller in magnitude and smaller in terms of spatial extent when the diffusion is isotropic. Also, for a given time-step, one can decrease these violations by refining the lattice cell size in the case of isotropic diffusion. On the other hand, neither decreasing the time-step nor refining the lattice cell size will eliminate the violation of the non-negative constraint for anisotropic diffusion.

(d) Non-convex domains: The magnitudes of the violation of the non-negative constraint are larger for non-convex domains. However, it needs to be emphasized that one may have violations even on convex domains.

(e) Comparison principle: The comparison and maximum principles should be accounted for separately. Satisfaction of one does not lead to satisfaction of the other.

(f) Decay property: The LBM-based formulations, in general, violate the decay property.

(g) The lattice Boltzmann method does not posses a variational structure similar to the one possessed by the finite element method. Due to this reason, the non-negative formulations proposed under the finite element method cannot be directly extended to the lattice Boltzmann method.

(h) The only procedure that is available to meet the non-negative constraint under the lattice Boltzmann method for anisotropic diffusion, is the clipping procedure, which basically chops off the negative values. But, this procedure fixes neither the violation of the decay property nor the violation of the comparison principle. Moreover, this method does not have any physical or mathematical basis, and it is rather *ad hoc*.

However, the reader may wonder how important properties such as maximum principle or the non-negative constraint may be violated when LBM is shown to converge to Equation (4.3). In the following section we will provide ample evidence that current LBMs violate the aforementioned properties. Furthermore, convergence in a Chapman-Enskog analysis may not be the correct tool to study preservation of discrete maximum principle. Violation of maximum principle and the non-negative constraint by LBM can be found in a number of References: for instance cf. Figure 9 in [96], and Figures 8-10 in [97]. Even though the methods presented in these References are stable and converge to the advection-diffusion equation, they do not preserve maximum principle. *A known drawback of the Chapman-Enskog analysis is that it introduces spurious oscillations.* Reference [98] provides a beautiful discussion on this issue. A recent work that provides thorough mathematical analysis based on L^{∞}-convergence of the LBM solution and preservation of maximum principle for one-dimensional diffusion equation is presented in [99]. Another possibility is to equip the lattice Boltzmann method with the Boltzmann's H-theorem in order to guarantee the non-negativity of the discrete distributions [100]. It should be noted that occurrence of negative distributions, which may lead to violation of non-negative constraint, is not unique to the solution of advection-diffusion equation, but may also happen in numerical solution of various flow problems.

One should be wary of violations of the non-negative constraint, and the maximum and comparison principles in the numerical simulations using LBM. In the case of *isotropic* diffusion, the authors suggest investigating the occurrence of the mentioned violations, if any of these violations occur, they can be significantly reduced by refining the lattice cell size and the time-step in accordance with the CFL condition. However, in the case of *anisotropic* diffusion, no clear-cut guideline for reducing the violations exists. As demonstrated earlier, refining the discretization parameters may not improve the numerical solution. A future research direction could be development of LBM-based formulations for diffusion and advection-diffusion equations that respect the maximum and comparison principles, and meet the non-negative constraint. This chapter can also serve as a source of benchmark problems for such a research endeavor.

Chapter 5

HYBRID COUPLING FOR ADVECTION-DIFFUSION EQUATION

5.1 Introduction and Motivation

Transport of chemical species in porous media features a wide variety of time- and length-scales. Reaction and precipitation at the interface of fluid and solid [101–103], reactive flow and transport [104], and varied dynamics of (bio-)chemical reactions [105] are a few of the processes that occur at disparate scales. The three length-scales that are typically considered in the study on porous media are pore-scale (also referred to as fine-scale or micro-scale), meso-scale (also referred to as continuum-

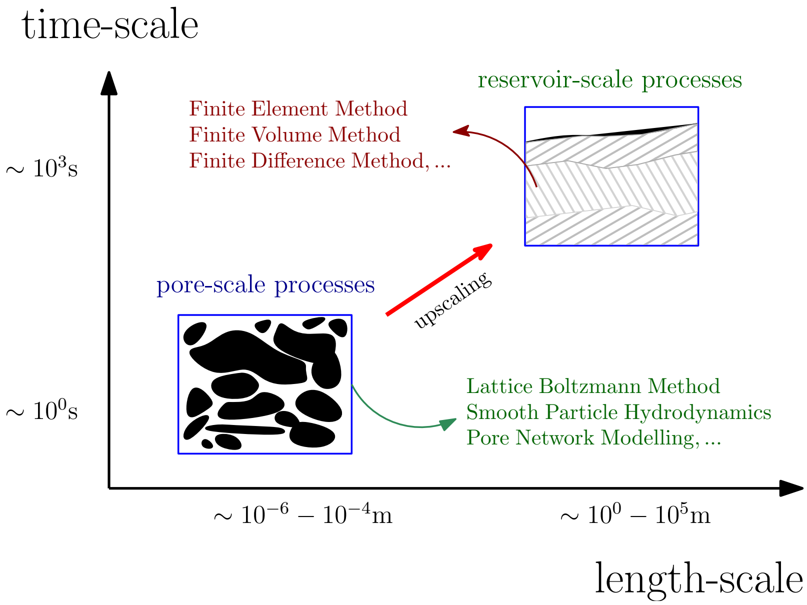


Figure 5.1: Disparate length and time-scales: This figure illustrates the disparity in time- and length-scales in porous media simulations.

scale or coarse-scale) and macro-scale (also referred to as field-scale). The properties of interest depend on the scale of observation, which implies that different modeling approaches are needed at different scales. Moreover, a numerical method appropriate for a particular length- or time-scale need not be a viable approach at a different scale. Due to this scale disparity, as shown in figure 5.1, the choice of a particular modeling approach or a particular numerical methodology that is appropriate for all the scales of observation is severely limited.

Coarse-scale modeling

Finite Element Method (FEM), Finite Volume Method (FVM) and Finite Difference Method (FDM) are commonly practiced schemes for coarse-scale fluid dynamics computations. However, fine-scale features may not be immediately included into the numerical solutions from these methods. Some of the efforts towards improving upon this shortcoming are the Variational Multi-Scale method in [4], Generalized Finite Element Method [5], Multi-Scale Finite Element Method [6], which can include some fine-scale spatial features into the finite element solution via manipulation of interpolation functions or the weak formulation. Although coarse-scale models can be solved in a computationally efficient manner and can include some limited fine-scale features, these models are not capable of capturing all the important pore-scale processes and their impact at the meso-scale and field-scale [106]. The source of this deficiency is, partly, the dependence of the model parameters on the length-scale. Furthermore, some processes in reactive-transport (e.g., some pore-scale reactions) cannot be upscaled from pore-scale to meso-scale [107].

Pore-scale modeling

Methods such as pore network modeling [108], Smooth Particle Hydrodynamics (SPH) [109] and the Lattice Boltzmann Method (LBM) [74] are amongst the most popular methods for fine-scale simulations. In particular, LBM offers great potential in including kinetic and atomistic details into the computational model. This fact originates from the main purpose of LBM, which is to numerically solve the Boltzmann equation [110]. This equation can describe the distribution of particles of a system in the phase space at any thermodynamic state. Sophisticated gas-interface interaction models [111, 112] and kinetic relations can also be included in the solution of the Boltzmann equation [112, 113]. *Despite the advantages of LBM over coarse-scale methods such as FEM or FVM, its application to real-world problems in subsurface modeling is impractical due to prohibitive computational cost.*

Hybrid modeling

It is now becoming evident that a viable approach for simulation of reactive-transport in porous media should consist of both fine-scale and coarse-scale models; for example, see the discussion in [114]. The modeling approaches that employ both fine-scale and coarse-scale models are collectively referred to as *hybrid* modeling. The motivation for hybrid modeling is four-fold:

(i) There is a need for increasing local modeling accuracy in certain applications.

Some examples include flow and transport along thin fractures, and to model processes in the well-bore cement that may act as escape passages for carbon-dioxide in geological carbon sequestration.

(ii) The need for hybrid modeling can arise when continuum assumptions locally break down in critical parts of the domain. For example, reactive-transport modeling under advection-dominated or reaction-dominated conditions, as described in [115].

(iii) The need for incorporating effects of the surrounding media on the subdomain for accurate predictions of flow and transport [116].

(iv) To achieve a manageable computational cost to solve realistic problems arising in subsurface applications.

Recently, there is a surge in research activity in hybrid modeling. A non-iterative coupling method for SPH and coarse-scale averaged SPH was proposed in [117] for advection-diffusion-reaction equation and precipitation in porous media. Using SPH for different length-scales allows the mentioned method to avoid predictor-corrector iterations in each time-step. The multi-scale algorithm proposed in [115] is based on FVM and uses an iterative approach to resolve the disparate length-scales in a transport process. In [118] coupling of finite element method and pore network modeling for flow problems in porous media, using the mortar method was introduced for the first time. This method was then extended in [119] to couple FDM and pore network model for simulation of flow and transport of chemical species. It utilizes the mortar finite element spaces to transfer information from one subdomain to another. The unknowns are updated iteratively to satisfy continuity of fluxes at the interface within a user-defined tolerance. In [120], these mortar-based methods are used to couple finite difference and cellular automata methods to model the bio-film development in porous media. Coupling of FDM and LBM for advection-diffusion equation is studied in [121], but non-matching grids and disparate time-steps are not considered. A hybrid method that incorporates LBM and FEM for simulation of the diffusion processes is proposed in [122]. A more recent effort in this direction is given in [123] that allows different time-steps and grid sizes for FEM and LBM domains.

Multiple temporal scales and multi-time-step methods

Multi-time-step (multi-rate) methods aim at resolving the disparity in time-scales in a system through use of appropriate time-steps and time-integrators for each subsystem. In recent years, development of multi-time-step methods has received much attention among researchers of various fields. These include: multi-rate methods based on Runge-Kutta schemes [124, 125], adaptive variational integrators for dynamics [126], multi-time-step methods based on non-overlapping domain decomposition [57, 127], and symplectic multi-time-step methods for molecular dynamics simulations [128, 129]. Multi-time-step coupling algorithms based on domain partitioning are often classified as either *staggered* or *monolithic* coupling schemes [130]. Staggered coupling methods update the solution in different subdomains through a predictor-corrector procedure. Hence, there is a time-lag between the solutions at different subdomains, which can result in numerical instabilities. However, this type of algorithms enjoy tremendous popularity because of modularity; one can employ available solvers and use them (with different time-steps) in a staggered coupling algorithm without any major modification. Unlike staggered coupling algorithms, monolithic schemes update the solution in the entire domain using a single iteration. These is no time-lag between the solution of different subdomains. As a result, monolithic coupling algorithms enjoy much better numerical stability than staggered coupling methods. However, current numerical solvers for partial differential equations cannot be immediately included in a monolithic coupling scheme and a major effort in developing computer codes is required. Also, multi-time-step integration requires careful design of a coupling algorithm [57]. Due to the aforementioned reasons, we shall employ a staggered coupling approach.

Domain decomposition methods

A natural way to develop a staggered coupling method is to employ domain decomposition techniques, which also offer an attractive framework for parallel computing. Over the years, a variety of overlapping and non-overlapping domain decomposition techniques have been developed [7, 131, 132]. These methods have the potential to employ non-matching computational grids in different subdomains; for instance, mortar finite element spaces [133, 134] and overlapping methods [135, 136] are among them. However, having different grid-sizes in different subdomains may not be enough to account for disparate time-scales that can be present in the model problem. In order to achieve computational efficiency for problems involving multiple temporal scales, one needs to employ tailored numerical time-integrators and time-steps for each active process. That is, domain decomposition techniques and multi-time-stepping schemes go hand in hand. Herein, we employ overlapping domain decomposition technique whose advantages will be discussed later.

An outline of the chapter

We provide an overview of our approach in Section 5.2. Section 5.3 provides the governing equations at the continuum-scale and the associated numerical modeling. Section 5.4 discusses the modeling at the pore-scale using the lattice Boltzmann method. An overview of overlapping domain decomposition techniques and information transfer across non-matching grids is given in Section 5.5. In Section 5.6, we present a robust hybrid multi-time-step coupling method that allows to couple pore-scale and continuum-scale subdomains. Section 5.7 presents several numerical results using the proposed hybrid modeling, and illustrates the robustness and utility of the proposed computational framework. Finally, conclusions are drawn in Section 5.8 along with a discussion on possible future research endeavors in the area of hybrid modeling.

5.2 An Overview of Our Approach

In this chapter, we present a hybrid method to couple the advection-diffusion equation at the continuum-scale with the Boltzmann equation at the pore-scale to

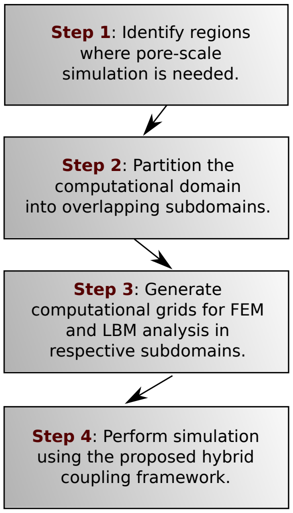

Step 1: Identify regions where pore-scale simulation is needed.
Step 2: Partition the computational domain into overlapping subdomains.
Step 3: Generate computational grids for FEM and LBM analysis in respective subdomains.
Step 4: Perform simulation using the proposed hybrid coupling framework.

Figure 5.2: The main steps under the proposed hybrid framework.

simulate the transport of chemical species. The proposed method can capture fine-scale features and processes by solving the lattice Boltzmann equation at the pore-scale. The response at the continuum-scale is captured by solving the advection-diffusion equation using the finite element method.

We use the domain decomposition technique to partition the computational domain into *fine-scale* and *coarse-scale* subdomains. For better numerical stability, we allow the coarse-scale and fine-scale computational subdomains to overlap, and appropriate boundary conditions are designed at the boundary of the individual computational subdomains. To capture disparate time-scale, the proposed computational framework allows different time-steps and different time integration algorithms in different subdomains. Furthermore, computational grids and different orders of interpolation can be employed in different subdomains. This enables the user to choose appropriate time-step, mesh and interpolation in each subdomain for stability and desired accuracy.

The first step in a hybrid simulation using the proposed framework is to partition the computational domain into regions for fine-scale and coarse-scale modeling by identifying the regions where pore-scale analysis is needed. Thanks to the design of the proposed framework, creating computational meshes for these two types of subdomains is easy and can be carried out independent of each other. Finally, the analysis is carried out by using appropriate models in different subdomains. The overall procedure is summarized in figure 5.2. Some of the salient features of the proposed framework are as follows:

(i) Various transport processes and reactions can be incorporated into the analysis. In particular, the user can include complex advection velocity field (which is encountered frequently in porous media applications) and cascade of geochemical reactions without any change in the design of the coupling framework.

(ii) One can divide the computational domain into multiple subdomains, and can independently employ in each subdomain either pore-scale modeling or continuum-scale modeling.

(iii) The computational grids in a subdomain need not conform with the computational grid in another subdomain. In particular, the finite element mesh in the continuum-scale modeling need not match with the lattice structure in the lattice Boltzmann method, which is employed in the pore-scale analysis. This provides a great degree of flexibility for the modelers.

a) One can employ meshes with different degrees of approximation independent of other subdomains. There is no need for compatibility among the interpolation spaces (e.g., mortar spaces) along the subdomain interfaces.

b) This allows to leverage on the existing computational methods for modeling at pore-scale and continuum-scale. There is no need to design new methods just to be compatible with the hybrid coupling.

(iv) One can employ different time-steps and different time stepping schemes in different subdomains, which is an attractive feature to possess for solving problems involving multiple temporal scales.

(v) An accurate transfer of data across non-matching grids has been incorporated into the proposed computational framework.

(vi) A novel way of implementing boundary conditions has been developed for the discretization under the lattice Boltzmann method. This enhances the accuracy at the pore-scale, and hence the overall accuracy of a hybrid coupling.

(vii) No initial guess at the interface of subdomains is needed. Hence, implementation is easier and the simulation procedure can be fully automated.

In order to achieve aforementioned features, the computational framework is developed by integrating the following main ingredients:

(a) A solver for continuum-scale modeling, which in our case will be a finite element formulation for advection-diffusion/dispersion equations.

(b) A solver for pore-scale modeling, which in our case will be the lattice Boltzmann method with an improved discretization of boundary conditions.

(c) An overlapping domain decomposition framework.

(d) An accurate algorithm to transfer scalars, vectors and tensors across non-matching computational grids.

(e) An iterative coupling algorithm to ensure compatibility of solution across the overlapping region.

A computational framework with the aforementioned features, which is essential to gain a fundamental understanding of subsurface processes, is currently not available. We therefore strive to design such a framework in this chapter. The details of the aforementioned ingredients along with the illustration of the performance of the proposed hybrid multi-time-step computational framework are provided in subsequent sections.

5.3 Continuum-Scale Modeling

We shall model the transport at the continuum-scale using unsteady advection-dispersion equations. To this end, consider a bounded open domain $\Omega_{\mathrm{c}} \subset \mathbb{R}^{nd}$ on which we seek to perform continuum modeling, where $\mathbb{R}$ denotes the set of real numbers and "*nd*" is the number of spatial dimensions. We assume that the boundary of this domain, $\partial\Omega_{\mathrm{c}}$, is comprised of $\Gamma_{\mathrm{c}}^{\mathrm{N}}$ and $\Gamma_{\mathrm{c}}^{\mathrm{D}}$ such that we have

$$\partial\Omega_{\mathrm{c}} = \Gamma_{\mathrm{c}}^{\mathrm{N}} \cup \Gamma_{\mathrm{c}}^{\mathrm{D}} \quad \text{, and} \quad \Gamma_{\mathrm{c}}^{\mathrm{D}} \cap \Gamma_{\mathrm{c}}^{\mathrm{N}} = \varnothing\cdot \tag{5.1}$$

Dirichlet boundary conditions are enforced on $\Gamma_{\mathrm{c}}^{\mathrm{D}}$, and Neumann boundary conditions are enforced on $\Gamma_{\mathrm{c}}^{\mathrm{N}}$. A spatial point in Ω_{c} will be denoted by $\mathbf{x}$. We use div$[\cdot]$

and grad$[\cdot]$, respectively, to denote the spatial divergence and gradient operators. The time interval of interest is denoted by $\mathcal{I} = (0, \mathrm{T}]$, and the time is denoted by t. The initial boundary value problem at the continuum-scale can be written as follows:

$$\frac{\partial \mathrm{u}}{\partial \mathrm{t}} + \mathrm{div}\left[\mathbf{v}\mathrm{u} - \mathrm{D}\,\mathrm{grad}\left[\mathrm{u}\right]\right] = \mathrm{s} \quad (\mathbf{x}, \mathrm{t}) \in \Omega_{\mathrm{c}} \times \mathcal{I}, \tag{5.2a}$$

$$\mathrm{u}(\mathbf{x}, \mathrm{t} = 0) = \mathrm{u}_0(\mathbf{x}) \quad \mathbf{x} \in \Omega_{\mathrm{c}}, \tag{5.2b}$$

$$\mathrm{u}(\mathbf{x}, \mathrm{t}) = \mathrm{u}^{\mathrm{p}}(\mathbf{x}, \mathrm{t}) \quad (\mathbf{x}, \mathrm{t}) \in \Gamma_{\mathrm{c}}^{\mathrm{D}} \times \mathcal{I}, \text{ and} \tag{5.2c}$$

$$\widehat{\mathbf{n}} \cdot (\mathbf{v}\mathrm{u} - \mathrm{D}\,\mathrm{grad}[\mathrm{u}]) = \mathrm{q}^{\mathrm{p}} \quad (\mathbf{x}, \mathrm{t}) \in \Gamma_{\mathrm{c}}^{\mathrm{N}} \times \mathcal{I}, \tag{5.2d}$$

where u is the concentration, $\mathbf{v}$ is the divergence-free advection velocity (i.e., div$[\mathbf{v}]$ = 0 in Ω_{c}), D is the dispersion coefficient, and s is the source/sink term. Although the dependence is not explicitly indicated, all of the mentioned quantities depend on the spatial coordinates and time. The dispersion coefficient D is positive and can be spatially heterogeneous. The initial concentration in Ω_{c} is denoted by u_0, and u^{p} is the prescribed concentration on $\Gamma_{\mathrm{c}}^{\mathrm{D}}$. The outward unit normal to $\partial\Omega_{\mathrm{c}}$ is denoted by $\widehat{\mathbf{n}}$, and q^{p} is the prescribed flux on $\Gamma_{\mathrm{c}}^{\mathrm{N}}$.

The solution to the above equations can exhibit disparate spatial and temporal scales, which depend on the relative strengths of advection, dispersion and reaction processes, and volumetric source [137, 138]. We employ the finite element method for the numerical modeling at the continuum-scale.

The finite element method

We shall introduce the following function spaces:

$$\mathsf{C} := \left\{ \mathrm{u} : \Omega_{\mathrm{c}} \times \mathcal{I} \to \mathbb{R} \,\middle|\, \mathrm{u}(\mathbf{x}, \mathrm{t}) \in H^1(\Omega_{\mathrm{c}}) \text{ and } \mathrm{u}(\mathbf{x} \in \Gamma_{\mathrm{c}}^{\mathrm{D}}, \mathrm{t}) = \mathrm{u}^{\mathrm{p}} \ \forall \mathrm{t} \in \mathcal{I} \right\}, \text{ and} \tag{5.3a}$$

$$\mathsf{W} := \left\{ \mathrm{w} : \Omega_{\mathrm{c}} \to \mathbb{R} \,\middle|\, \mathrm{w}(\mathbf{x}) \in H^1(\Omega_{\mathrm{c}}) \text{ and } \mathrm{w}(\mathbf{x} \in \Gamma_{\mathrm{c}}^{\mathrm{D}}) = 0 \right\}, \tag{5.3b}$$

where $H^1(\Omega_{\mathrm{c}})$ is a Sobolev space defined on Ω_{c} [45]. We shall denote the standard L_2 inner product over a set $\mathcal{K}$ as follows:

$$(\mathrm{w}, \mathrm{u})_{\mathcal{K}} \equiv \int_{\mathcal{K}} \mathrm{w} \cdot \mathrm{u} \, \mathrm{d}\mathcal{K} \cdot \tag{5.4}$$

For convenience, we shall drop the subscript $\mathcal{K}$ if $\mathcal{K} = \Omega_{\mathrm{c}}$. We shall employ the semi-discrete methodology to decouple the spatial and temporal discretizations [139]. There are a multitude of ways to construct a weak formulation for equations (5.2a)–(5.2d). In this chapter, we shall limit to the Galerkin formulation and the Streamline Upwind/Petrov-Galerkin formulation [140]. However, it should be noted that any other finite element (or even a finite volume) formulation can also be employed in the modeling at the coarse-scale.

The Galerkin formulation.

Find $\mathrm{u}(\mathbf{x},\mathrm{t}) \in \mathsf{C}$ such that we have

$$(\mathrm{w}, \partial \mathrm{u}/\partial \mathrm{t}) + (\mathrm{w}, \mathbf{v} \cdot \mathrm{grad}[\mathrm{u}]) + (\mathrm{grad}[\mathrm{w}], \mathrm{D}\mathrm{grad}[\mathrm{u}]) = (\mathrm{w}, \mathrm{s}) + (\mathrm{w}, \mathrm{q}^{\mathrm{p}})_{\Gamma_{\mathrm{c}}^{\mathrm{N}}} \quad \forall \mathrm{w} \in \mathsf{W}, \tag{5.5a}$$

$$\mathrm{u} = \mathrm{u}_0 \quad \forall \mathbf{x} \in \Omega_{\mathrm{c}}, \ \mathrm{t} = 0, \text{ and} \tag{5.5b}$$

$$\mathrm{u} = \mathrm{u}^{\mathrm{p}} \quad \forall (\mathbf{x}, \mathrm{t}) \in \Gamma_{\mathrm{c}}^{\mathrm{D}} \times \mathcal{I}. \tag{5.5c}$$

Note that the Dirichlet boundary conditions are enforced strongly. We shall employ the Galerkin formulation only for dispersion-dominated problems, as this formulation is known to perform poorly for advection-dominated problems. This shortcoming can be partly alleviated by employing a stabilized formulation instead.

The Streamline Upwind/Petrov-Galerkin (SUPG) formulation

The SUPG formulation is a popular stabilized formulation, and it reads as follows: Find $\mathrm{u}(\mathbf{x},\mathrm{t}) \in \mathsf{C}$ such that we have

$$\begin{aligned}(\mathrm{w}, \partial \mathrm{u}/\partial \mathrm{t}) &+ (\mathrm{w}, \mathbf{v} \cdot \mathrm{grad}[\mathrm{u}]) + (\mathrm{grad}[\mathrm{w}], \mathrm{D}\mathrm{grad}[\mathrm{u}]) \\ &+ \sum_{e=1}^{N_{\mathrm{FEM}}} (\mathbf{v} \cdot \mathrm{grad}[\mathrm{w}], \tau_e \mathrm{R}[\mathrm{u}])_{\Omega_{\mathrm{c}}^{e}} = (\mathrm{w}, \mathrm{s}) + (\mathrm{w}, \mathrm{q}^{\mathrm{p}})_{\Gamma_{\mathrm{c}}^{\mathrm{N}}} \quad \forall \mathrm{w} \in \mathsf{W}\end{aligned} \tag{5.6}$$

where N_{FEM} is the total number of finite elements and τ_e is the stabilization parameter for element e. The spatial domain contained in element e is shown by Ω_{c}^{e}. The residual $\mathrm{R}[\mathrm{u}]$ is defined as follows:

$$\mathrm{R}[\mathrm{u}] = \frac{\partial \mathrm{u}}{\partial \mathrm{t}} + \mathrm{div}\left[\mathbf{v}\mathrm{u} - \mathrm{D}\mathrm{grad}\left[\mathrm{u}\right]\right] - \mathrm{s}. \tag{5.7}$$

The Dirichlet boundary condition and initial condition will remain as in equation (5.5). We employ the stabilization parameter τ_e as given in [141]. That is,

$$\tau_e = \frac{h_e}{2p\,\|\mathbf{v}\|}\chi\left(P_e^h\right), \quad \text{and} \quad P_e^h = \frac{\|\mathbf{v}\|\,h_e}{2p\mathrm{D}}, \quad \chi(\alpha) = \coth(\alpha) - 1/\alpha, \tag{5.8}$$

where p is the order of finite element interpolation functions and D is the *isotropic* coefficient of dispersion. The element size is denoted by h_e, and P_e^h is the element Péclet number.

5.4 Pore-Scale Modeling: The Lattice Boltzmann Method

We shall use Ω_{f} to denote the region in which one seeks to perform pore-scale modeling. We use the Boltzmann equation to describe the transport processes at the pore-scale. The Boltzmann equation provides a statistical description of the state of matter away from the thermodynamic equilibrium [112]. This equation describes the evolution of the distribution of particles in the phase space, from which macroscopic variables can be easily computed by taking appropriate moments. If one neglects the external force term, the Boltzmann equation can be written as

$$\frac{\partial f}{\partial \mathrm{t}} + \mathbf{v}\cdot\mathrm{grad}[f] = \mathcal{Q}\left[f, f^{\mathrm{eq}}\right] \qquad \text{in } \Omega_{\mathrm{f}}, \tag{5.9}$$

where f is the distribution function, $\mathbf{v}$ is the macroscopic (or background) velocity, and $\mathcal{Q}$ is the collision term. Herein, we will consider the Bhatnagar-Gross-Krook (BGK) collision model [89], which can be written as

$$\mathcal{Q}\left[f, f^{\mathrm{eq}}\right] = \frac{1}{\lambda}(f^{\mathrm{eq}} - f), \tag{5.10}$$

where f^{eq} is the distribution of particles in the phase space at the thermodynamics equilibrium. The parameter λ is called the relaxation time. In this chapter, we will use the Maxwell-Boltzmann distribution for the equilibrium distribution f^{eq}. That is,

$$f^{\mathrm{eq}}(\mathbf{x}, \boldsymbol{\zeta}; \mathbf{v}, \mathrm{u}) = \frac{\mathrm{u}}{\sqrt{2\pi \mathrm{RT}}} \exp[-(\boldsymbol{\zeta} - \mathbf{v}) \cdot (\boldsymbol{\zeta} - \mathbf{v})/2\mathrm{RT}], \tag{5.11}$$

where $\mathrm{u}(\mathbf{x}, \mathrm{t})$ is the concentration, R is the ideal gas constant, and T is the temperature. The velocity of particles is indicated by $\boldsymbol{\zeta}$.

In order to provide a complete description of the Boltzmann equation for a physical problem, appropriate boundary conditions have to be included. Fortunately, the mathematical theory of boundary conditions for Boltzmann equation is rather well-developed; for example, see [111–113]. However, to use the existing theories to their full extent, one has to incorporate detailed dynamics for interaction between the particle and the surrounding surface of the domain. Obviously, the continuum model problem given in equation (5.2) lacks such information. Hence, the Boltzmann equation provides a much more powerful framework to account for detailed dynamics of gas-surface interaction that the continuum model is incapable of. In this chapter, we will assume that the user merely intends to replicate the behavior of the macroscopic solution of equation (5.2) and an in-depth treatment of gas-surface interaction is of no interest. In the following, we will provide an overview of discretization of Boltzmann equation.

The lattice Boltzmann method

The lattice Boltzmann method is a popular numerical method to solve the Boltzmann equation (5.9). This method offers great potential for parallelization [142] and simulation in domains with complex spatial features [75]. We will employ standard lattice model $DnQm$ to discretize the velocity space. These discrete velocities are identified by vectors $\boldsymbol{e}_i,\ i = 1, \cdots, m$. The discrete population corresponding to the lattice velocity $\boldsymbol{e}_i$ is denoted by f_i. Considering the lattice cell size of h and a time-step Δt, the discrete form of Boltzmann equation can be written as

$$|f_i\rangle\left(\boldsymbol{x} + \boldsymbol{e}_i\Delta t, t + \Delta t\right) = |f_i\rangle\left(\boldsymbol{x}, t\right) + |\mathcal{Q}_i\rangle\left(\boldsymbol{x}, t\right), \tag{5.12}$$

where the column vector of discrete populations is denoted using the Dirac notation $|\cdot\rangle$ [143]. Location of a lattice node is shown by $\boldsymbol{x}$ and t is a discrete time-level. The discrete collision operator $\mathcal{Q}_i$ is defined as

$$|\mathcal{Q}_i\rangle\left(\boldsymbol{x}, t\right) = \frac{1}{\tau}\left(|f_i^{\text{eq}}\rangle\left(\boldsymbol{x}, t\right) - |f_i\rangle\left(\boldsymbol{x}, t\right)\right) \cdot \tag{5.13}$$

The non-dimensional relaxation-time is denoted by τ and is defined as

$$\tau = \frac{1}{2} + \frac{D}{c_s^2 \Delta t}, \tag{5.14}$$

with c_s being the lattice sound velocity (e.g., in the case of $D2Q9$ lattice model $c_s = \Delta x / \sqrt{3}\Delta t$). For the equilibrium distribution in equation (5.13), we will employ the following approximation to the Maxwell-Boltzmann distribution in equation (5.11) as

$$f_i^{\text{eq}}(\boldsymbol{x}, t; u) = w_i u \left(1 + \frac{\boldsymbol{e}_i \cdot \mathbf{v}}{c_s^2} + \frac{1}{2}\frac{(\boldsymbol{e}_i \cdot \mathbf{v})^2}{c_s^4} - \frac{\mathbf{v} \cdot \mathbf{v}}{c_s^2}\right), \tag{5.15}$$

where w_i is the weight associated with $\boldsymbol{e}_i$ and $\mathbf{v}$ is the advection velocity. Concentration is denoted by u. In the rest of the chapter, we will assume that $\|\mathbf{v}\|/c_s \ll 1$ (low Mach number assumption). Macroscopic quantities of interest, in this case concentration and flux, can be obtained by the following relations

$$u(\boldsymbol{x},t) = \sum_{i=1}^{m} f_i(\boldsymbol{x},t) \text{ and} \tag{5.16a}$$

$$\boldsymbol{q}(\boldsymbol{x},t) = \sum_{i=1}^{m} f_i(\boldsymbol{x},t)\boldsymbol{e}_i\cdot \tag{5.16b}$$

Despite ever-growing popularity of lattice Boltzmann methods for computational fluid dynamics assumptions, these methods are prone to produce unphysical values for populations f_i; for example, see [144]. Obviously, for equation (5.9) to be meaningful, the value of population f needs to be non-negative. Some of the approaches toward resolving this issue can be found in [145–147]. Herein, we will propose a simple condition on the LBM discretization that guarantees non-negative values for discrete populations. We assume that the initial values for populations are also non-negative; for instance, this can be achieved if one takes $f_i(\boldsymbol{x},t=0) = f_i^{\mathrm{eq}}(\boldsymbol{x},t=0)$. From equations (5.12) and (5.13) we can conclude that if $1-1/\tau \geq 0$, then the discrete populations at time-level $t+\Delta t$ will also be non-negative. Note that the streaming process does not contribute to negativity (an appropriate treatment of boundary conditions comes later). This condition leads to the following result

$$\Delta t \leq \frac{2\mathrm{D}}{c_s^2}\cdot \tag{5.17}$$

For instance, in the case of a $D2Q9$ lattice model, one should have

$$h^2 \leq 6\Delta t\mathrm{D}\cdot \tag{5.18}$$

The advantage of this method compared to methods such as entropic lattice Boltzmann method is that one does not need to solve a non-linear equation at each lattice node. Hence, it is much easier to code and computationally more efficient. Furthermore, the standard collision and streaming steps in the lattice Boltzmann method have remained untouched and no further modification is necessary.

To complete the description of lattice Boltzmann method for advection-diffusion equation, we need to demonstrate how to apply boundary conditions so that the resulting discrete populations are non-negative. In the following we will provide new methods for enforcing Dirichlet- and Neumann-type boundary conditions in equation (5.2).

Boundary conditions for the lattice Boltzmann method

Over the past few decades, a multitude of methods for enforcing macroscopic boundary conditions in the context of lattice Boltzmann methods for flow and transport equations have been proposed. For example, see [92, 94, 148]. However, note that the boundary conditions typically considered for flow or transport problems in a macroscopic framework do not imply a unique configuration of particles in phase space. Another drawback of these methods can be that they may result in negative values for discrete populations. Other physical properties of the solution, such as monotonicity of entropy production may also be lost following enforcement of boundary conditions. Designing a numerical method to enforce boundary conditions appropriately, is indeed a challenging topic. Herein, to partially rectify the aforementioned problems, we propose a new framework for enforcing Dirichlet and Neumann boundary conditions for lattice Boltzmann method. These methods are based on the assumption that the system encompassed in domain Ω is connected to a bath of particles that reside in a specific state of thermodynamics equilibrium. This thermodynamic state can be identified by maximizing the entropy function, subject to a hydrodynamic constraint.

In this chapter, we will utilize the Boltzmann's $\mathcal{H}$ function to find the state of equilibrium. Obviously, the entropy $\mathcal{S}$ is related to the $\mathcal{H}$ function via the relation

$$\mathcal{S} = -\mathcal{H}. \tag{5.19}$$

Hence, maximization of entropy $\mathcal{S}$ is mathematically equivalent to minimizing $\mathcal{H}$. The $\mathcal{H}$ function at each point is defined as

$$\mathcal{H}(|f_i\rangle)\big|_{(\boldsymbol{x},t)} = \sum_{i=1}^{m} f_i(\boldsymbol{x},t)\ \log\left[\frac{f_i(\boldsymbol{x},t)}{w_i}\right]. \tag{5.20}$$

For brevity, we will use the notation

$$\mathsf{M}^{-}(\mathbf{x}) := \left\{i \middle| \, \boldsymbol{e}_i \cdot \widehat{\mathbf{n}}(\mathbf{x}) < 0,\ i = 1, \cdots, m \text{ and } \mathbf{x} \in \partial\Omega\right\}, \tag{5.21}$$

where $\widehat{\mathbf{n}}(\mathbf{x})$ is the unit outward normal to $\partial\Omega$. Obviously, the unknown populations near the boundary are f_i with $\boldsymbol{e}_i \in \mathsf{M}^{-}(\mathbf{x})$ for every point $\mathbf{x}$ on $\partial\Omega$. The rest of the discrete populations are known from the collision and streaming steps prior to enforcement of boundary conditions.

(a) *Dirichlet boundary condition*: Let $\mathbf{x} \in \Gamma^{\mathrm{D}}$ and $\mathrm{u}^{\mathrm{p}}(\mathbf{x},t)$ be the prescribed concentration at that point. The unknown populations are calculated by solving the following constrained optimization problem as

$$\min_{f_j,\, j \in \mathsf{M}^-(\mathbf{x})} \ \mathcal{H}\left(\left|f_i\right\rangle\right)\Big|_{(\mathbf{x},t)}, \tag{5.22a}$$

$$\text{subject to } \sum_{i=1}^{m} f_i(\mathbf{x},t) = \mathrm{u}^{\mathrm{p}}(\mathbf{x},t), \tag{5.22b}$$

where the function $\mathcal{H}$ is defined in equation (5.20). This minimization problem will result in relation

$$f_i(\mathbf{x},t) = \frac{w_i}{\sum_{j\in \mathsf{M}^-(\mathbf{x})} w_j} \left(\mathrm{u}^{\mathrm{p}}(\mathbf{x},t) - \sum_{k \notin \mathsf{M}^-(\mathbf{x})} f_k(\mathbf{x},t) \right) \quad i \in \mathsf{M}^-(\mathbf{x})$$

$$f_i(\mathbf{x},t) \geq 0 \quad \forall i = 1, \cdots, m\cdot \tag{5.23}$$

Note that the function $\mathcal{H}$ is only defined for non-negative arguments.

(b) *Neumann boundary condition*: Let $\mathbf{x} \in \Gamma^{\mathrm{N}}$ and $\mathrm{q}^{\mathrm{p}}(\mathbf{x},t)$ be the prescribed flux at that point. The unknown populations are governed by the following constrained optimization problem

$$\min_{f_j,\, j \in \mathsf{M}^-(\mathbf{x})} \ \mathcal{H}(\left|f_i\right\rangle)\Big|_{(\mathbf{x},t)}, \tag{5.24a}$$

$$\text{subject to } \sum_{i=1}^{m} f_i(\mathbf{x},t)\boldsymbol{e}_i \cdot \widehat{\mathbf{n}}(\mathbf{x}) = \mathrm{q}^{\mathrm{p}}(\mathbf{x},t), \tag{5.24b}$$

with $\widehat{\mathbf{n}}(\mathbf{x})$ being the unit outward normal defined earlier. This minimization problem will result in a non-linear equation in terms of Lagrange multiplier for the hydrodynamic constraint

$$\sum_{i\in\mathsf{M}^-(\mathbf{x})} w_i(\boldsymbol{e}_i \cdot \widehat{\mathbf{n}}(\mathbf{x}))\exp\left[-1 - \gamma \boldsymbol{e}_i \cdot \widehat{\mathbf{n}}(\mathbf{x})\right] = \mathrm{q}^{\mathrm{p}}(\mathbf{x},t) - \sum_{j\notin\mathsf{M}^-(\mathbf{x})} (\boldsymbol{e}_j \cdot \widehat{\mathbf{n}}(\mathbf{x}))f_j(\mathbf{x},t), \tag{5.25}$$

where γ is the Lagrange multiplier. Once the value of γ is known, the populations can be found using the relation

$$f_i(\mathbf{x}, t) = w_i \exp\left[-1 - \gamma \boldsymbol{e}_i \cdot \widehat{\mathbf{n}}(\mathbf{x})\right] \quad i \in \mathsf{M}^-(\mathbf{x}), \tag{5.26}$$

which guarantees non-negative values for f_i. In the case of one-dimensional lattice models (e.g. the $D1Q2$ lattice) and the $D2Q4$ or $D2Q5$ models, this method reduces to the conventional bounce-back method. But, in general this method is different than bounce-back or specular reflection methods.

Through this method of enforcing boundary conditions, which is based on maximization of entropy, we ensure monotonic increase in entropy. The physical interpretation of this method is that the system in Ω is connected to systems in thermodynamic equilibrium. The adjacent systems connected to Ω through Γ^{D} and Γ^{N} are in different states of thermodynamic equilibrium. Minimization of the function $\mathcal{H}$ ensures that the equilibrium condition for particles near the boundary is respected. The constraints in equations (5.22a) and (5.24a) are the macroscopic hydrodynamic conditions of the system at the respective points of the domain.

A numerical example

We now assess the accuracy of the proposed methods for boundary conditions under LBM. Consider the domain $\Omega = (0,1) \times (0,1)$, with the diffusion coefficient $\mathrm{D} = \frac{4}{5\pi^2}$ and advection velocity being $\mathbf{v} = \mathbf{0}$. The source term is also zero throughout the domain. The initial concentration is taken as

$$\mathrm{u}_0(\mathrm{x}, \mathrm{y}) = \sin(\pi \mathrm{y}) \cos(\pi \mathrm{x}/2), \tag{5.27}$$

Table 5.1: Numerical results for LBM: In this table, numerical values for discretization parameters and the calculated error at time-level $t = 0.25$ are shown.

Case	h	Δt	$\mathcal{E}(t = 0.25)$
1	4×10^{-2}	3.3×10^{-3}	2.5×10^{-3}
2	2×10^{-2}	8.2×10^{-4}	6.2×10^{-4}
3	10^{-2}	2.1×10^{-4}	1.4×10^{-4}
4	5×10^{-3}	5.2×10^{-5}	1.7×10^{-5}

and the boundary conditions are

$$-\mathrm{D}\mathrm{grad}[\mathrm{u}] \cdot \widehat{\mathbf{n}} = 0 \quad \text{on } \Gamma^{\mathrm{N}} \text{ and} \tag{5.28a}$$

$$\mathrm{u} = 0 \quad \text{on } \Gamma^{\mathrm{D}}, \tag{5.28b}$$

where $\Gamma^{\mathrm{N}} = \{0\} \times (0,1)$ and $\Gamma^{\mathrm{D}} = \partial\Omega - \Gamma^{\mathrm{N}}$. We will employ the $D2Q9$ lattice model with grid spacing of h. The time-step will be chosen according to equation (5.17) to avoid negative values for discrete distributions. This problem is solved using several choices of discretization parameters as given in Table 5.1. We will use the following definition for calculation of error

$$\mathcal{E}(t) = \max_i \left\{ |u(\boldsymbol{x}_i, t) - \mathrm{u}_{\mathrm{exact}}(\boldsymbol{x}_i, t)| \right\}, \tag{5.29}$$

where $u(\boldsymbol{x}_i, t)$ is the computed numerical value at i-th node and time-level t. The exact solution is denoted by $\mathrm{u}_{\mathrm{exact}}$. Numerical results from LBM with the proposed methods for boundary conditions are shown in figure 5.3. The variation of error with respect to the cell-size has been documented in Table 5.1 and figure 5.4, which show a second-order convergence.

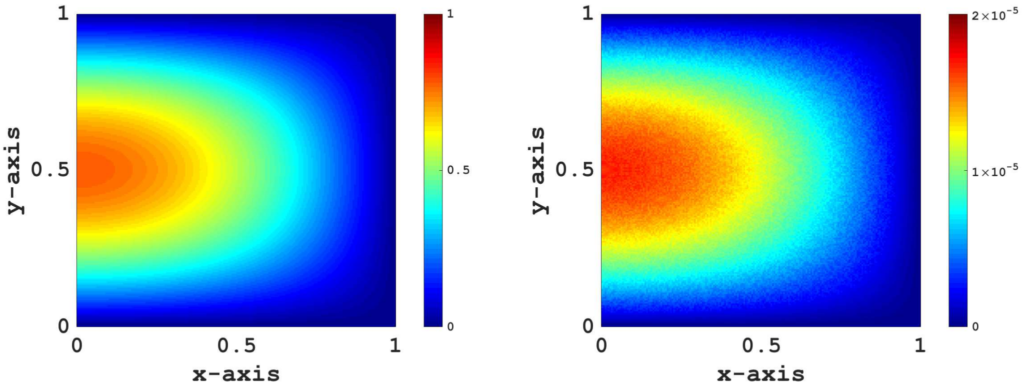
Comparison with other methods

Consider the domain $\Omega = (0,1) \times (0,1)$ with zero-flux boundary conditions enforced on $\partial\Omega$. The initial condition is taken as:

$$\mathrm{u}_0(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} \in [a,b] \times [a,b] \\ 0 & \text{otherwise} \end{cases}, \tag{5.30}$$

where we take $a = 0.4$ and $b = 0.6$. The diffusion coefficient is $\mathrm{D} = 10^{-2}$ and the advection velocity is zero. The $D2Q9$ lattice model is used. Figure 5.5 shows the numerical result from the lattice Boltzmann method, along with the proposed methods for enforcing boundary conditions. The bound given by equation (5.17) for cell-size and time-step is respected. Hence, all discrete populations, and consequently, concentration at all nodes are non-negative. The change in the Boltzmann $\mathcal{H}$ function is monotonic, which means that the $\mathcal{H}$-theorem is satisfied.

Note that a zero-flux boundary (or any other macroscopic boundary condition) can lead to various interpretations in the context of kinetic theory. For instance, a rigid and impermeable wall can lead to a zero-flux condition. Also, zero-flux can mean that there is a bath of particles at a Maxwell-Boltzmann equilibrium state with background velocity $\mathbf{v} = \mathbf{0}$. These interpretations are all valid in their own right. One needs to account for more physical details and choose the right method for enforcing those conditions. To show the difference in the numerical results due to different treatment of boundaries under lattice Boltzmann method, the given numerical example is solved using bounce-back and specular reflection methods [149, 150]. The difference in the solution is shown in figure 5.6. This



(a) concentration at $t = 0.25$

(b) difference between the numerical and exact solutions at $t = 0.25$

Figure 5.3: Numerical results for LBM: Concentration and error in concentration are shown over the computational domain. These results correspond to Case 4 in Table 5.1.

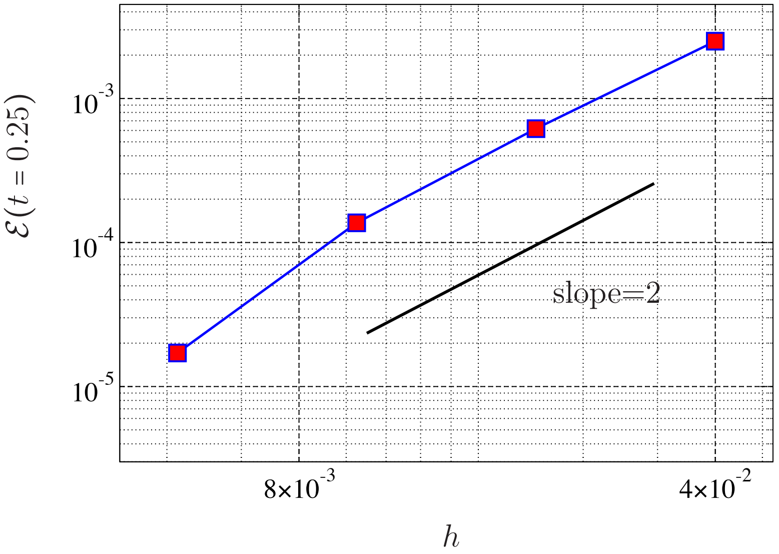
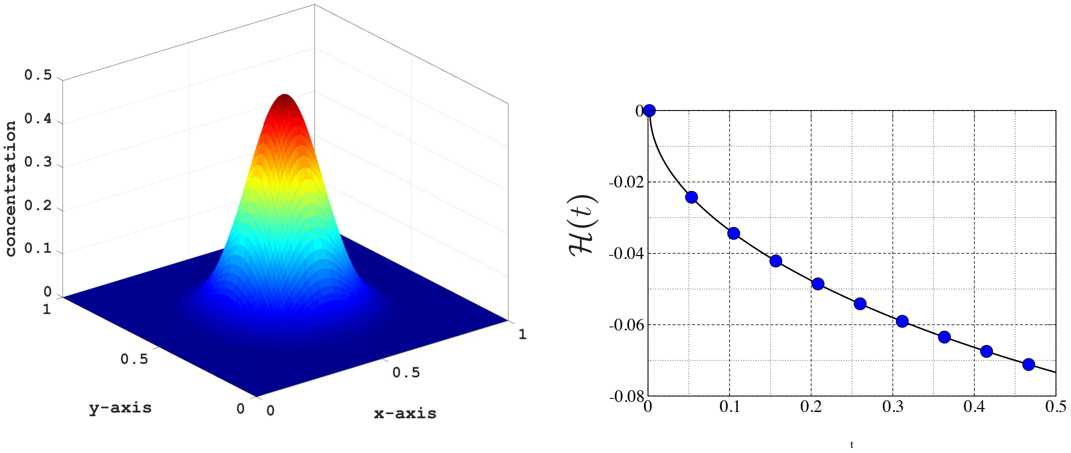


Figure 5.4: Numerical results for LBM: In this figure, the error in the numerical solution is shown against the lattice cell size.

difference should not be taken as a drawback of lattice Boltzmann method. It is in fact one of the advantages of kinetics-based methods over continuum-based methods. Extra information on the nature of interaction of particles with the boundary can be included in the numerical model. A continuum-based method may not be able to account for such details.



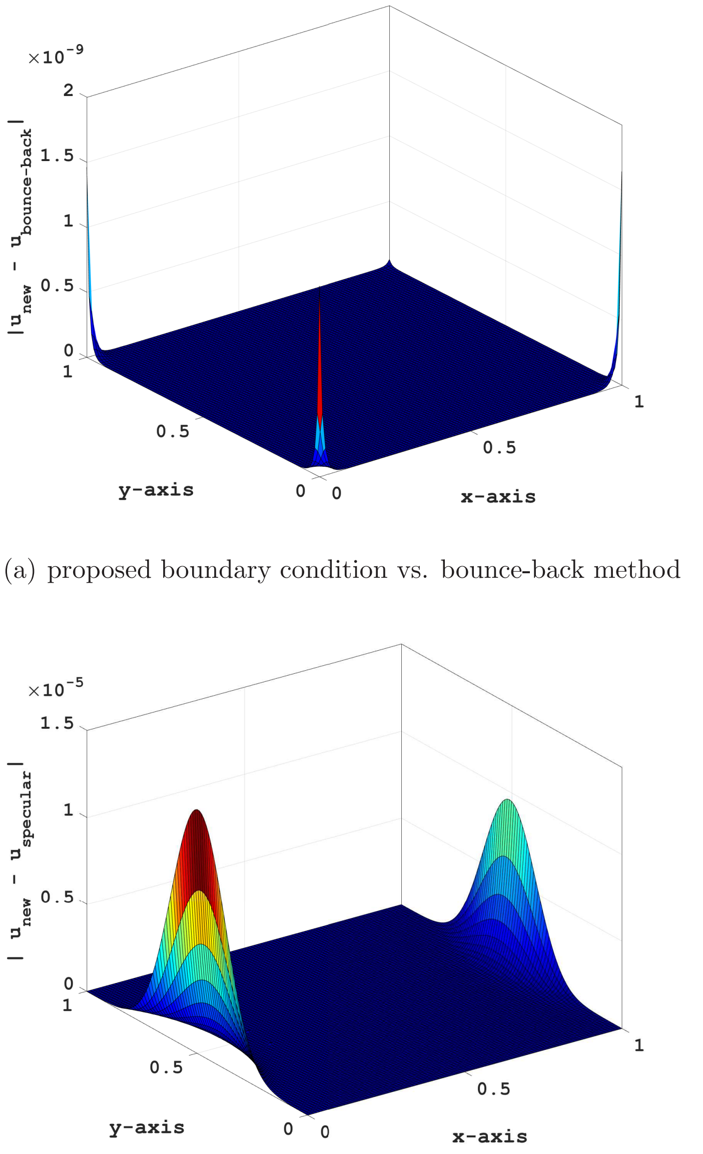
(a) Concentration at $t = 0.5$

(b) Boltzmann's $\mathcal{H}$ function

Figure 5.5: Numerical results for LBM: This figure presents sample numerical results under the lattice Boltzmann method with the new boundary conditions.

5.5 An Overlapping Domain Decomposition Method

Domain decomposition methods are powerful methods for obtaining numerical solutions for partial differential equations [7, 131]. These methods are particularly effective in a parallel computing setting. The basic idea is to split the computational domain into an arbitrary number of subdomains and seek the numerical solution in different subdomains separately. These subdomains can be *overlapping* or *non-overlapping*. In a non-overlapping domain decomposition scheme, one needs to account for an interface equation to enforce compatibility of numerical solutions near the interface between subdomains. Two of the more popular methods for constructing interface compatibility conditions are Lagrange multiplier framework, Steklov-Poincaré framework [131]. Introduction of such an interface condition may lead to higher complexity in the algorithm design but is also shown to give accurate nu-



(a) proposed boundary condition vs. bounce-back method

(b) proposed boundary condition vs. specular reflection method

Figure 5.6: Numerical results for LBM: In this figure, the difference between the numerical solution due to different treatment of zero-flux boundary is shown.

merical solutions. Overlapping domain decomposition do not require addition of a new interface constraint equation. In the proposed hybrid coupling method, we shall employ the overlapping domain decomposition approach. We now describe the iterative Schwartz method for numerical solution of a partial differential equation in an overlapping domain decomposition scheme.

Consider a domain Ω with boundary $\Gamma = \partial\Omega$. Consider the following equation defined on this domain as

$$\mathcal{L}[\mathrm{u}] = \mathrm{f} \quad \text{in } \Omega \text{ and} \tag{5.31a}$$

$$\mathrm{u} = \mathrm{u}^{\mathrm{p}} \qquad \text{on } \Gamma\cdot \tag{5.31b}$$

For simplicity, we assume that the boundary condition is purely Dirichlet, and employ two overlapping subdomains Ω_1 and Ω_2 (i.e., $\Omega_1 \cap \Omega_2 \neq \varnothing$ and $\Omega_1 \cup \Omega_2 = \Omega$). The governing partial differential equations in each subdomain will be as

$$\begin{cases} \mathcal{L}[\mathrm{u}_1] = \mathrm{f}_1 & \text{in } \Omega_1 \\ \mathrm{u}_1 = \mathrm{u}_2 & \text{on } \Gamma_1 \cap \Omega_2 \\ \mathrm{u}_1 = \mathrm{u}^{\mathrm{p}} & \text{on } \Gamma_1 \cap \Gamma \end{cases}, \text{ and } \begin{cases} \mathcal{L}[\mathrm{u}_2] = \mathrm{f}_2 & \text{in } \Omega_2 \\ \mathrm{u}_2 = \mathrm{u}_1 & \text{on } \Gamma_2 \cap \Omega_1 \\ \mathrm{u}_2 = \mathrm{u}^{\mathrm{p}} & \text{on } \Gamma_2 \cap \Gamma \end{cases}, \tag{5.32}$$

where the subindex is used to show the restriction of that quantity to the respective subdomain. The numerical solution to the system given in (5.32) can be found to be

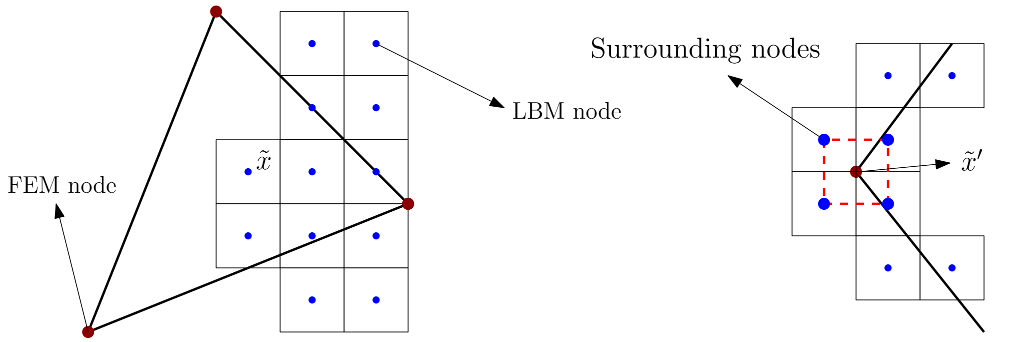
$$\begin{cases} \tilde{\mathcal{L}}[u_1^k] = f_1 & \text{in } \Omega_1 \\ u_1^k = u_2^{k-1} & \text{on } \Gamma_1 \cap \Omega_2 \\ u_1^k = \mathrm{u}^{\mathrm{p}} & \text{on } \Gamma_1 \cap \Gamma \end{cases}, \text{ and } \begin{cases} \tilde{\mathcal{L}}[u_2^{k+1}] = f_2 & \text{in } \Omega_2 \\ u_2^{k+1} = u_1^k & \text{on } \Gamma_2 \cap \Omega_1 \\ u_2^{k+1} = \mathrm{u}^{\mathrm{p}} & \text{on } \Gamma_2 \cap \Gamma \end{cases}, \tag{5.33}$$

where $\tilde{\mathcal{L}}$ is the discrete differential operator and super-indices $k-1$, k and $k+1$ are used to show consecutive iterations. The numerical solution in one subdomain, from the previous iteration, is used to determine the Dirichlet condition on the boundary of other subdomain. This approach can be extended to the case were more than two subdomains are involved.

The advantages of overlapping domain decomposition methods compared to non-overlapping methods are simpler algorithm design, increased flexibility in choice of numerical solver in different subdomains, and easy incorporation of non-matching grids and multi-time-stepping. In the following section, we will further scrutinize the methods for projecting data from a coarse mesh to a fine grid, and vice versa.

Transfer of information across non-matching grids

Typically, the grid-size for a coarse-scale simulation is much larger than the grid-size for a fine-scale simulation. Under lattice Boltzmann method, small cell-size can help accounting for complex spatial features of the computational domain in a fine-scale simulation. Upscaled (averaged) models for flow and transport in porous media such as Darcy's model do not need any such details of the pore structure, hence, the computational mesh for numerical solution of these models can be coarse. Under the proposed hybrid coupling method and the domain decomposition schemes introduced above, interaction among different subdomains occurs through the interface between any two subdomains. Transfer of information, consequently, needs to be done between non-matching grids that are disparate in size. This issue has been an active area of study in recent years. For instance, in simulation of fluid-structure interaction problems, traction at the interface of fluid and solid needs to be interpolated between



Transfer from FEM to LBM

Transfer from LBM to FEM

Figure 5.7: Transfer of information across non-matching grids: A pictorial description for interpolation across non-matching grids is provided.

non-matching grids [151–153]. In the context of overlapping domain decomposition schemes, numerical methods for flow and transport simulation on overlapping grids in [154, 155] and a study of stability of interpolation at the interface of subdomains in [156] can be mentioned. However, in this chapter, since we intend to use different numerical methods in different subdomains (i.e., FEM or LBM), the interpolation for concentration is not alike. Our approach to transfer the values of concentration at the interface and across non-matching grids is described next.

Consider a two-dimensional domain and let $\tilde{\boldsymbol{x}}$ be the coordinates of a cell lying on the boundary of a subdomain of LBM discretization. Then, the values of the concentration at this point can be approximated via the finite element interpolation on the element that contains the point $\tilde{\boldsymbol{x}}$. To approximate concentration at a point $\tilde{\boldsymbol{x}}'$ that lies on the boundary of the subdomain with FEM discretization, one needs to locate the surrounding cells (of the LBM solution). Hence, the point $\tilde{\boldsymbol{x}}'$ is inside the square patch with the surrounding LBM nodes at the corners. Concentration at this finite element node can be approximated using the values of concentration at the surrounding points of the square patch. For instance, one can use a four-node quadrilateral finite element interpolation (figure 5.7 provides a pictorial description). Obviously, three-dimensional cases can be handled similarly, however, the choice of interpolation function can be more varied (e.g., one can use interpolation functions over hexagonal or tetrahedral elements identified with surrounding LBM nodes).

To demonstrate this technique, consider the following function defined over domain $\Omega = (0,1) \times (0,1)$:

$$\mathrm{g}(\mathrm{x},\mathrm{y}) = \sin(2\pi\mathrm{x})\sin(2\pi\mathrm{y})\cdot \tag{5.34}$$

The coarse grid size will be denoted by h (linear three-node triangular elements used) and the fine grid size is shown by h'. The maximum error in the domain is denoted by $\mathcal{E}_{\max}$ and is defined by:

$$\mathcal{E}_{\max} = \max_{(x,y)} |\mathrm{g}(x,y) - g(x,y)|\,, \tag{5.35}$$

where (x,y) is a point on grid and $g(x,y)$ is the approximation of function g on a computational grid (either coarse or fine). Numerical results for transferring information across non-matching grids is given in Tables 5.2 and 5.3. From Table 5.2, we can conclude that the accuracy on the fine grid changes as $\mathcal{O}(h^2)$, which is expected, as it complies with the convergence rate of finite element approximation [157]. However, the error on the coarse grid, with information transferred to it from the fine domain, is $\mathcal{O}(h')$. Figures 5.8 and 5.9 show some demonstrative numerical results and outline the process given above. From this numerical experiment we conclude that, a bottleneck in convergence of the proposed coupling method can be the accuracy of fine to coarse grid information transfer.

Table 5.2: Transfer of information across non-matching grids: The numerical result for transfer of information from coarse grid to fine grid is given. The error is $\mathcal{O}(h^2)$, as expected.

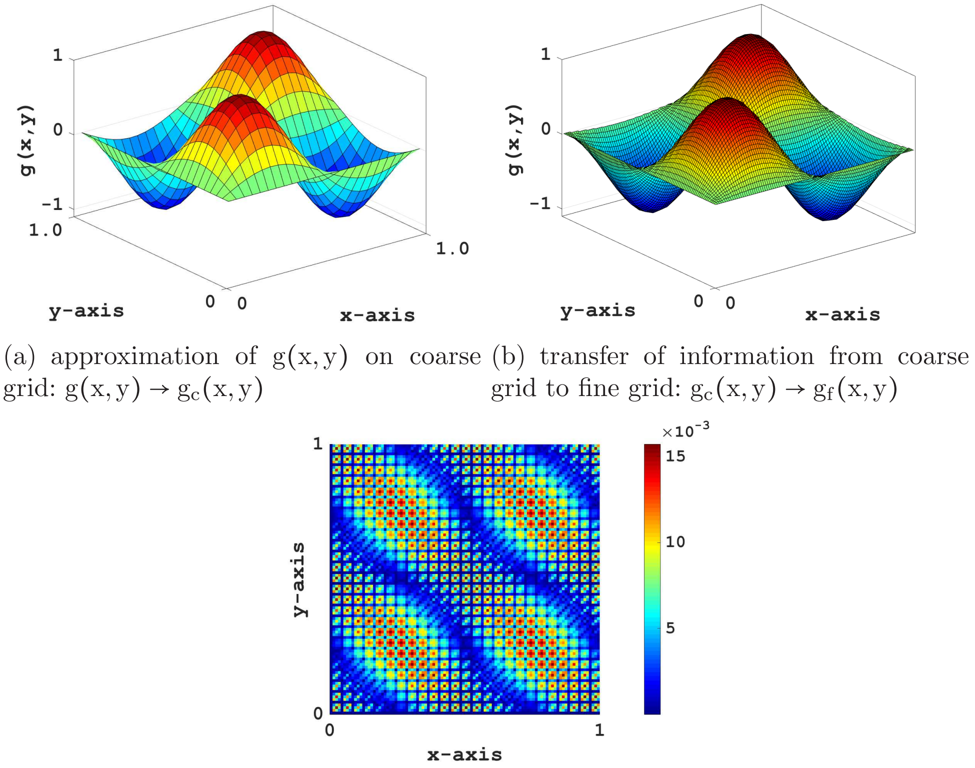
Case	h	h'	$\mathcal{E}_{\max}$
1	10^{-1}	10^{-2}	9.55×10^{-2}
2	4.0×10^{-2}	10^{-2}	1.57×10^{-2}
3	2.0×10^{-2}	10^{-2}	3.94×10^{-3}

Table 5.3: Transfer of information across non-matching grids: In this table, numerical values for transferring information from fine grid to coarse grid is presented. The error in the values of the coarse-grid approximation behaves as $\mathcal{O}(h')$.

Case	h	h'	$\mathcal{E}_{\max}$
1	4.0×10^{-2}	2.0×10^{-2}	1.13×10^{-1}
2	4.0×10^{-2}	10^{-2}	5.75×10^{-2}
3	4.0×10^{-2}	5.0×10^{-3}	2.90×10^{-2}

5.6 A New Hybrid Multi-Time-Step Coupling

In this section, we shall present a robust coupling method that allows hybrid modeling to be able to couple pore- and continuum-scale subdomains with disparate time-scales for solute transport in porous media. The spatial domain of interest Ω is partitioned into overlapping subdomains. The subdomains where *fine-scale* features of the solution are sought are denoted by Ω_{f}. Subdomains in which *coarse-scale* features are solved for are shown by Ω_{c}. Figure 5.10 provides a pictorial description of this partitioning scheme. In this chapter, we will employ a finite element discretization in subdomain Ω_{c}. This finite element method is applied to the equation (5.2). The fine-scale features in subdomain Ω_{f} are solved for using the lattice Boltzmann



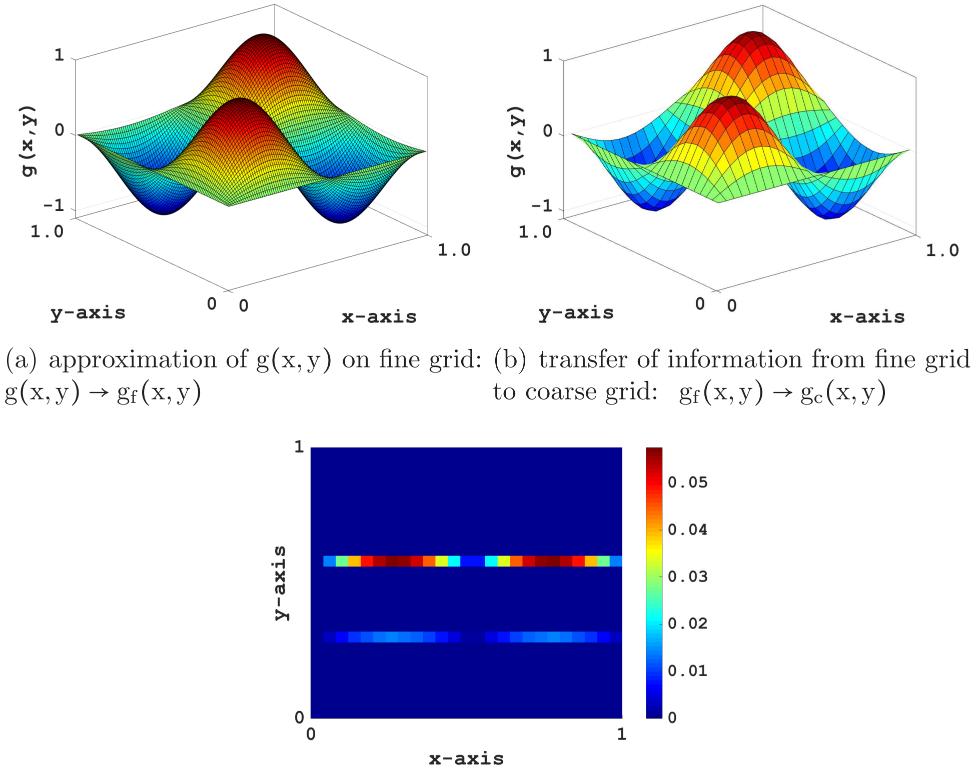
(a) approximation of g(x,y) on coarse grid: $g(x,y) \to g_c(x,y)$

(b) transfer of information from coarse grid to fine grid: $g_c(x,y) \to g_f(x,y)$

(c) error on the fine grid: $|g_f(x,y) - g(x,y)|$

Figure 5.8: Transfer of information across non-matching grids: The function $g(x,y)$ is first approximated on a coarse grid. The data corresponds to Case 2 in Table 5.2.

method, which solves the Boltzmann's transport equation in equation (5.9) in the mentioned region. Compatibility of the solutions is enforced using a Dirichlet condition at $\Gamma_{f\to c}$ and $\Gamma_{c\to f}$. The time- and space-continuous partial differential equations



(a) approximation of g(x, y) on fine grid: $g(x,y) \rightarrow g_f(x,y)$

(b) transfer of information from fine grid to coarse grid: $g_f(x,y) \rightarrow g_c(x,y)$

(c) error on coarse grid: $|g_c(x,y) - g(x,y)|$

Figure 5.9: Transfer of information across non-matching grids: The function g(x, y) is first approximated on a fine grid. Then, it is mapped onto a coarse grid using the method described in figure 5.7. The data corresponds to Case 2 in Table 5.3.

in each subdomain, along with their respective boundary conditions, are as

$$\text{in } \Omega_c : \left\{ \begin{array}{ll} \partial \mathrm{u}/\partial \mathrm{t} + \mathrm{div}\left[\mathbf{v}\mathrm{u} - \mathbf{D}\mathrm{grad}\left[\mathrm{u}\right]\right] = \mathrm{s} & (\mathbf{x}, \mathrm{t}) \in \Omega_c \times \mathcal{I} \\ \mathrm{u}(\mathbf{x}, \mathrm{t} = 0) = \mathrm{u}_0(\mathbf{x}) & \mathbf{x} \in \Omega_c \\ \mathrm{u}(\mathbf{x}, \mathrm{t}) = \mathrm{u}^{\mathrm{p}}(\mathbf{x}, \mathrm{t}) & (\mathbf{x}, \mathrm{t}) \in \Gamma^{\mathrm{D}} \times \mathcal{I} \\ (\mathbf{v}\mathrm{u} - \mathbf{D}\mathrm{grad}\left[\mathrm{u}\right]) \cdot \widehat{\mathbf{n}}(\mathbf{x}) = \mathrm{q}^{\mathrm{p}}(\mathbf{x}, \mathrm{t}) & (\mathbf{x}, t) \in \Gamma^{\mathrm{N}} \times \mathcal{I} \\ \textit{coupling condition}: \ \mathrm{u}(\mathbf{x}, \mathrm{t}) = \tilde{\mathrm{u}}_{\mathrm{f}\rightarrow\mathrm{c}}(\mathbf{x}, \mathrm{t}) & (\mathbf{x}, \mathrm{t}) \in \Gamma_{\mathrm{f}\rightarrow\mathrm{c}} \times \mathcal{I} \end{array} \right. , \tag{5.36}$$

and the governing equations in the fine-scale subdomain are

$$\text{in } \Omega_{\mathrm{f}}: \left\{ \begin{array}{ll} \partial f/\partial \mathrm{t} + \mathbf{v}\cdot \mathrm{grad}[f] = (f^{\mathrm{eq.}} - f)/\lambda & (\mathbf{x},\boldsymbol{\zeta},\mathrm{t}) \in \Omega_{\mathrm{f}} \times \mathbb{R}^{n} \times \mathcal{I} \\ f(\mathbf{x},\boldsymbol{\zeta},t=0) = f^{\mathrm{eq}}(\mathbf{x},\boldsymbol{\zeta},\mathrm{t}=0;\mathrm{u}_0(\mathbf{x}),\mathbf{v}) & (\mathbf{x},\boldsymbol{\zeta}) \in \Omega_{\mathrm{f}} \times \mathbb{R}^{n} \\ \int f \mathrm{d}\boldsymbol{\zeta} = \mathrm{u}^{\mathrm{p}}(\mathbf{x},\mathrm{t}) & (\mathbf{x},\mathrm{t}) \in \Gamma^{\mathrm{D}} \times \mathcal{I} \\ (\int f\boldsymbol{\zeta}\mathrm{d}\boldsymbol{\zeta})\cdot \widehat{\mathbf{n}}(\mathbf{x}) = \mathrm{q}^{\mathrm{p}}(\mathbf{x},\mathrm{t}) & (\mathbf{x},t) \in \Gamma^{\mathrm{N}} \times \mathcal{I} \\ \textit{coupling condition}: \int f \mathrm{d}\boldsymbol{\zeta} = \tilde{\mathrm{u}}_{\mathrm{c}\to\mathrm{f}}(\mathbf{x},\mathrm{t}) & (\mathbf{x},\mathrm{t}) \in \Gamma_{\mathrm{c}\to\mathrm{f}} \times \mathcal{I} \end{array} \right. . \tag{5.37}$$

This set of equations provides a basis to employ numerical methods of different origins in the same computational domain. The advection-diffusion equation is rooted in the continuum theory. The Boltzmann's equation however, is based on the kinetic theory. Using equation (5.36), one can solve for physical features at different temporal and spatial scales (macroscopic vs. mesoscopic), in a single computational framework. In the following, we will provide the temporal and spatial discretization of equation (5.36).

Space and time discretization

Coarse-scale problem

The coarse-scale problem is defined by equation (5.2), over domain Ω_{c} in Figure 5.10. We will use the semi-discrete methodology [139] , which gives the following time-continuous equation for the coarse-scale problem

$$\boldsymbol{M}\dot{\boldsymbol{u}} + \boldsymbol{K}\boldsymbol{u} = \boldsymbol{s}, \tag{5.38}$$

where $\boldsymbol{M}$ is the capacity matrix, $\boldsymbol{K}$ is the transport matrix and $\boldsymbol{u}$ is the nodal concentration. The superposed dot denotes the time derivative. The discretized right-hand-side of the finite element weak formulation is shown by $\boldsymbol{s}$. For time discretization, we will use the following notation

$$t^{(n)} = n\Delta t_{\mathrm{c}}, \quad \boldsymbol{u}(t^{(n)}) \approx \boldsymbol{d}^{(n)}, \text{ and } \quad \dot{\boldsymbol{u}}(t^{(n)}) \approx \boldsymbol{v}^{(n)}, \tag{5.39}$$

where Δt_{c} is the time-step used for integrating the coarse-scale problem. Using the trapezoidal method for time integration yields the following system of equations

$$\boldsymbol{M}\boldsymbol{v}^{(n+1)} + \boldsymbol{K}\boldsymbol{d}^{(n+1)} = \boldsymbol{s}^{(n+1)} \text{ and} \tag{5.40a}$$

$$\boldsymbol{d}^{(n+1)} = \boldsymbol{d}^{(n)} + \Delta t_{\mathrm{c}}(1-\vartheta)\boldsymbol{v}^{(n)} + \Delta t_{\mathrm{c}}\vartheta\boldsymbol{v}^{(n+1)}, \tag{5.40b}$$

where $\vartheta \in [0,1]$ is the time-integration parameter [158]. In this chapter, we will use $\vartheta = 1/2$, which gives a second-order accurate and unconditionally stable time-integrator (the midpoint rule). Once the the value of flux $\tilde{\mathrm{u}}_{\mathrm{c}\to\mathrm{f}}$ is known, the values for nodal concentrations $\boldsymbol{d}^{(n+1)}$ and the rate variable $\boldsymbol{v}^{(n+1)}$ can be found. In the following section we will briefly overview the discretization of the fine-scale problem.

Fine-scale problem

Our objective is to solve for the distribution of particles in the phase space defined by $\Omega_{\mathrm{f}} \times \mathbb{R}^n$. This goal can be achieved by solving the Boltzmann equation (5.9) numerically. The lattice Boltzmann method, introduced in Section 5.4, can provide relevant numerical results for simulation of the advection-diffusion process.

Consider a uniform grid, with the spacing between the cells equal to h_{f}, defined over the domain Ω_{f}. We will denote the time-step for the fine-scale problem by Δt_{f}, and the ratio $\eta = \Delta t_{\mathrm{c}}/\Delta t_{\mathrm{f}}$. The procedure for updating the discrete populations over a time-step is the same as what was outlined earlier in Section 5.4. In the following section, we will describe the new computational framework in detail and point out the transfer of data from fine-scale to coarse-scale domain.

The proposed hybrid computational framework

Before providing a step-by-step procedure for a numerical simulation using the proposed framework, we need to introduce a set of tools that will be useful. These tools will enable multi-time-step integration and information transfer across non-matching grids. The details are as below:

(i) Initializing the discrete unknowns: In Ω_{c} we utilize a finite element discretization. The nodal concentrations $\boldsymbol{d}$ can be simply initialized according to $\mathrm{u}_0(\mathbf{x})$. In Ω_{f} however, we assume that for the given initial concentration, the discrete populations f_i are given as

$$f_i(\boldsymbol{x}, t = 0) = f_i^{\mathrm{eq}}(\boldsymbol{x}, t = 0; \mathrm{u}_0(\boldsymbol{x}))\cdot \tag{5.41}$$

Other methods for initializing the discrete populations can also be considered.

(ii) Information transfer across the interface: To identify values of prescribed concentration on interfaces $\Gamma_{\mathrm{f}\to\mathrm{c}}$ and $\Gamma_{\mathrm{c}\to\mathrm{f}}$, we need to approximate the concentration at nodes lying on these boundaries. Figure 5.11 is an illustrative example of lattice and finite elements at the boundary of each subdomain. We will denote the coordinates of the point j, numbered in figure 5.11, as (x_j, y_j) and the concentration at that node as u_i. For given concentrations at nodes 1 to 4, the concentration at node 5, which belongs to a finite element in Ω_c, can be approximated as

$$u_5 \approx u_1(1-\gamma_x)(1-\gamma_y) + u_2\gamma_x(1-\gamma_y) + u_3\gamma_x\gamma_y + u_4(1-\gamma_x)\gamma_y, \qquad (5.42)$$

where $\gamma_x = (x_5 - x_1)/h_\mathrm{f}$ and $\gamma_y = (y_5 - y_1)/h_\mathrm{f}$. This method is obviously synonymous to approximation via a four-node quadrilateral element with its vertices lying on nodes 1 to 4. To transfer information from Ω_c to Ω_f, for instance at node number 1, one can use the finite element approximation in the element that includes the coordinates of node 1 (element i shown in figure 5.11). This value will serve as a Dirichlet-type condition on $\Gamma_{\mathrm{c}\to\mathrm{f}}$ and can be enforced using equation 5.22a. Three-dimensional cases can be handled similarly.

(iii) Multi-time-step integration: The solution in Ω_c advances in time with a time-step of Δt_c. This time-step is typically much larger than the time-step needed for fine-scale problem in subdomain Ω_f. However, to perform time-integration in Ω_f, we need to know the concentration on $\Gamma_{\mathrm{c}\to\mathrm{f}}$, which can only be determined by the numerical values in Ω_c. To approximate the concentration on $\Gamma_{\mathrm{c}\to\mathrm{f}}$ at intermediate time-level $\jmath$, between t and $t+\Delta t_\mathrm{c}$, we will use the following interpolation in time:

$$u_\mathrm{c}(\boldsymbol{x}, t + \frac{\jmath}{\eta}\Delta t_\mathrm{c}) \approx \left(\frac{\jmath}{\eta}\right) u_\mathrm{c}(\boldsymbol{x}, t + \Delta t_\mathrm{c}) + \left(1 - \frac{\jmath}{\eta}\right) u_\mathrm{c}(\boldsymbol{x}, t) \quad \boldsymbol{x} \in \Gamma_{\mathrm{c}\to\mathrm{f}}, \qquad (5.43)$$

where u_c is the concentration in subdomain Ω_c. Here, we have assumed that the rate of change in concentration remains constant in a time-step Δt_c. This value for $u_\mathrm{c}(\boldsymbol{x}, t + \frac{\jmath}{\eta}\Delta t_\mathrm{c})$ will be enforces as a Dirichlet condition on the solution in Ω_f.

(iv) Sub-iterations at each time-step: In order to ensure convergence of the proposed algorithm, one needs to transfer information between the subdomains iteratively. Compatibility of the numerical solutions from the pore and fine-scale problems at the overlap region $\Omega_{\mathrm{f}} \cap \Omega_{\mathrm{c}}$ has a vital role in accuracy of the numerical solution in the entire domain Ω. Figure 5.12 illustrates one iteration in a time-step Δt_{c}. The solution of the coarse-scale problem advances by Δt_{c} in step 1. Using the updated values of solution in Ω_{c}, boundary conditions onto subdomain Ω_{f} at intermediate time-levels is determined. The solution of the fine-scale domain advances by time-step Δt_{f} successively. The new numerical values are then used to find the concentrations on boundary $\Gamma_{\mathrm{f}\to\mathrm{c}}$, which is used to update the solution in the coarse-scale domain in the next iteration. This procedure is repeated an arbitrary number of times in order to satisfy accuracy requirements defined by the user.

Given the tools described above, one can implement the proposed coupling method in a systematic manner. A step-by-step procedure is given in Algorithm 1. In the following section, we will provide numerical examples to showcase the performance of this framework.

The case of many subdomains

Thus far, the proposed coupling algorithm is presented for the case of only two subdomains, a coarse-scale subdomain Ω_{c} and a fine-scale domain Ω_{f}. However, in practical applications decomposition into multiple subdomains may be required. In

Algorithm 1 Hybrid multi-time-step coupling framework: Outline of the algorithm for proposed framework.

```
Initialize $\boldsymbol{u}$ in $\Omega_\mathrm{c}$ and $f_i$ in $\Omega_\mathrm{f}$ for $t = 0$.
Set $t \leftarrow 0$
while $t < \mathrm{T}$ do
    Set $t \leftarrow t + \Delta t$
    Set Iter $\leftarrow 0$.
    while Iter $\leq$ MaxIter do
        Set Iter $\leftarrow$ Iter $+ 1$.
        Find $\tilde{u}_{\mathrm{f}\to\mathrm{c}}$ defined on $\Gamma_{\mathrm{f}\to\mathrm{c}}$.
        Advance the solution in $\Omega_\mathrm{c}$ by $\Delta t_\mathrm{c}$.
        Find $\tilde{u}_{\mathrm{c}\to\mathrm{f}}$ defined on $\Gamma_{\mathrm{c}\to\mathrm{f}}$ at time-levels $t - \Delta t_c$ and $t$.
        Set $\jmath \leftarrow 0$
        while $\jmath \leq \eta$ do
            Set $\jmath \leftarrow \jmath + 1$
            Advance the solution in $\Omega_\mathrm{f}$ by $\Delta t_\mathrm{f}$ to find $f_i(\boldsymbol{x}, t + \jmath\Delta t_\mathrm{f})$ (stream and collide).
            Impose Dirichlet boundary condition on $\Gamma_{\mathrm{c}\to\mathrm{f}}$ with
                $\tilde{u}_{\mathrm{c}\to\mathrm{f}}(\boldsymbol{x}, t + \jmath\Delta t_\mathrm{f}) = (1 - \jmath/\eta)\tilde{u}_{\mathrm{c}\to\mathrm{f}}(\boldsymbol{x}, t - \Delta t_\mathrm{c}) + (\jmath/\eta)\tilde{u}_{\mathrm{c}\to\mathrm{f}}(\boldsymbol{x}, t)$.
        end while
        From the new numerical values in $\Omega_\mathrm{f}$ find $\tilde{u}_{\mathrm{f}\to\mathrm{c}}$.
    end while
end while
```

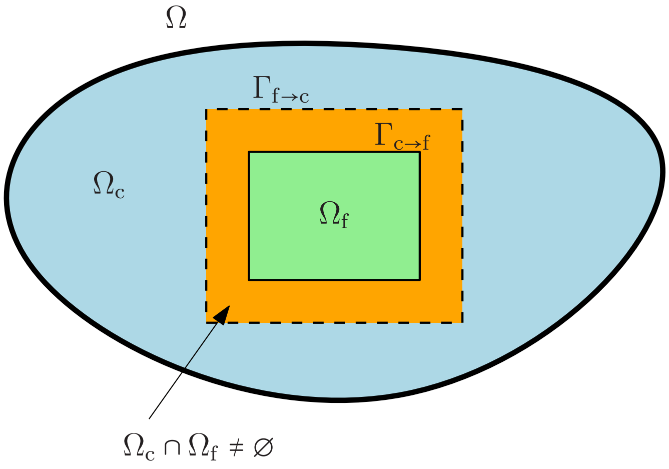

Ω
$\Gamma_{\mathrm{f}\to\mathrm{c}}$
$\Gamma_{\mathrm{c}\to\mathrm{f}}$
Ω_c
Ω_f
$\Omega_\mathrm{c} \cap \Omega_\mathrm{f} \neq \varnothing$

Figure 5.10: Overlapping domain partitioning: The proposed coupling method employs overlapping domain partitioning. This figure illustrates the decomposition of the computational domain Ω into the subdomains where coarse-scale and fine-scale features are sought after.

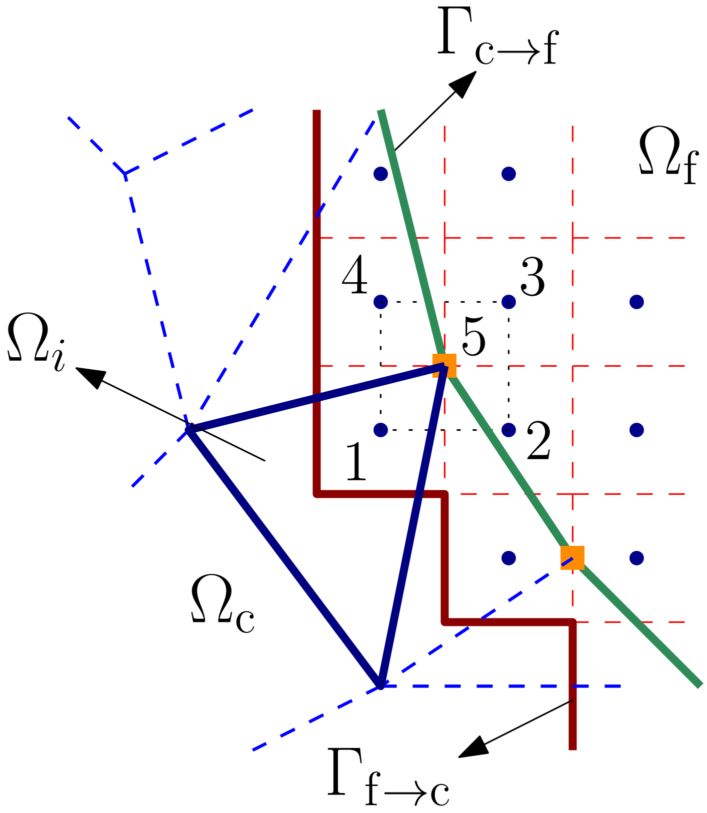


Figure 5.11: Information transfer across non-matching grids: In this figure, transfer of information at the interface of subdomains is depicted.

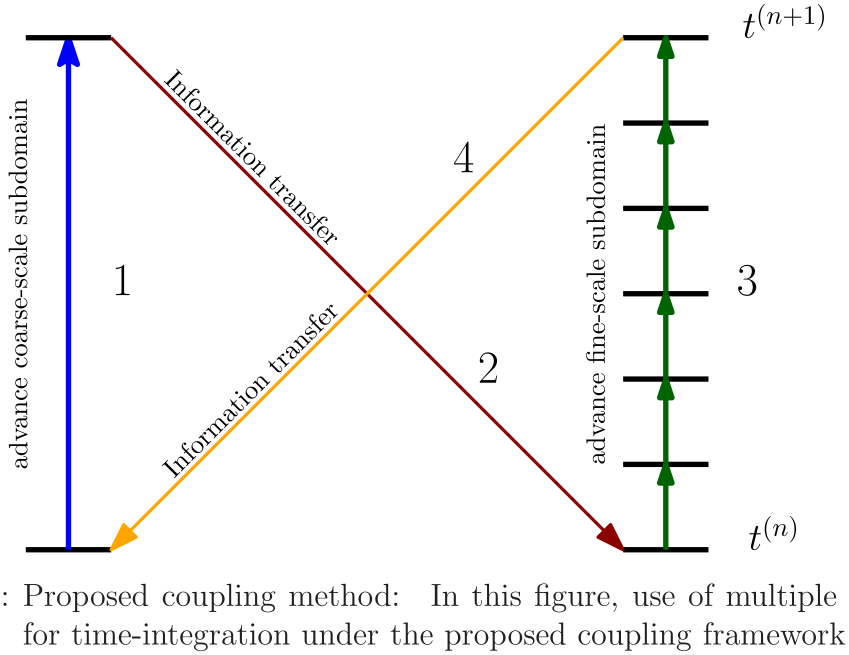


Figure 5.12: Proposed coupling method: In this figure, use of multiple time-steps for time-integration under the proposed coupling framework is demonstrated.

this section, we will present the proposed coupling method for cases where there are multiple coarse and fine-scale subdomains.

Suppose that the domain $\Omega \subset \mathbb{R}^n$ is partitioned into coarse and fine-scale subdomains, given as

$$\Omega = \underbrace{\left(\bigcup_{i=1}^{N_\mathrm{c}} \Omega_{\mathrm{c},i}\right)}_{\text{coarse–scale subdomains}} \bigcup \underbrace{\left(\bigcup_{j=1}^{N_\mathrm{f}} \Omega_{\mathrm{f},j}\right)}_{\text{fine–scale subdomains}}, \tag{5.44}$$

where all coarse and fine-scale subdomains are overlapping. The number of coarse-scale subdomains is shown by N_c and N_f is the number of fine-scale subdomains. Each subdomain $\Omega_{\mathrm{f},j}$ (for $j = 1, \cdots, N_\mathrm{f}$) is a fine-scale subdomain and will be integrated using the lattice Boltzmann method with grid size $h_{\mathrm{f},j}$ and time-step Δt_j. Coarse-scale subdomains $\Omega_{\mathrm{f},i}$ (for $i = 1, \cdots, N_\mathrm{c}$) are solved using the finite element method with mesh-size $h_{\mathrm{c},i}$ and $\Delta t_{\mathrm{c},i}$. The details regarding multi-time-stepping and transferring data from coarse-scale grid to fine-scale grid (and vise versa) remains the same as before. Since discretization parameters for coarse-scale domains are much larger than the ones used in the fine-scale subdomains, the solution in coarse-scale domains advances first, then the updated values near the interface of coarse-scale/fine-scale subdomains are used for multi-time-step integration. Obviously, even coarse-scale subdomains can be integrated with different time-steps. Multi-time-step integration for the coarse-scale subdomains can be done in the same spirit as for the coarse-scale subdomains presented earlier. However, an alternative approach would be to use the method presented in [127] to solve the coarse-scale subdomains (that share an interface), and then use the updated solution to transfer to fine-scale domains. We will not follow this procedure here, but it can be explored in future research endeavors. We will denote the system time-step, the same definition used in [127], by Δt. The proposed coupling framework for the case of multiple subdomains is given in Algorithm 2.

Algorithm 2 Hybrid multi-time-step coupling framework for many subdomains: The algorithmic procedure for the proposed framework is outlined.

```
Set $t \leftarrow 0$

while $t + \Delta t < T$ do

    for Iter = $1, \cdots$ do

        for $i = 1, \cdots, N_c$ do

            Advance the solution in subdomain $\Omega_{c,i}$ by one system time-step, subject to boundary values from the solutions from previous iteration.

        end for
        for $j = 1, \cdots, N_f$ do

            Advance the solution in subdomain $\Omega_{f,j}$ by one system time-step, subject to boundary values approximated by equation (5.43).
        end for
    end for
end while
```

5.7 Representative Numerical Results

In this section, we will apply the proposed coupling algorithm to one- and two-dimensional problems. The performance of the new method with respect to discretization parameters will be studied. Computer implementation is done using NumPy [159] and FEniCS [160] software packages.

Advection and diffusion of one-dimensional Gaussian hill

Consider $\Omega = (0,1)$ with zero-flux condition imposed on both ends. The initial concentration is given as

$$\mathrm{u}_0(\mathrm{x}) = \frac{\phi}{\sqrt{2\pi\sigma_0^2}} e^{-(\mathrm{x}-\mathrm{x}_0)^2/2\sigma_0^2}, \tag{5.45}$$

where $\phi = 10^{-1}$ and $\sigma_0 = 10^{-2}$. The initial location of the tip of the Gaussian hill is at x_0 and is set to be 3×10^{-1}. The advection velocity in the entire domain is taken to be $\mathrm{v} = 1$ and the diffusion coefficient is $\mathrm{D} = 10^{-2}$. The source term is taken to be zero and the time-interval of interest is $\mathrm{T} = 4 \times 10^{-2}$. We will use the proposed hybrid coupling method to numerically solve this problem. We will use the finite element method with the Galerkin formulation in Ω_c and the lattice Boltzmann method in Ω_f. To showcase the performance of the proposed method, we will use the following definition for error (error in ∞-norm)

$$\mathcal{E}(t) = \max_{i=1,\cdots,N} \left| u(\boldsymbol{x}_i, t) - \mathrm{u}_{\mathrm{exact}}(\boldsymbol{x_i}, t) \right|, \tag{5.46}$$

where N is the number of nodes for numerical solution, $u(\boldsymbol{x}, t)$ is the approximate solution at point $\boldsymbol{x}_i$ and time t. The exact solution is represented by $\mathrm{u}_{\mathrm{exact}}$. Following the definition given in (5.46), the error in Ω_c and Ω_f will be denoted by $\mathcal{E}_\mathrm{c}$ and $\mathcal{E}_\mathrm{f}$ respectively. We will denote the length of the overlap region $\Omega_\mathrm{f} \cap \Omega_\mathrm{c}$ by L_{overlap}. The domain partitioning is as

$$\Omega_\mathrm{c} = \left(0, \frac{1}{2} + \frac{L_{\mathrm{overlap}}}{2}\right), \text{ and } \quad \Omega_\mathrm{f} = \left(\frac{1}{2} - \frac{L_{\mathrm{overlap}}}{2}, 1\right) \cdot \tag{5.47}$$

We will employ two-node linear finite elements of equal lengths h_c to discretize Ω_c. The time-step is set to be $\Delta t_\mathrm{c} = h_\mathrm{c}^2/2\mathrm{D}$. Subdomain Ω_f is discretized using a uniform grid with spacing h_f and a time-step of $\Delta t_\mathrm{f} = h_\mathrm{f}^2/2\mathrm{D}$. The $D1Q2$ lattice model will be used in Ω_f. The number of sub-iterations in each time-step is shown by MaxIter.

Figure 5.13 shows a comparison between the numerical solution from the hybrid coupling framework and the exact solution. The concentration profile is shown when the front is passing through the overlap region and afterwards. In both cases, the numerical solution is in accordance with the exact solution.

The numerical experiments discussed here show that the proposed hybrid coupling framework gives an accurate solution to the advection-diffusion equation and is indeed a converging scheme (see figures 5.14 and 5.15). From these numerical experiments, we conclude that the convergence of the numerical solution under the proposed framework is $\mathcal{O}(h)$. In the following, the effect of discretization in coarse and fine-scale subdomains, effect of length of overlap region and the number of sub-iterations on the accuracy of the numerical solution are described.

(1) Discretization in fine-scale domain: Our numerical experiments indicate that for a given discretization in the coarse-scale domain (i.e., h_c and Δt_c), refinement of parameters h_f and Δt_f improves the overall accuracy of numerical solution. The results presented in Table 5.4 show that the mentioned refinement reduces the error in both fine-scale and coarse-scale subdomains.

(2) Discretization in coarse-scale domain: Considering the numerical results presented in Table 5.5, one can conclude that for a given discretization in fine-scale domain (i.e., h_f and Δt_f), refinement of respective parameters in the coarse-scale domain does not necessarily improve accuracy. This behavior can be attributed

to the fact that the lower accuracy in the fine-scale domain (due to use of lattice Boltzmann method), results in a less accurate estimation of the concentration on $\Gamma_{\mathrm{f}\to\mathrm{c}}$. Hence, the numerical solution in the coarse-scale region converges to a solution other than the exact solution.

(3) Length of overlap region: For a given discretization in subdomains Ω_{f} and Ω_{c}, increase in the length of the overlap region results in reduction of overall accuracy. This conclusion can be drawn from the numerical experiments presented in Table 5.6. However, if the grid-size and time-step in both subdomains change simultaneously, convergence rate of the numerical solution to the exact solution may slow down. Following the numerical results given in Tables 5.7, 5.8 and 5.9, as well as figures 5.15, 5.16 and 5.17 shows that convergence under simultaneous refinement in both subdomains has an inverse relation to the length of the overlap region.

(4) Number of sub-iterations in each time-step: In the numerical experiments performed, increasing the maximum number of sub-iterations to values greater than 4 did not result in a significant improvement in accuracy. However, compatibility, especially near the overlap region, can be improved by increasing the number of sub-iterations.

(5) Order of interpolation in the coarse-scale subdomain: Figure 5.18 shows the point-wise error in the coarse-scale subdomain, for different orders of interpolation in finite elements and under multi-time-stepping. For different cases, the error in the fine-scale subdomain remains largely unchanged from one case to

another. Error in the coarse-scale subdomain decreases by increasing the order of interpolation, however, the error near the overlap region remains unchanged. The figure the error accumulates near the overlapping region under both multi-time-stepping and under single time-step in all the subdomains.

These observations regarding the effect of number of sub-iterations and length of the overlap region are in accordance with the theory of overlapping domain decomposition methods [161, 162]. It seems that, generally, decrease in the size of the overlap region reduces the rate of convergence and the error decreases proportional to the inverse of square root of number of sub-iterations. One key observation from these numerical experiments is that majority of error in the numerical solution accumulates near the overlap region. This error can be much higher than the error in the rest of the domain and refinement in either of the subdomains may not improve it. Hence, a topic for future research can be designing efficient methods for removing the accumulated error in the overlap region under the proposed coupling framework.

Here, we showed that one can use highly disparate mesh-size and time-steps in different subdomains. Furthermore, we showed that to improve accuracy throughout the computational domain, grid refinement in the fine-scale domain is sufficient. We also demonstrated that mesh refinement only in the coarse-scale domain may not lead to a more accurate numerical solution.

Table 5.4: Advection and diffusion of one-dimensional Gaussian hill: In this table, the accuracy of the numerical solution using the proposed coupling framework is shown. Here, only cell size and time-step in the fine-scale domain are refined. Note that despite the refinement in the fine-scale domain only, the accuracy of the solution in the entire computational domain is improving.

h_c	Δt_c	h_f	Δt_f	$\mathcal{E}_\mathrm{c}(\mathrm{T})$	$\mathcal{E}_\mathrm{f}(\mathrm{T})$
10^{-2}	5.00×10^{-3}	5.00×10^{-3}	1.25×10^{-3}	3.67×10^{-3}	1.70×10^{-2}
10^{-2}	5.00×10^{-3}	2.50×10^{-3}	3.13×10^{-4}	1.94×10^{-3}	7.42×10^{-3}
10^{-2}	5.00×10^{-3}	1.25×10^{-3}	7.81×10^{-5}	1.02×10^{-3}	3.48×10^{-3}
10^{-2}	5.00×10^{-3}	6.25×10^{-4}	1.95×10^{-5}	5.50×10^{-4}	1.80×10^{-3}

Table 5.5: Advection and diffusion of one-dimensional Gaussian hill: In this table, performance of the proposed method for numerical solution of the one-dimensional problem is shown. In this case, element size and time-step refinement are done only in the coarse-scale domain. The discretization parameters in the fine-scale domain remain unchanged in the fine-scale domain. It can be observed that refinement, merely in the coarse-scale domain, has adverse effect on the accuracy of numerical solution. This experiment shows that the numerical method with the slowest convergence has the dominant role in overall accuracy.

h_c	Δt_c	h_f	Δt_f	$\mathcal{E}_\mathrm{c}(\mathrm{T})$	$\mathcal{E}_\mathrm{f}(\mathrm{T})$
10^{-2}	5.00×10^{-3}	1.25×10^{-3}	7.81×10^{-5}	1.02×10^{-3}	3.48×10^{-3}
5.00×10^{-3}	1.25×10^{-3}	1.25×10^{-3}	7.81×10^{-5}	1.65×10^{-3}	3.71×10^{-3}
2.50×10^{-3}	3.13×10^{-4}	1.25×10^{-3}	7.81×10^{-5}	2.01×10^{-3}	3.74×10^{-3}
1.25×10^{-3}	7.81×10^{-5}	1.25×10^{-3}	7.81×10^{-5}	2.22×10^{-3}	3.74×10^{-3}

Table 5.6: Advection and diffusion of one-dimensional Gaussian hill: This numerical experiment indicates that increasing the length of the overlapping region could have adverse effect on the accuracy of the numerical solution.

h_c	Δt_c	h_f	Δt_f	L_overlap	$\mathcal{E}_\mathrm{c}(\mathrm{T})$	$\mathcal{E}_\mathrm{f}(\mathrm{T})$
10^{-2}	5.00×10^{-3}	1.25×10^{-3}	7.81×10^{-5}	2.00×10^{-2}	5.78×10^{-4}	3.08×10^{-3}
10^{-2}	5.00×10^{-3}	1.25×10^{-3}	7.81×10^{-5}	4.00×10^{-2}	5.85×10^{-4}	3.43×10^{-3}
10^{-2}	5.00×10^{-3}	1.25×10^{-3}	7.81×10^{-5}	8.00×10^{-2}	8.63×10^{-4}	3.47×10^{-3}
10^{-2}	5.00×10^{-3}	1.25×10^{-3}	7.81×10^{-5}	10^{-1}	1.02×10^{-3}	3.48×10^{-3}

Table 5.7: **Advection and diffusion of one-dimensional Gaussian hill**: In this table, values of discretization parameters and errors in each subdomain are provided. In all the cases, $\eta = 4$ and $L_{\text{overlap}} = 4 \times 10^{-2}$. The number of sub-iterations in each time-step is 10.

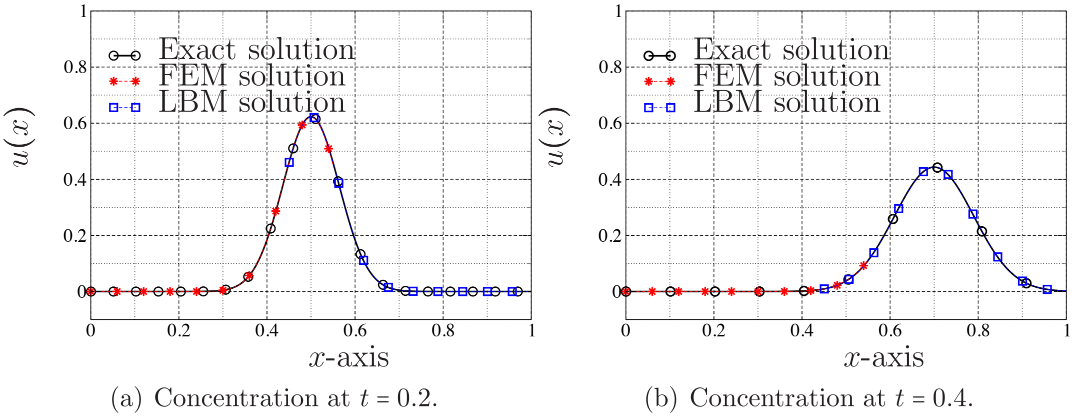
h_{c}	Δt_{c}	h_{f}	Δt_{f}	$\mathcal{E}_{\text{c}}(\text{T})$	$\mathcal{E}_{\text{f}}(\text{T})$
1.00×10^{-2}	5.00×10^{-3}	5.00×10^{-3}	1.25×10^{-3}	2.55×10^{-3}	1.64×10^{-2}
5.00×10^{-3}	1.25×10^{-3}	2.50×10^{-3}	3.13×10^{-4}	2.22×10^{-3}	7.54×10^{-3}
2.50×10^{-3}	3.13×10^{-4}	1.25×10^{-3}	7.81×10^{-5}	1.40×10^{-3}	3.70×10^{-3}
1.25×10^{-3}	7.81×10^{-5}	6.25×10^{-4}	1.95×10^{-5}	7.79×10^{-4}	1.85×10^{-3}
6.25×10^{-4}	1.95×10^{-5}	3.13×10^{-4}	4.88×10^{-6}	4.11×10^{-4}	1.75×10^{-3}
3.13×10^{-4}	4.88×10^{-6}	1.56×10^{-4}	1.22×10^{-6}	2.11×10^{-4}	1.73×10^{-3}

Table 5.8: **Advection and diffusion of one-dimensional Gaussian hill**: Values for the discretization parameters and errors in fine and coarse-scale subdomains are given. In all cases, $\eta = 4$, $L_{\text{overlap}} = 10^{-2}$ and the number of sub-iterations is 10.

h_{c}	Δt_{c}	h_{f}	Δt_{f}	$\mathcal{E}_{\text{c}}(\text{T})$	$\mathcal{E}_{\text{f}}(\text{T})$
5.00×10^{-3}	1.25×10^{-2}	2.50×10^{-3}	3.13×10^{-4}	3.32×10^{-3}	4.09×10^{-3}
2.50×10^{-3}	3.13×10^{-4}	1.25×10^{-3}	7.81×10^{-5}	2.07×10^{-3}	2.12×10^{-3}
1.25×10^{-3}	7.81×10^{-5}	6.25×10^{-4}	1.95×10^{-5}	1.14×10^{-3}	1.72×10^{-3}
6.25×10^{-4}	1.95×10^{-5}	3.13×10^{-4}	4.88×10^{-6}	6.00×10^{-4}	1.72×10^{-3}

Table 5.9: **Advection and diffusion of one-dimensional Gaussian hill**: Discretization and errors in fine-scale and coarse-scale domains are given in this table. The number of sub-iterations in each time-step is 10. In all the cases, $\eta = 4$ and $L_{\text{overlap}} = 10^{-1}$.

h_{c}	Δt_{c}	h_{f}	Δt_{f}	$\mathcal{E}_{\text{c}}(\text{T})$	$\mathcal{E}_{\text{f}}(\text{T})$
1.00×10^{-2}	5.00×10^{-3}	5.00×10^{-3}	1.25×10^{-3}	3.67×10^{-3}	1.70×10^{-2}
5.00×10^{-3}	1.25×10^{-3}	2.50×10^{-3}	3.13×10^{-4}	3.16×10^{-3}	7.67×10^{-3}
2.50×10^{-3}	3.13×10^{-4}	1.25×10^{-3}	7.81×10^{-5}	2.01×10^{-3}	3.74×10^{-3}
1.25×10^{-3}	7.81×10^{-5}	6.25×10^{-4}	1.95×10^{-5}	1.13×10^{-3}	1.86×10^{-3}
6.25×10^{-4}	1.95×10^{-5}	3.13×10^{-4}	4.88×10^{-6}	6.02×10^{-4}	1.73×10^{-3}



(a) Concentration at $t = 0.2$.

(b) Concentration at $t = 0.4$.

Figure 5.13: Advection and diffusion of one-dimensional Gaussian hill: This figure shows the exact and numerical concentration profiles at two different time-levels. At $t = 0.2$, the front passes through the overlap region. The numerical solution shows good agreement with the exact solution.

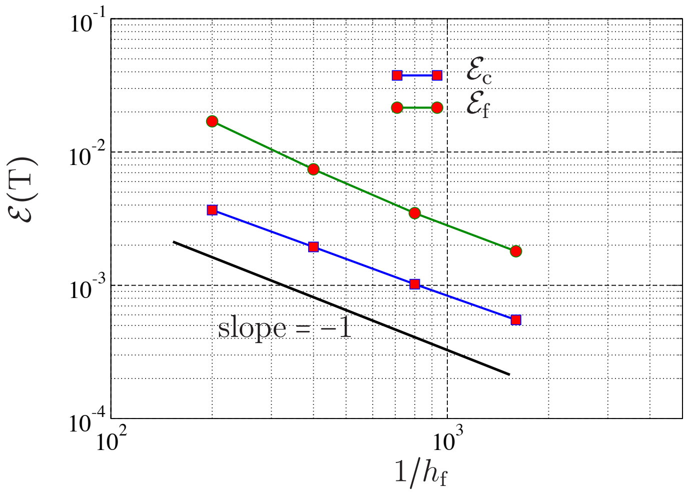


Figure 5.14: Advection and diffusion of one-dimensional Gaussian hill: This figure shows the error in the coarse-scale and fine-scale subdomains against refinement in the fine-scale region. Table 5.4 provides the numerical values employed in this numerical simulation.

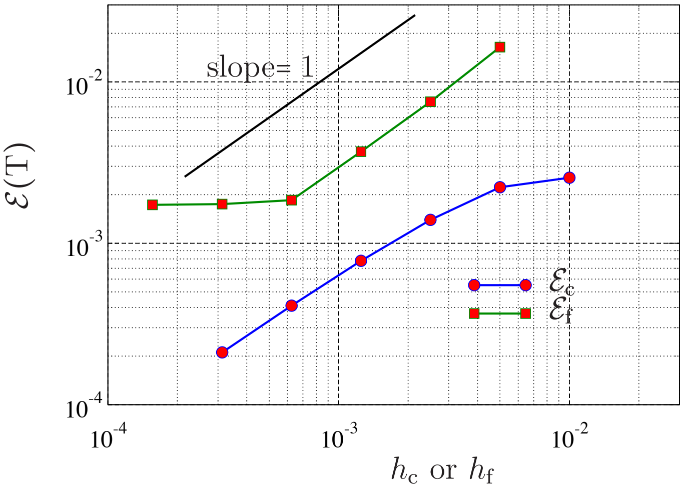


Figure 5.15: Advection and diffusion of one-dimensional Gaussian hill: The error in the fine and coarse-scale subdomains is plotted against grid-size. In all cases $h_\mathrm{c} = 2h_\mathrm{f}$.

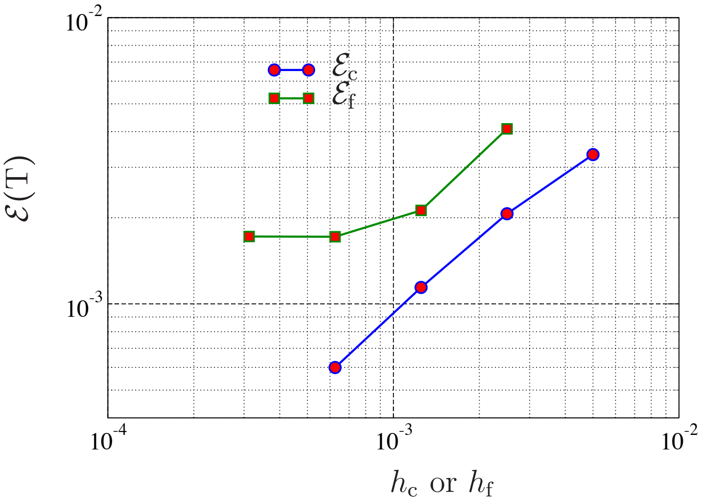


Figure 5.16: Advection and diffusion of one-dimensional Gaussian hill: In this figure, the error in the coarse- and fine-scale subdomains is shown. In this case the length of the overlap region is $L_\mathrm{overlap} = 10^{-2}$.

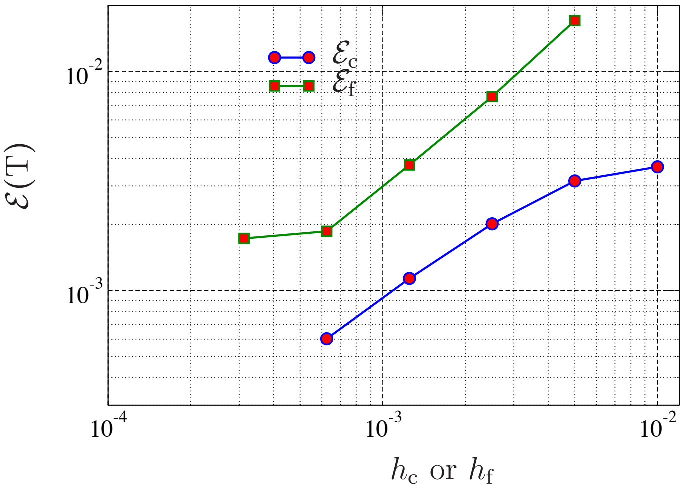


Figure 5.17: Advection and diffusion of one-dimensional Gaussian hill: Error in the fine and coarse-scale domains with respect to mesh size in each subdomain is shown. Here, the length of the overlap region is $L_{\mathrm{overlap}} = 10^{-1}$.

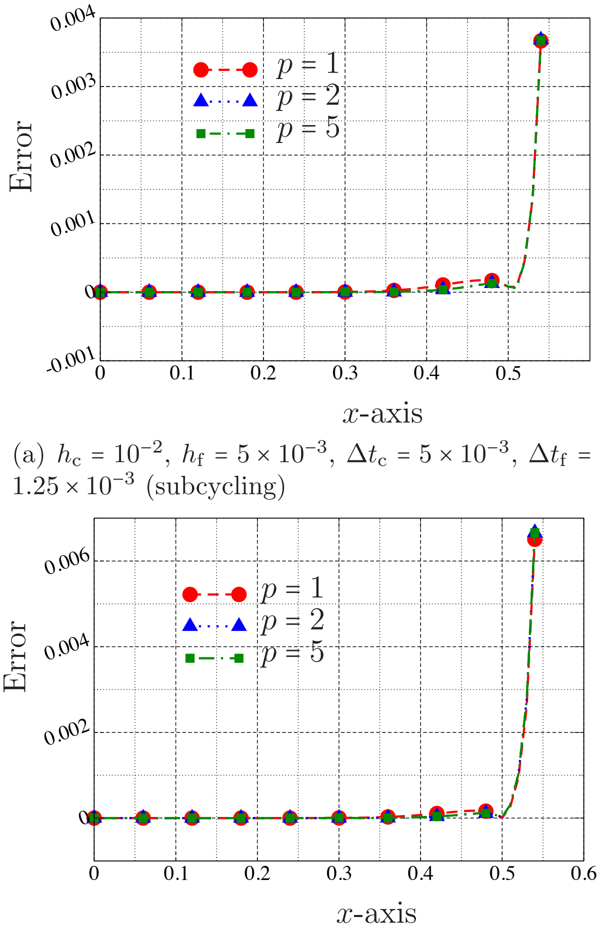
Simulation of fast bimolecular reaction using multiple subdomains

This example will be used to demonstrate the application of the proposed hybrid framework for bimolecular fast reactions and its ability to handle multiple subdomains. To this end, we simulate the evolution of the concentrations of the participating chemical species in the following bimolecular reaction

$$\mathrm{n_A A + n_B B \rightarrow n_C C}, \tag{5.48}$$

where $\mathrm{n_A}$, $\mathrm{n_B}$ and $\mathrm{n_C}$ are the stoichiometry coefficients. Here, we have chosen $\mathrm{n_A} = 1$, $\mathrm{n_B} = 2$ and $\mathrm{n_C} = 1$. The computational domain $\Omega = (0, 1)$ is partitioned into the following two coarse-scale and one fine-scale subdomains:

$$\Omega_{\mathrm{c},1} = (0, 0.40), \ \Omega_{\mathrm{f}} = (0.39, 0.61) \quad \text{and} \quad \Omega_{\mathrm{c},2} = (0.6, 1.0)\cdot$$



(a) $h_{\mathrm{c}} = 10^{-2}$, $h_{\mathrm{f}} = 5 \times 10^{-3}$, $\Delta t_{\mathrm{c}} = 5 \times 10^{-3}$, $\Delta t_{\mathrm{f}} = 1.25 \times 10^{-3}$ (subcycling)

(b) $h_{\mathrm{c}} = h_{\mathrm{f}} = 10^{-2}$, $\Delta t_{\mathrm{c}} = \Delta t_{\mathrm{f}} = 5 \times 10^{-3}$ (no subcycling)

Figure 5.18: Advection and diffusion of one-dimensional Gaussian hill: In this figure, point-wise error in the coarse-scale domain at time $t = \mathrm{T}$ is shown. Different orders of interpolation (denoted by p here) in the finite elements are used.

The time-interval of interest is $\mathrm{T} = 0.5$. The coefficient of diffusion is $\mathrm{D} = 10^{-2}$ and the advection velocity is zero throughout the domain. We will enforce zero-flux boundary conditions at $\mathrm{x} = 0$ and $\mathrm{x} = 1$. The initial values for each of the species is as

$$\mathrm{u}_{0,i}(\mathrm{x}) = \frac{\phi_{0,i}}{\sqrt{2\pi\sigma^2}} \exp\left[-(\mathrm{x} - \mathrm{x}_{0,i})^2/2\sigma^2\right] \quad i = \mathrm{A}, \mathrm{B}, \mathrm{C}, \tag{5.49}$$

where $\sigma = 0.1$, $\phi_{0,\mathrm{A}} = 0.1$, $\mathrm{x}_{0,\mathrm{A}} = 0.3$, $\phi_{0,\mathrm{B}} = 0.05$ and $\mathrm{x}_{0,\mathrm{B}} = 0.7$. The initial concentration of the species C is zero in the entire domain. To solve the problem numerically, it is convenient to introduce the invariants

$$\alpha = \mathrm{u}_{\mathrm{A}} + \frac{\mathrm{n}_{\mathrm{A}}}{\mathrm{n}_{\mathrm{C}}}\mathrm{u}_{\mathrm{C}} \quad \text{and} \quad \beta = \mathrm{u}_{\mathrm{B}} + \frac{\mathrm{n}_{\mathrm{B}}}{\mathrm{n}_{\mathrm{C}}}\mathrm{u}_{\mathrm{C}}, \tag{5.50}$$

where u_{A}, u_{B} and u_{C} are the concentrations of the chemical species A, B and C, respectively. Once numerical values for α and β are found, the concentrations of the participating chemical species can be calculated as

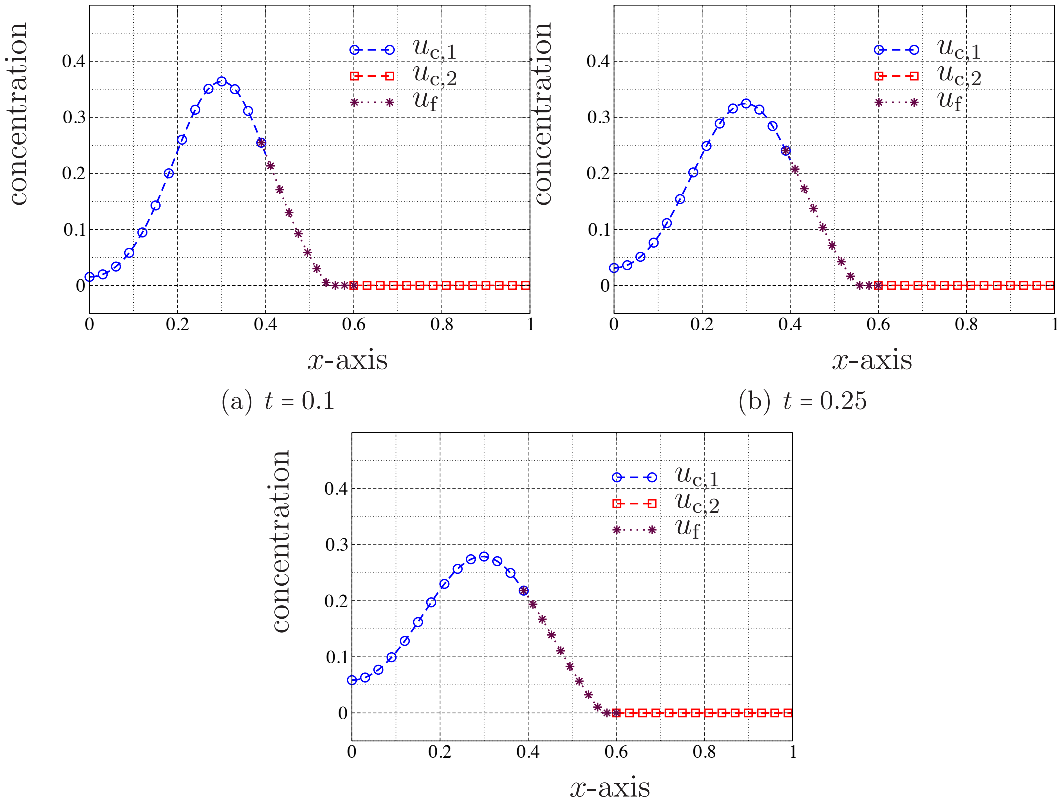
$$\mathrm{u}_{\mathrm{A}} = \max\left\{\alpha - \frac{\mathrm{n}_{\mathrm{A}}}{\mathrm{n}_{\mathrm{B}}}\beta,\ 0\right\}, \tag{5.51a}$$

$$\mathrm{u}_{\mathrm{B}} = \frac{\mathrm{n}_{\mathrm{B}}}{\mathrm{n}_{\mathrm{A}}}\max\left\{-\alpha + \frac{\mathrm{n}_{\mathrm{A}}}{\mathrm{n}_{\mathrm{B}}}\beta,\ 0\right\}, \text{ and} \tag{5.51b}$$

$$\mathrm{u}_{\mathrm{C}} = \frac{\mathrm{n}_{\mathrm{C}}}{\mathrm{n}_{\mathrm{A}}}\left(\alpha - \mathrm{u}_{\mathrm{A}}\right)\cdot \tag{5.51c}$$

Subdomains $\Omega_{\mathrm{c},1}$ and $\Omega_{\mathrm{c},2}$ are discretized using the finite element method, with a mesh size of $h_{\mathrm{c},1} = h_{\mathrm{c},2} = 10^{-2}$ and time-step of $\Delta t_{\mathrm{c},1} = \Delta t_{\mathrm{c},2} = 5 \times 10^{-3}$. Subdomain Ω_{f} is solved using the lattice Boltzmann method with cell size of $h_{\mathrm{f}} = 10^{-3}$ and time-step $\Delta t_{\mathrm{f}} = 2 \times 10^{-5}$. The number of sub-iterations at each time-level is set to 10.

Numerical results at various time-levels are presented in figures 5.19–5.21, which show the concentrations of all the participating chemical species from the coarse-scale subdomains (which are denoted by $\mathrm{u}_{\mathrm{c},1}$ and $\mathrm{u}_{\mathrm{c},2}$) and the fine-scale subdomain (which is denoted by u_{f}). As evident from these figures, the numerical solution is compatible near and in the overlap region, and the proposed hybrid framework has performed well.



(a) $t = 0.1$

(b) $t = 0.25$

(c) $t = 0.5$

Figure 5.19: Fast bimolecular reaction in a one-dimensional domain: Concentration of chemical species A at different time-levels is shown.

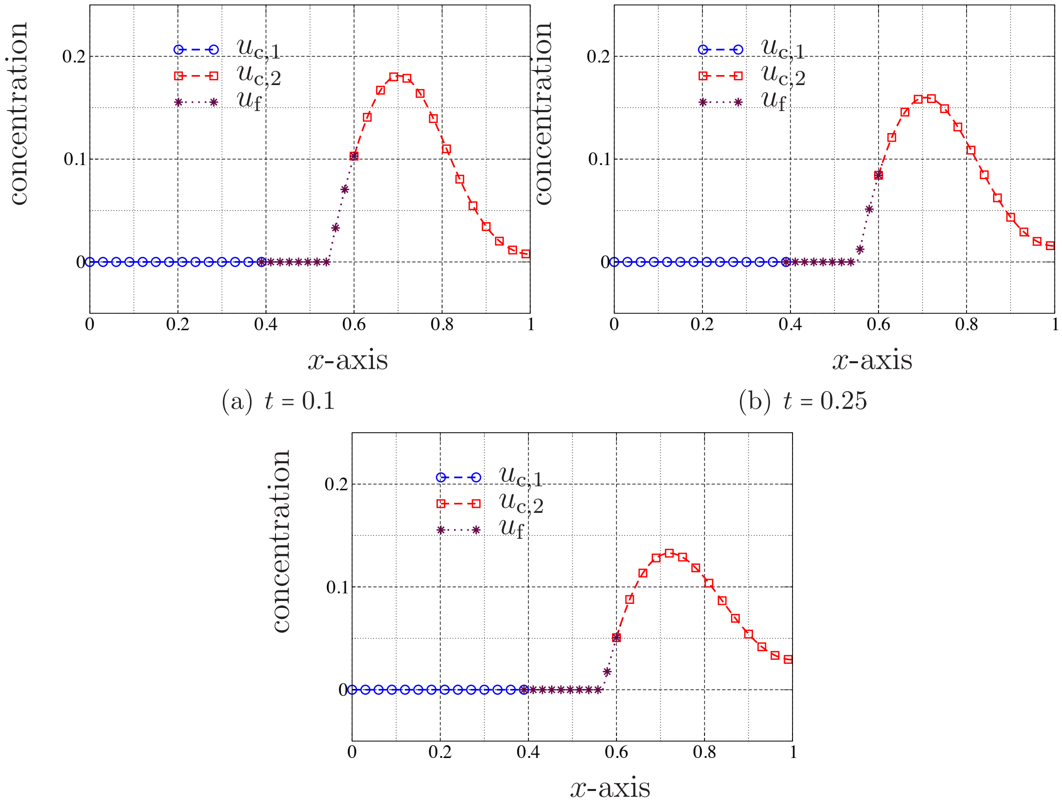
Advection and diffusion in a homogeneous medium

Consider $\Omega = (0,2) \times (0,1/4)$ with $\Gamma^{\mathrm{D}} = \{0\} \times [0,1/4]$ and $\Gamma^{\mathrm{N}} = \partial\Omega - \Gamma^{\mathrm{D}}$ corresponding to the following boundary conditions

$$\mathrm{u}^{\mathrm{p}}(\mathbf{x},\mathrm{t}) = 1 \quad \mathbf{x} \in \Gamma^{\mathrm{D}},\ \mathrm{t} \in \mathcal{I} \text{ and} \tag{5.52}$$

$$\mathrm{q}^{\mathrm{p}}(\mathbf{x},\mathrm{t}) = 0 \quad \mathbf{x} \in \Gamma^{\mathrm{N}},\ \mathrm{t} \in \mathcal{I}, \tag{5.53}$$

where $\mathcal{I} = (0,T]$ is the time interval of interest. The initial concentration in the entire domain is taken to be $\mathrm{u}_0(\mathbf{x}) = 0$. The isotropic diffusion coefficient is $\mathrm{D} = 5 \times 10^{-3}$. Here, we shall use the proposed framework to numerically solve this problem for



(a) $t = 0.1$

(b) $t = 0.25$

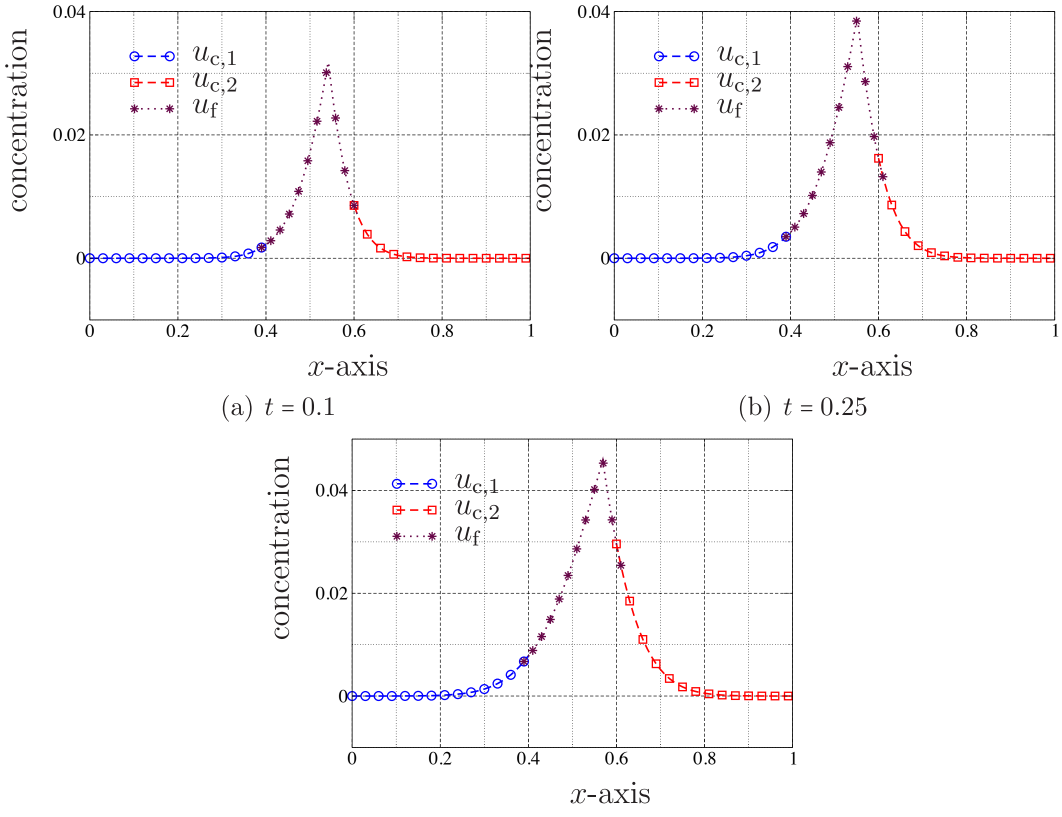
(c) $t = 0.5$

Figure 5.20: Fast bimolecular reaction in a one-dimensional domain: In this figure, concentration of species B is shown.

different Péclet numbers. We will define the coarse-scale domain Ω_c and the fine-scale domain Ω_f as

$$\Omega_c = \left(0, 1 + \frac{L_{\text{overlap}}}{2}\right) \times (0, 1/4), \text{ and } \quad \Omega_f = \left(0, 1 - \frac{L_{\text{overlap}}}{2}\right) \times (0, 1/4), \tag{5.54}$$

where we pick $L_{\text{overlap}} = 4/100$. The SUPG formulation (5.6) with linear three-node triangular elements will be used in Ω_c. Numerical solution in Ω_f will be sought for using lattice Boltzmann method with the $D2Q4$ lattice model. In figure 5.22 non-matching grid sized for finite element and lattice Boltzmann methods in the given domain is illustrated. We shall solve the problem for two different choices of advection velocity:



(a) $t = 0.1$

(b) $t = 0.25$

(c) $t = 0.5$

Figure 5.21: Fast bimolecular reaction in a one-dimensional domain: Concentration of species C is shown. The fine-scale subdomain is located near the region where majority of production occurs.

(i) *Case 1*: Considering the uniform advection velocity of $\mathrm{v_x} = 5 \times 10^{-2}$ and $\mathrm{v_y} = 0$ over domain Ω, we find the Péclet number as $P = 20$. The element-size in the coarse-scale domain is $h_{\mathrm{f}} \approx 7 \times 10^{-2}$, and the grid spacing for LBM is $h_{\mathrm{f}} = 10^{-2}$. The time-steps in the coarse-scale and fine-scale subdomains are $\Delta t_{\mathrm{c}} = 5.1 \times 10^{-1}$ and $\Delta t_{\mathrm{f}} = 10^{-2}$ respectively. Note that the ratio between the coarse and fine time-steps is $\eta = 51$. The number of iterations is MaxIter $= 5$. The result is shown in figure 5.23. The numerical solution from FEM and LBM retained good compatibility while the concentration front passed through the subdomain interfaces. The coupling of the two methods did not result in any disruptions on the propagation of the chemical species in the domain.

(ii) *Case 2*: Here, we will take $\mathrm{v_x} = 5 \times 10^{-1}$ and $\mathrm{v_y} = 0$. In this case the advection velocity is much higher than the previous case, hence, the Péclet number is $P = 200$. In this case, the gradient of concentration near the front is steep. We take $h_\mathrm{c} \approx 2.5{\times}10^{-2}$, $h_\mathrm{f} = 2{\times}10^{-3}$ in coarse and fine-scale subdomains respectively. The time-steps are $\Delta t_\mathrm{c} = 10^{-1}$ and $\Delta t_\mathrm{f} = 4 \times 10^{-4}$. The ratio between the time-steps is $\eta = 250$. Similar to the previous case, the number of sub-iterations is MaxIter = 5. The numerical results are shown in figure 5.24. One of the numerical difficulties that can occur in this case is the spurious oscillations in the concentration. It can be observed that the numerical solution in the coarse-scale domain experiences some of this oscillations (see figure 5.24(a)), however, it should be noted that this weak instability is not due to the hybrid coupling and is an artifact of the finite element formulation. With mesh refinement, these instabilities can be removed. Note that when the front is reaching the interface of the subdomains, some minor incompatibility between the numerical solution of different subdomains in the overlap region is seen (see figure 5.24(b)). This incompatibility can be alleviated by increasing the number of sub-iterations in each time-step. As expected, once the front leaves the coarse-scale domain completely, no node-to-node oscillations remain. In figure 5.25, the numerical solution using smaller time-steps and mesh size is shown. The time-step in the coarse-scale domain is $\Delta t_\mathrm{c} = 2 \times 10^{-2}$ and $\Delta t_\mathrm{f} = 4 \times 10^{-4}$ in the fine-scale domain. The element-size in the coarse-scale subdomain is $h_\mathrm{c} \approx 1.8 \times 10^{-2}$ and in the

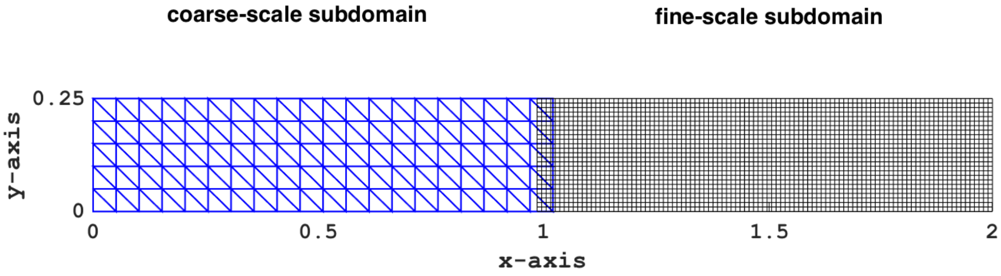


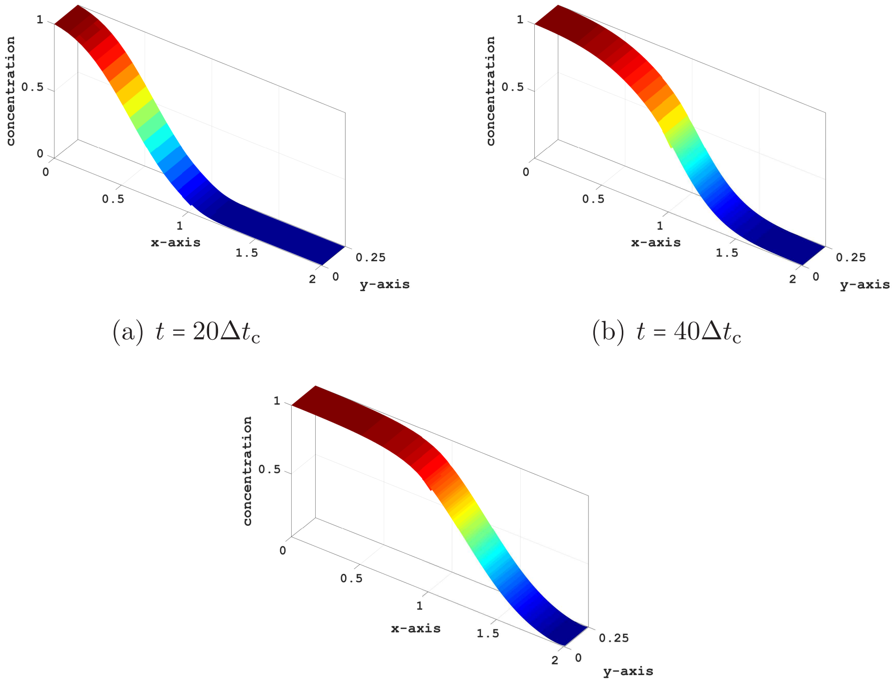
Figure 5.22: Advection and diffusion in a homogeneous medium: This figure illustrates the overlapping domain decomposition as well as the non-matching grids for coarse- and fine-scale domains. The length of the overlap region is $L_{\mathrm{overlap}} = 4 \times 10^{-2}$.

fine-scale subdomain is $h_{\mathrm{f}} = 2 \times 10^{-3}$. The number of sub-iterations in each time-step is increased to 10. Hence, spurious oscillations and incompatibility in the overlap region (while the front is passing through the interface) are largely reduced.

In this numerical experiment we conclude that in order to capture interior/boundary layers more accurately, mere mesh or time-step refinement is not enough. One needs to increase the number of sub-iterations in each time-step.

Hybrid simulation of dissolution of calcium carbonate in porous media

Calcium carbonate $CaCO_3$ is a common chemical compound found in the subsurface. The dissolution of calcium carbonate is an important geochemical equilibrium



(a) $t = 20\Delta t_{\mathrm{c}}$

(b) $t = 40\Delta t_{\mathrm{c}}$

(c) $t = 60\Delta t_{\mathrm{c}}$

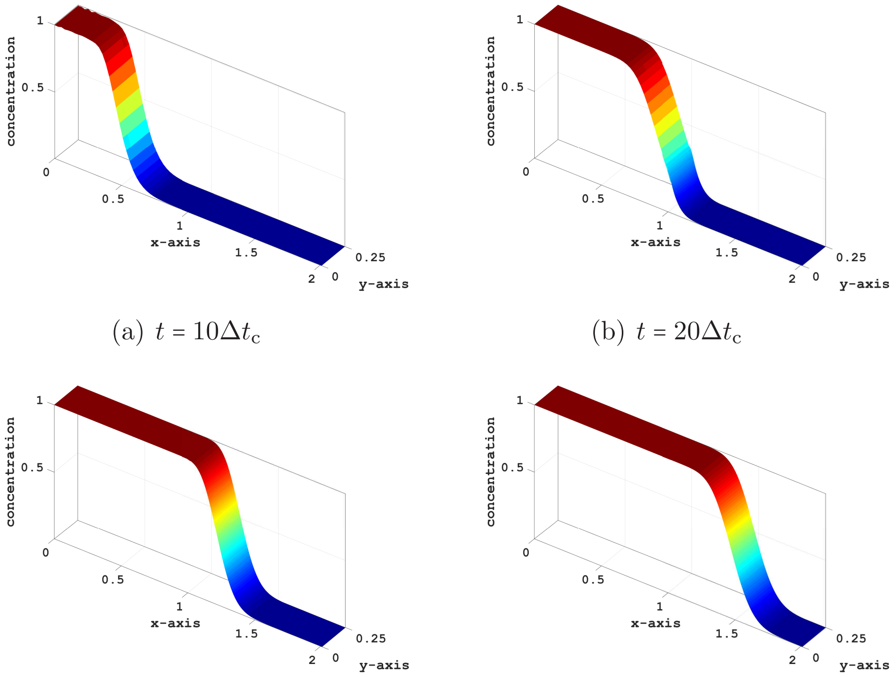
Figure 5.23: Advection and diffusion in a homogeneous medium: In this figure the concentration at different time-levels is shown. In this case Péclet number is $P = 20$. In each time-step, we have employed 5 sub-iterations to ensure the compatibility of the solution in the overlap region.

reaction, which arises in a wide variety of subsurface applications [163]. The chemical reaction takes the form

$$\mathrm{CaCO_3} \rightleftharpoons \mathrm{Ca^{2+}} + \mathrm{CO_3^{2-}} \cdot \tag{5.55}$$

For convenience, we shall use u_1, u_2 and u_3 to denote the concentrations of $\mathrm{CaCO_3}$, $\mathrm{Ca^{2+}}$ and $\mathrm{CO_3^{2-}}$, respectively. This chemical reaction is known to have a product solubility constant K_{sp} of about 3.36×10^{-9} at room temperature [164]. The product solubility for this chemical reaction can be written as

$$K_{\mathrm{sp}} = \frac{\mathrm{u}_2\mathrm{u}_3}{\mathrm{u}_1} \cdot \tag{5.56}$$



(a) $t = 10\Delta t_c$

(b) $t = 20\Delta t_c$

(c) $t = 30\Delta t_c$

(d) $t = 40\Delta t_c$

Figure 5.24: Advection and diffusion in a homogeneous medium: Concentration at different time-levels is shown. In this case Péclet number is $P = 200$. In each time-step we use 5 sub-iterations.

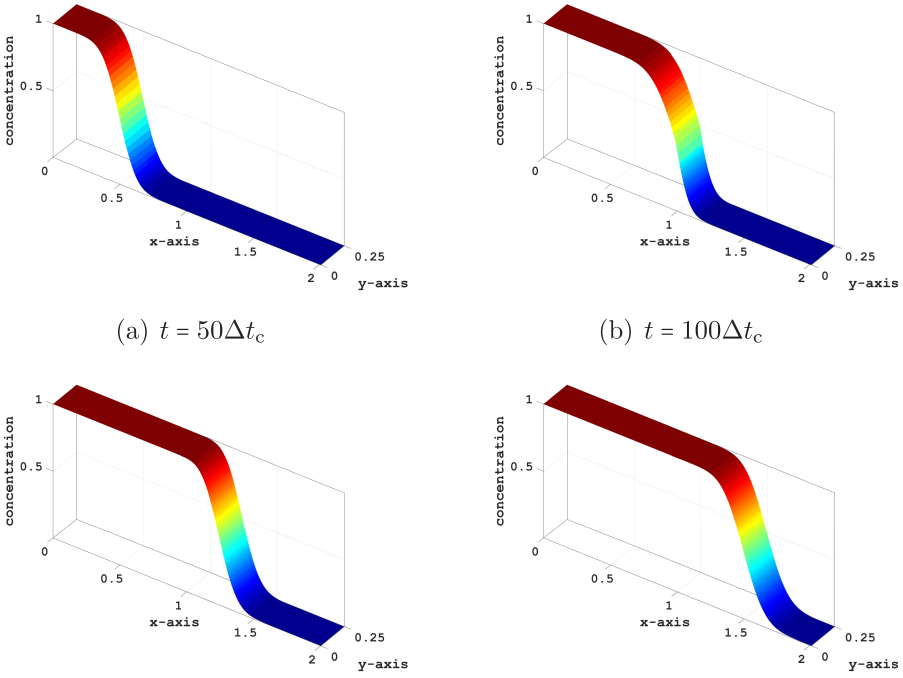
We introduce the two reaction invariants

$$\psi_1 = \mathrm{u}_1 - \mathrm{u}_2 \text{ and} \tag{5.57a}$$

$$\psi_2 = \mathrm{u}_3 - \mathrm{u}_2. \tag{5.57b}$$

It should be emphasized that ψ_1 and ψ_2 are not the concentrations of any real chemical species. These invariants are introduced to simplify the problem, as they decouple the governed equations and hence can be solved for separately; for example, see [165]. Once the values of ψ_1 and ψ_2 are found, the concentration of the species Ca^{2+} can be determined using the relation

$$\mathrm{u}_2 = \frac{1}{2}\left(-\left(\psi_2 + K_{\mathrm{sp}}\right) + \sqrt{\left(\psi_2 + K_{\mathrm{sp}}\right)^2 + 4K_{\mathrm{sp}}\psi_1}\right), \tag{5.58}$$



(a) $t = 50\Delta t_{\mathrm{c}}$

(b) $t = 100\Delta t_{\mathrm{c}}$

(c) $t = 150\Delta t_{\mathrm{c}}$

(d) $t = 200\Delta t_{\mathrm{c}}$

Figure 5.25: Advection and diffusion in a homogeneous medium: In this figure, concentration at different time-levels is shown.

which is obtained by solving equations (5.56)–(5.57) for u_2. The values of u_1 and u_3 can then be determined using equations (5.57a)–(5.57b). Here, we are interested in determining the fate of the chemical species due to the chemical reaction and transport. We employ the LBM to simulate the transport problem at the pore-scale (fine-scale) and the FEM at the continuum-scale.

The computational domain is shown in figure 5.26 where in $L_{\mathrm{x}} = 2$ and $L_{\mathrm{y}} = 1$. The radius of the solid obstacles in Ω_{f} (the fine-scale problem) is taken as $r = 10^{-1}$. The length of the overlap region is set to $L_{\mathrm{overlap}} = 10^{-1}$. Obviously, because of the geometry of Ω_{f}, a more detailed description of the flow is required. We used LBM with a $D2Q9$ lattice model to solve the Navier-Stokes equations in the fine-scale subdomain Ω_{f} [75, 150].

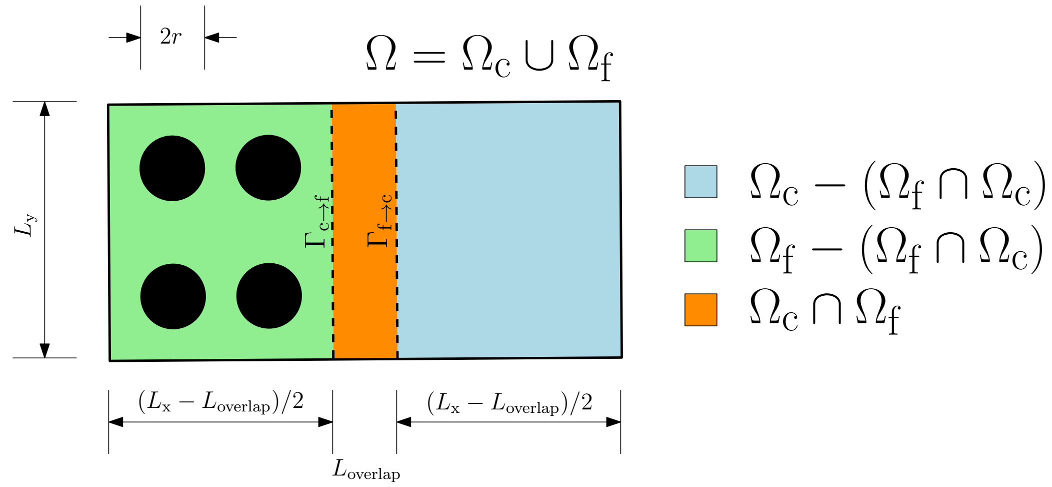
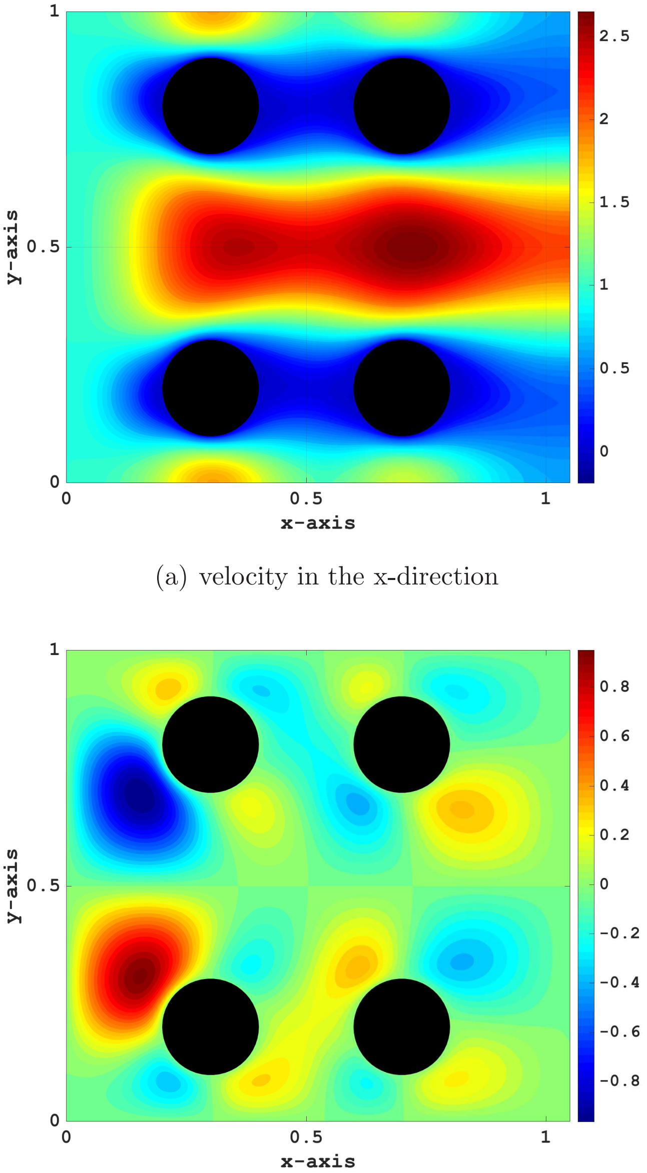


Figure 5.26: Dissolution of calcite problem: Computational domain and its decomposition into fine and coarse-scale subdomains are shown. The black circles represent the solid phase in the porous medium.

The prescribed components on the inlet velocity on the boundary $x = 0$ are $v_x = 1$ and $v_y = 0$. The pressure on $\Gamma_{f\to c}$ is set to be zero and periodic boundary conditions are enforced on the boundaries located at $y = 0$ and $y = 1$ for $0 < x < (L_x + L_{overlap})/2$. The resulting velocity field is shown in figure 5.27 and will be used as the advection velocity for the fine-scale problem. In the overlap region, the average velocity in the x-direction is close to 1 and the average velocity in the y-direction is close to 0. Hence, the advection velocity in the coarse-scale domain is taken to be $v_x = 1$ and $v_y = 0$. The values of concentrations on the boundary of the domain are shown in figure 5.28 and the diffusion coefficient is taken to be $D = 10^{-1}$. For numerical simulation of the advection-diffusion problem, we will use $h_c = 5.0 \times 10^{-2}$ and $h_f = 4.0 \times 10^{-3}$. The time-steps are $\Delta t_c = 10^{-1}$ and $\Delta t_f = 4.0 \times 10^{-5}$ (the ratio between the time-steps is $\eta = 2500$). Furthermore, we will use the $D2Q9$ lattice model in the fine-scale domain (solved using LBM). The non-matching grid near the overlap region is shown in figure 5.29. Obviously, one of the advantages of the proposed coupling algorithm is that



(a) velocity in the x-direction

(b) velocity in the y-direction

Figure 5.27: Two-dimensional problem: The velocity field shown in this figure is obtained using a lattice Boltzmann simulation of incompressible Newtonian fluid. The black circles represent the solid obstacles in the porous medium.

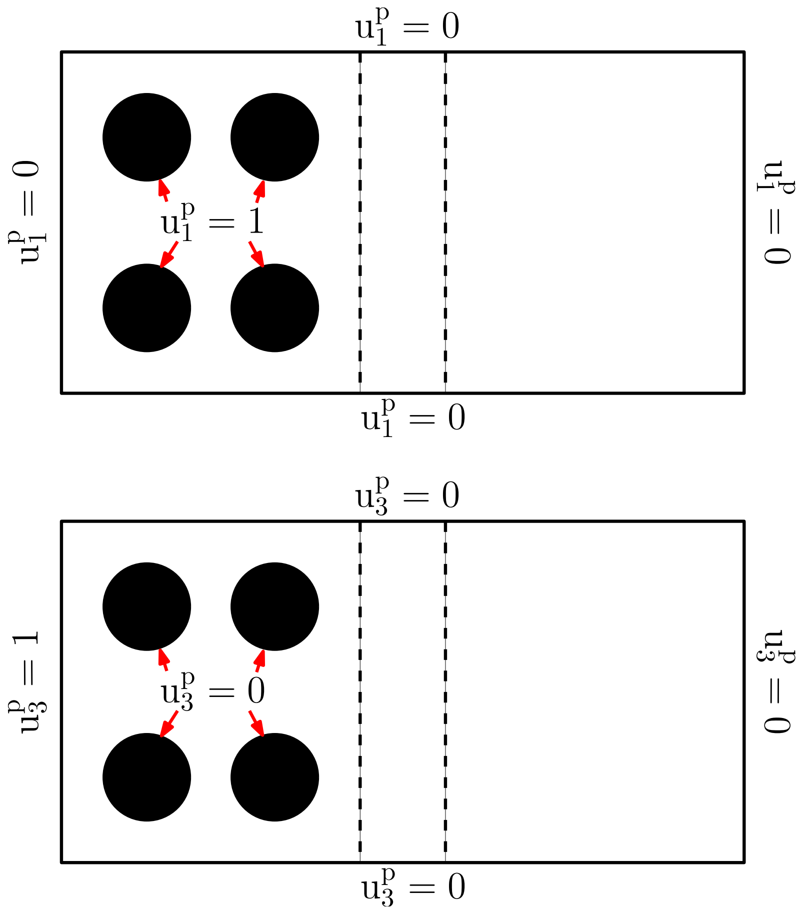


Figure 5.28: Dissolution of calcite: The boundary conditions for the simulation of dissolution of calcite in the porous medium are shown.

fine-scale features (such as advection velocity within the pores) can be accounted for without a noticeable overhead in the computational cost. In this problem, fine-scale features are sought after only in Ω_f, and a coarse estimate in Ω_c is deemed enough.

The concentrations of the participating chemical species are shown in figures 5.30–5.32. The numerical simulation reveals that the concentrations of $CaCO_3$ and CO_3^{2-} inside the domain increase with time. However, the evolution of Ca^{2+} cations is completely different from that of the other two chemical species. At earlier time-levels, when the concentrations of $CaCO_3$ and CO_3^{2-} are low within the domain, Ca^{2+} has a more noticeable presence throughout the domain. At later time-levels,

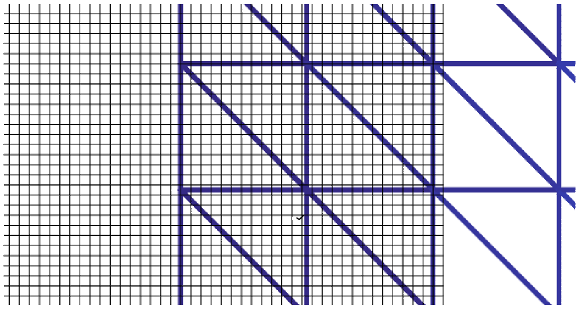


Figure 5.29: Dissolution of calcite: This figure shows the finite element mesh (which is indicated using triangular elements) and the lattice for LBM analysis (which is indicated by square cells) near the overlapping interface.

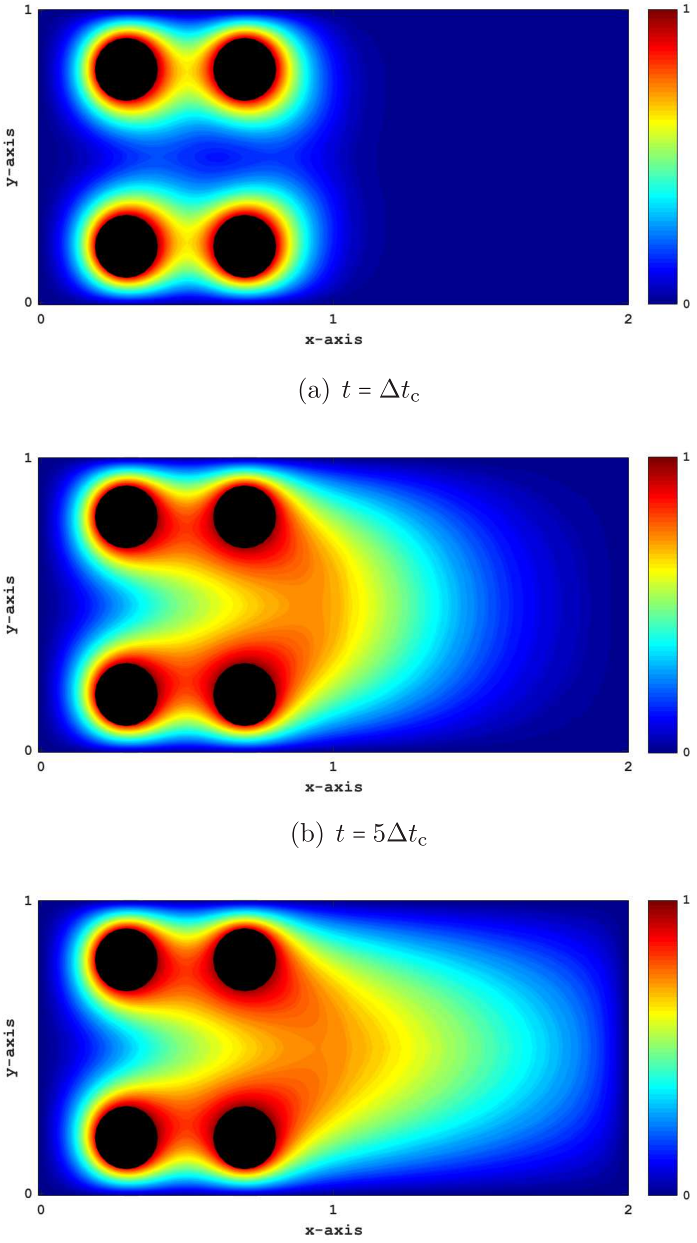
as a consequence of increasing concentration of CO_3^{2-} anions, Ca^{2+} disappears from much of the domain and gathers in the regions where the concentration of CO_3^{2-} is low. Figure 5.33 further corroborates this finding, in which the normalized total concentrations of chemical species are plotted against time. The total concentration in the entire domain, $\mathcal{C}_{\mathrm{total}}$, is defined as

$$\mathcal{C}_{\mathrm{total}}(t) = \int_{\Omega} u_i(\boldsymbol{x}, t) \; \mathrm{d}\Omega, \quad i = 1, 2, 3 \cdot \tag{5.59}$$

The normalization for each chemical species is done with respect to the corresponding maximum in the time interval of interest. That is,

$$\max_{t} \; \mathcal{C}_{\mathrm{total}}(t) \cdot \tag{5.60}$$

In this example, we have demonstrated how to use the proposed multi-time-step hybrid coupling framework for the analysis of geochemical processes by simultaneously incorporating both pore and continuum models. A detailed pore geometry and complex transport processes can be accounted for in the fine-scale domain, whereas a rough approximation can be sought in the coarse-scale domain.

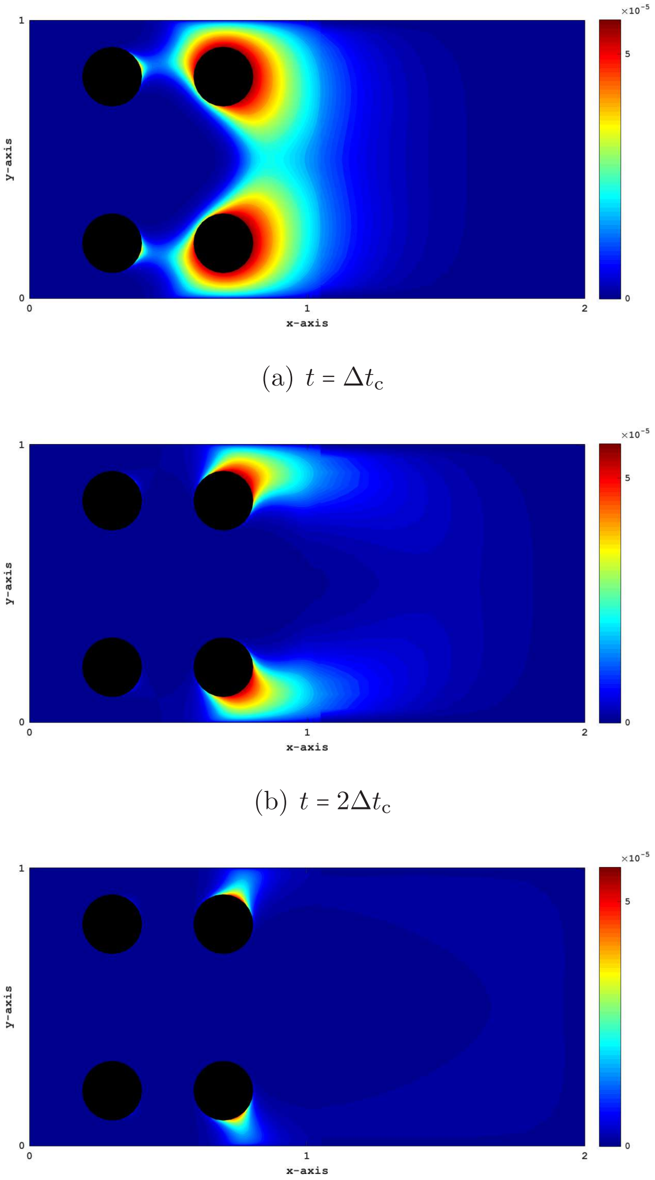


(a) $t = \Delta t_{\mathrm{c}}$

(b) $t = 5\Delta t_{\mathrm{c}}$

(c) $t = 10\Delta t_{\mathrm{c}}$

Figure 5.30: Dissolution of calcite: In this figure, concentration of calcite at different time-levels is shown. Initially, calcite is concentrated near the solid obstacles and is transported throughout the domain at later times.

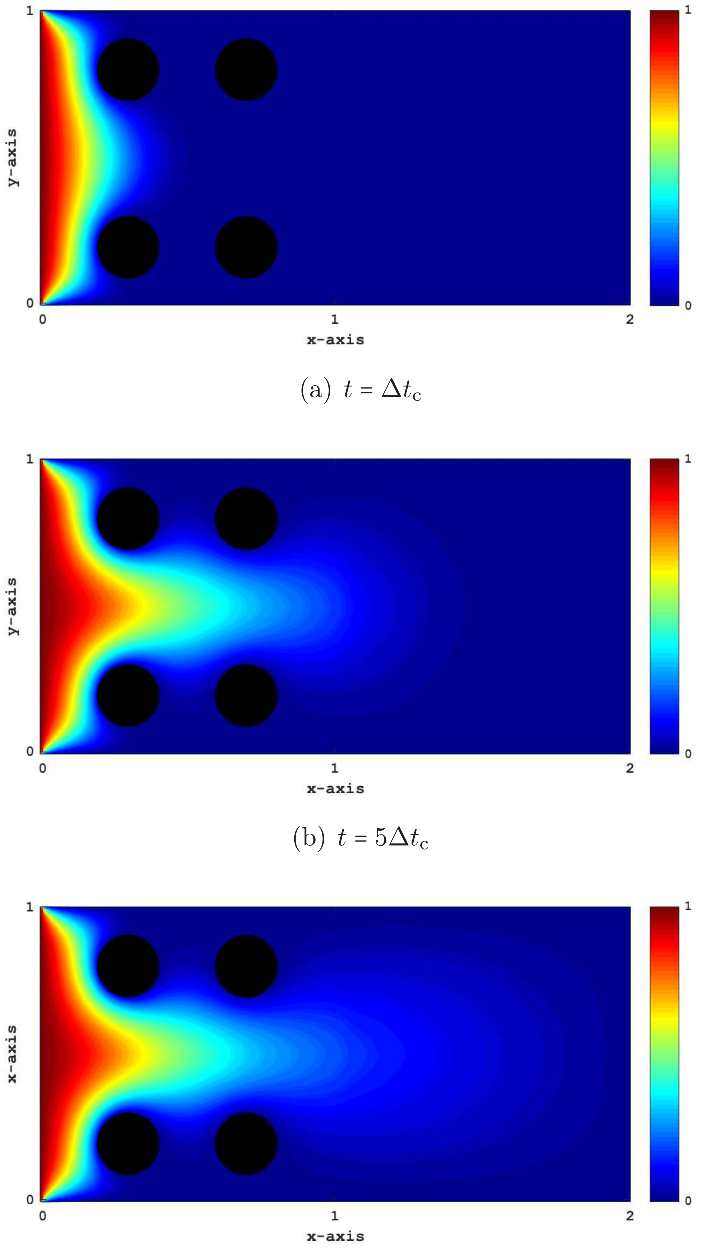


(a) $t = \Delta t_c$

(b) $t = 2\Delta t_c$

(c) $t = 3\Delta t_c$

Figure 5.31: Dissolution of calcite: In this figure, concentration of Ca^{2+} is shown. It can be observed that in the first time-steps, concentration of this chemical species is more spread out in the spatial domain.



(a) $t = \Delta t_c$

(b) $t = 5\Delta t_c$

(c) $t = 10\Delta t_c$

Figure 5.32: Dissolution of calcite: Concentration of the CO_3^{2-} is shown at different time-levels. This chemical species is often in solute form.

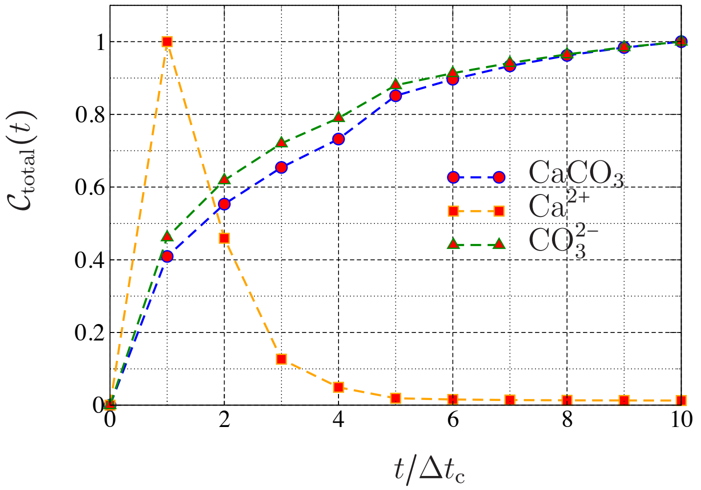


Figure 5.33: Dissolution of calcite: This figure shows the variation of the total concentration in the entire domain of each participating chemical species with respect to time.

5.8 Concluding Remarks

Simulation of transport of chemical species in porous media poses several challenges. These include disparate mathematical scales in space and time, not all the essential physical and chemical processes can be upscaled from the pore-scale to the meso-scale, high computational cost to solve realistic problem; just to name a few. In this chapter, we have presented a computational framework that can make multi-scale simulation of transport in porous media feasible even for realistic problems. The framework allows to take into account the features and processes at the pore-scale and still be able to solve problems at the field-scale with manageable computational cost. The findings and advances made in this chapter can be listed as follows:

(i) Simulation of advection and diffusion using LBM: The lattice Boltzmann method for simulation of transport is outlined. A drawback of LBM in such simulations can be the possibility of discrete distributions attaining unphysical (negative) values. To rectify this issue, we presented a bound on discretization parameters under LBM that guarantees non-negativity of discrete populations. Furthermore, new methods for enforcing macroscopic boundary values, in the form of Neumann or Dirichlet conditions, on the numerical solution from the LBM are proposed. These methods are based on entropy principles and warrant non-negative values for discrete populations.

(ii) Information transfer across non-matching grids: Methods for transferring information from one computational grid to another non-matching grid were documented. Accuracy of these methods with respect to grid size in different domains is also explored.

(iii) Governing equations for hybrid simulation: Time and space continuous partial differential equations for coupled analysis are presented. These equations provide a precise mathematical framework for further developments in this area of research.

(iv) Hybrid coupling computational framework: A numerical framework, based on domain decomposition, was presented that can employ different numerical methods (e.g., finite element method and lattice Boltzmann method) in different subdomains. This framework can account for pore-scale processes as well as continuum scale models. Also, disparate spatial and temporal discretization can be incorporated. Hence, the primary factor in choosing grid size and time-steps in each subdomain is the accuracy in that subdomain. The hybrid coupling framework poses no restriction on the discretization parameters in different subdomains. Furthermore, various chemical reaction dynamics among the present chemical species can be included using LBM and other approximations of the same phenomena in the finite element solver. In all of the numerical experiments, this framework was numerically stable and accurate. Interior layers can be captured accurately and typical weak instabilities in the solution can be suppressed using appropriate numerical techniques (such as stabilized finite element formulations) in those subdomains. We also demonstrated application of this framework in assessing the fate of chemical species in a sample geochemical reaction problem. As a courtesy of its domain decomposition basis, this framework provides the user with great flexibility in distributing the computational workload onto different processors and possibly in a heterogeneous GPU-CPU computing setup. For instance, the subdomains solved using the lattice Boltzmann method can be transferred to a GPU, while other subdomains where the finite element method is used can be solved for using a different processing environment. This computational framework can handle multiple subdomains using the multiplicative Schwartz methods.

We shall conclude the chapter by outlining some possible future research directions.

(R1) A good research endeavor can be towards a comprehensive mathematical analysis (i.e., stability, accuracy and convergence properties) of the proposed computational framework.

(R2) One can implement the proposed computational framework in a combined

GPU-CPU computing environment, and study the numerical performance of such an implementation.

(R3) Substantial progress in development of hybrid methods can result from extension of the proposed computational framework to fully coupled thermal-flow-transport processes, including precipitation at the solid-fluid interface (in pore-scale) and application of such methods to simulation of viscous fingering and other physical instabilities.

Chapter 6

CONCLUDING REMARKS AND FUTURE WORK

The findings and achievements of this research endeavor can be summarized as follows:

(1) Development of monolithic and multi-time-step coupling method for elastodynamics: Based on domain decomposition methods for partial differential equations, we proposed a coupling method that allows different time-integrators and time-steps in different regions of the computational domain. The governing time-dependent equations in the case of elastodynamics form a system of differential/algebraic equations with differential index 3. We considered the Newmark family of time-integration methods, and derived numerical stability conditions under the proposed coupling algorithm. Under this method, the transfer of information among subdomains is done in one iteration (at each time-level), hence, the proposed method can be regarded as a *monolithic* coupling algorithm. We studied the stability of this method using the method of influence of perturbations and derived bounds on the drift in the primary variables at the interface of subdomains. Furthermore, physical energy conservation under this method was carefully studied and as a result new classification of energy conservation properties for coupling algorithms was developed. Based on these findings, accuracy of the proposed method with respect to time-step ratios was studied and

a criteria for assessing the effect of multi-time-stepping on numerical accuracy was derived. Several numerical examples demonstrated the performance and the theoretical findings regarding the proposed method. The results of this research effort are available in reference [57].

(2) Development of monolithic and multi-time-step coupling method for advection-diffusion-reaction equations: Non-overlapping domain decomposition method was used to construct a computational framework for strong coupling of different time-steps and time-integrators in different subdomains. The proposed methods use two different methods of enforcing interface compatibility constraints. One is based on compatibility of concentrations at the interface and the other based on Baumgarte's stabilization technique. Numerical stability of both methods was studied using the energy method. We showed that under the continuity of concentrations, mixed implicit/explicit integration results in numerical stability. However, the second method based on Baumgarte stabilization allows implicit/explicit time-integration. We further studied the stability of these coupling methods using the method of influence of perturbations. We also derived bounds on the drift at the interface of subdomains. Several numerical examples were used to demonstrate performance of the proposed methods and the theoretical findings. The results are published in [127].

(3) Studying the preservation of maximum principle and the non-negative constraint under the lattice Boltzmann method: Lattice Boltzmann methods have gained much traction in the past few years as robust computational alternatives to con-

ventional computational fluid dynamics tools. Some of the recent efforts have been toward developing single and multiple-relaxation-time methods to simulate advection-diffusion processes. Considering the importance of maximum principle and the non-negative constraint in generating reliable and physically meaningful results, performance of these methods in preserving these properties needs to be studied. We implemented some popular lattice Boltzmann techniques for a variety of test problems. We observed that preservation of the mentioned properties under current lattice Boltzmann methods is not guaranteed. We discussed possible sources of these violations and proposed a criteria for time and space discretization that ensures non-negativity of the numerical solution in the case of the isotropic diffusion. The results of this research effort is available in reference [144].

(4) Development of a hybrid and multi-time-step coupling algorithm for solute transport in porous media: Obtaining reliable estimates for the transport processes in porous media is not possible without relying on different numerical models for different length or time-scales. Presence of pore-scale reactions, solid-fluid interface chemistry and the complex geometry of pores vastly effects the outcome of the model in field-scale simulations. In this research endeavor, we took the challenge of developing a computational framework based on overlapping domain partitioning for multi-scale simulation of transport process. For pore-scale simulation we used the lattice Boltzmann method and for continuum-scale simulation we used the finite element method. We developed new *entropic* meth-

ods for transferring information form the continuum-scale problem to the lattice Boltzmann method. Hence, the proposed framework allows different grid-sizes and time-steps in different subdomains. We applied the proposed method to a variety of benchmark and also chemical reaction problems to showcase its performance. The results are available in reference [166].

Based on these findings and developments, the following research directions for future work are suggested:

(1) Implementation of the proposed monolithic coupling algorithms in a parallel computing setup and studying its performance. For instance, how the number of subdomains and their time-step ratios can help us to reduce the overall computational cost.

(2) Further study of the numerical accuracy of the proposed monolithic coupling methods using more precise mathematical tools.

(3) Extending the proposed monolithic coupling methods to incorporate non-matching finite element grids in different subdomains.

(4) Developing adaptive domain partitioning techniques and applying the proposed monolithic coupling methods to crack propagation problems in solid mechanics.

(5) Developing multiple-relaxation-time lattice Boltzmann methods that preserve the maximum principle and the non-negative constraint.

(6) Devising numerical methodologies to improve the convergence rate of the proposed hybrid coupling method for transport problems.

(7) Extending the proposed hybrid coupling algorithm for coupled flow and transport problems in porous media.

Bibliography

[1] C. Geuzaine and J.-F. Remacle. Gmsh: A three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79:1309–1331, 2009.

[2] G. Karypis and V. Kumar. A fast and highly quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20:359–392, 1999.

[3] J. T. Oden, T. Belytschko, J. Fish, T. J. R. Hughes, C. Johnson, D. Keyes, A. Laub, L. Petzold, D. Srolovitz, and S. Yip. Revolutionizing engineering science through simulation: A report of the national science foundation blue ribbon panel on simulation-based engineering science. *National Science Foundation, Arlington, Virginia*, 2006.

[4] T. J. R. Hughes, G. R. Feijóo, L. Mazzei, and J. Quincy. The variational multiscale method: A paradigm for computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, 166(1):3–24, 1998.

[5] J. M. Melenk and I. Babuška. The partition of unity finite element method: Basic theory and applications. *Computer Methods in Applied Mechanics and Engineering*, 139(1):289–314, 1996.

[6] T. Y. Hou and X. H. Wu. A multiscale finite element method for elliptic problems in composite materials and porous media. *Journal of computational physics*, 134(1):169–189, 1997.

[7] A. Toselli and O. Widlund. *Domain Decomposition Methods: Algorithms and Theory*, volume 3. Springer, Heidelberg, 2005.

[8] A. Gravouil and A. Combescure. Multi-time-step explicit-implicit method for non-linear structural dynamics. *International Journal for Numerical Methods in Engineering*, 50:199–225, 2001.

[9] K. B. Nakshatrala, K. D. Hjelmstad, and D. A. Tortorelli. A FETI-based domain decomposition technique for time dependent first-order systems based on a DAE approach. *International Journal for Numerical Methods in Engineering*, 75:1385–1415, 2008.

[10] E. Hairer, S. P. Norsett, and G. Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer-Verlag, New York, USA, second edition, 2011.

[11] T. Belytschko, H. J. Yen, and R. Mullen. Mixed methods for time integration. *Computer Methods in Applied Mechanics and Engineering*, 17–18:259–275, 1979.

[12] W. K. Liu and T. Belytschko. Mixed-time implicit-explicit finite elements for transient analysis. *Computers and Structures*, 15:445–450, 1982.

[13] T. Belytschko and R. Mullen. Explicit integration of structural problems. *Finite Elements in Nonlinear Mechanics*, 2:697–720, 1977.

[14] T. Belytschko and R. Mullen. Stability of explicit-implicit mesh partitions in time integration. *International Journal of Numerical Methods in Engineering*, 12:1575–1586, 1978.

[15] T. J. R. Hughes. Generalization of selective integration procedures to anisotropic and nonlinear media. *International Journal for Numerical Methods in Engineering*, 15:1413–1418, 1980.

[16] T. J. R. Hughes and W. K. Liu. Implicit-explicit finite elements in transient analysis: stability theory. *Journal of Applied Mechanics*, 45:371–374, 1978.

[17] T. J. R. Hughes and W. K. Liu. Implicit-explicit finite elements in non-linear transient analysis. *Computer Methods in Applied Mechanics and Engineering*, 17:159–182, 1979.

[18] P. Smolinski, T. Belytschko, and M. Neal. Multi-time-step integration using nodal partitioning. *International Journal of Numerical Methods in Engineering*, 26:349–359, 1988.

[19] W. J. T. Daniel. The subcycled Newmark algorithm. *Computational Mechanics*, 20:272–281, 1997.

[20] P. B. Nakshatrala, K. B. Nakshatrala, and D. A. Tortorelli. A time-staggered partitioned coupling algorithm for transient heat conduction. *International Journal for Numerical Methods in Engineering*, 78:1387–1406, 2009.

[21] A. Akkasale. Stability of Coupling Algorithms. Master's thesis, Texas A&M University, College Station, Texas, USA, 2011. http://repository.tamu.edu/handle/1969.1/ETD-TAMU-2011-05-9500.

[22] L. Petzold. Differential/algebraic equations are not ODEs. *SIAM Journal on Scientific and Statistical Computing*, 3:367–384, 1982.

[23] K. B. Nakshatrala, A. Prakash, and K. D. Hjelmstad. On dual Schur domain decomposition method for linear first-order transient problems. *Journal of Computational Physics*, 228:7957–7985, 2009.

[24] U. M. Ascher and L. R. Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations.* SIAM, Philadelphia, USA, 1998.

[25] A. Bonelli, O. S. Bursi, L. He, G. Magonette, and P. Pegon. Convergence Analysis of a Parallel Interfield Method for Heterogeneous Simulations with Dynamics Substructuring. *International Journal For Numerical Methods in Engineering*, 758:800–825, 2008.

[26] A. Bonelli O. S. Bursi, L. He and P. Pegon. Novel Generalized-α Methods for Interfield Parallel Integration of Heterogeneous Structural Dynamics Systems. *Journal of Computational and Applied Mathematics*, 234:2250–2258, 2010.

[27] A. Bonelli C. Jia, O. S. Bursi and Z. Wang. Novel Partitioned Time Integration Methods for DAE Systems Based on L-Stable Linearly Implicit Algorithms.

International Journal for Numerical Methods in Engineering, 87:1148–1182, 2011.

[28] N. Mahjoubi and S. Krenk. Multi-time-step domain coupling method with energy control. *International Journal for Numerical Methods in Engineering*, 83:1700–1718, 2010.

[29] A. Combescure N. Mahjoubi, A. Gravouil and N. Greffet. A monolithic energy conserving method to couple heterogeneous time integrators with incompatible time steps in structural dynamics. *Computer Methods in Applied Mechanics and Engineering*, 200:1069–1086, 2011.

[30] A. Prakash and K. D. Hjelmstad. A FETI-based multi-time-step coupling method for Newmark schemes in structural dynamics. *International Journal for Numerical Methods in Engineering*, 61:2183–2204, 2004.

[31] T. J. R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Prentice-Hall, Englewood Cliffs, New Jersey, USA, 1987.

[32] N. M. Newmark. A method of computation for structural dynamics. *Journal of the Engineering Mechanics Division, ASCE*, 85:67–94, 1959.

[33] W. L. Wood. *Practical Time-Stepping Schemes*. Oxford University Press, New York, USA, 1990.

[34] B. Leimkuhler and S. Reich. *Simulating Hamiltonian Dynamics*. Cambridge University Press, Cambridge, UK, 2005.

[35] M. Geradin and D. Rixen. *Mechanical Vibrations: Theory and Applications to Structural Dynamics*. John Wiley & Sons Ltd., Chichester, U.K., second edition, 1997.

[36] A. Toselli and O. Widlund. *Domain Decomposition Methods*. Springer-Verlag, New York, USA, 2004.

[37] C. W. Gear. Differential-algebraic equations, indices and integral algebraic equations. *SIAM Journal on Numerical Analysis*, 27:1527–1534, 1990.

[38] K. Brenan, S. Campbell, and L. Petzold. *Numerical Solutions of Initial-Value Problems in Differential-Algebraic Equations*. North-Holland, New York, USA, 1989.

[39] M. Geradin and A. Cardona. *Flexible Multibody Dynamics: A Finite Element Approach*. John Wiley & Sons Ltd., Chichester, U.K., 2001.

[40] J. Baumgarte. Stabilization of constraints and integrals of motion in dynamical systems. *Computer Methods in Applied Mechanics and Engineering*, 1:1–16, 1972.

[41] A. Younes, P. Ackerer, and F. Lehmann. Stability and accuracy analysis of Baumgarte's constraint violation stabilization method. *ASME Journal of Mechanical Design*, 117:446–453, 1995.

[42] O. A. Bauchau. Review of contemporary approaches for constraint enforcement in multibody systems. *Journal of Computational and Nonlinear Dynamics*, 3:1–8, 2008.

[43] P. Flores, R. Pereira, M. Machado, and E. Seabra. Investigation on the Baumgarte stabilization method for dynamic analysis of constrained multibody systems. *Proceedings of EUCOMES 08*, pages 305–312, 2009.

[44] A. Prakash. *Multi-time-step Domain Decomposition and Coupling Methods for Non-linear Structural Dynamics*. PhD thesis, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA, 2007.

[45] L. C. Evans. *Partial Differential Equations*. American Mathematical Society, Providence, Rhode Island, 1998.

[46] R. D. Richtmyer and K. W. Morton. *Difference Methods for Initial-Value Problems*. Krieger Publishing Company, Malabar, Florida, USA, 1994.

[47] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Springer-Verlag, New York, 1996.

[48] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer-Verlag, New York, 2009.

[49] C. Farhat and F. Roux. A method of finite element tearing and interconnecting and its parallel solution algorithm. *International Journal for Numerical Methods in Engineering*, 32:1205–1227, 1991.

[50] P. M. Gresho and R. L. Sani. *Incompressible Flow and the Finite Element Method: Advection-Diffusion*, volume 1. John Wiley & Sons, Inc., Chichester, U.K., 2000.

[51] R. McOwen. *Partial Differential Equations: Methods and Applications*. Prentice Hall, New Jersey, 1996.

[52] D. Walgraef. *Spatio-Temporal Pattern Formation*. Springer-Verlag, New York, 1997.

[53] R. B. Bird, W. E. Stewart, and E. N. Lightfoot. *Transport Phenomena*. John Wiley & Sons, Inc., New York, second edition, 2006.

[54] R. Codina. On stabilized finite element methods for linear systems of convection-diffusion-reaction equations. *Computer Methods in Applied Mechanics and Engineering*, 188:61–82, 2000.

[55] M. Augustin, A. Caiazzo, A. Fiebach, J. Fuhrmann, V. John, A. Linke, and R. Umla. An assessment of discretizations for convection-dominated convection-diffusion equations. *Computer Methods in Applied Mechanics and Engineering*, 200:3395–3409, 2011.

[56] E. Burman and A. Ern. Nonlinear diffusion and discrete maximum principle for stabilized Galerkin approximations of the convection-diffusion-reaction equation. *Computer Methods in Applied Mechanics and Engineering*, 191:3833–3855, 2002.

[57] S. Karimi and K. B. Nakshatrala. On multi-time-step monolithic coupling algorithms for elastodynamics. *Journal of Computational Physics*, 273:671–705, 2014.

[58] S. Piperno, C. Farhat, and B. Larrouturou. Partitioned procedures for the transient solution of coupled aeroelastic problems Part I: Model problem, theory and two-dimensional application. *Computer Methods in Applied Mechanics and Engineering*, 124:79–112, 1995.

[59] S. Piperno. Explicit/implicit fluid/structure staggered procedures with a structural predictor and fluid subcycling for 2D inviscid aeroelastic simulations. *International Journal for Numerical Methods in Fluids*, 25:1207–1226, 1997.

[60] C. W. Gear and L. R. Petzold. ODE methods for the solution of differential/algebraic systems. *SIAM Journal on Numerical Analysis*, 27:716–728, 1984.

[61] L. Petzold. Numerical solution of differential-algebraic equations in mechanical systems simulation. *Physica D*, 60:269–279, 1992.

[62] J. Donea and A. Huerta. *Finite Element Methods for Flow Problems*. John Wiley & Sons, Inc., Chichester, 2003.

[63] O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method : Vol.1*. McGraw-Hill, New York, 1989.

[64] F. Brezzi and M. Fortin. *Mixed and Hybrid Finite Element Methods, volume 15 of Springer series in computational mathematics*. Springer-Verlag, New York, 1991.

[65] D. Z. Turner, K. B. Nakshatrala, and K. D. Hjelmstad. A stabilized formulation for the advection-diffusion equation using the Generalized Finite Element

Method. *International Journal for Numerical Methods in Fluids*, 66:64–81, 2011.

[66] L. P. Franca, G. Hauke, and A. Masud. Revisiting stabilized finite element methods for the advective-diffusive equation. *Computer Methods in Applied Mechanics and Engineering*, 195:1560–1572, 2006.

[67] A. N. Brooks and T. J. R. Hughes. Streamline-upwind/Petrov-Galerkin methods for convection dominated flows with emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32:199–259, 1982.

[68] T. J. R. Hughes, L. Franca, and G. Hulbert. A new finite element formulation for computational fluid dynamics: VIII. The Galerkin/least-squares method for advective-diffusive equations. *Computer Methods in Applied Mechanics and Engineering*, 73:173–189, 1989.

[69] V. John and P. Knobloch. On spurious oscillations at layers diminishing (SOLD) methods for convection-diffusion equations: Part I - A review. *Computer Methods in Applied Mechanics and Engineering*, 196:2197–2215, 2007.

[70] P. A. Farrell, P. W. Hemker, and G. I. Shishkin. Discrete approximations for singularly perturbed boundary value problems with parabolic layers. *Journal of Computational Mathematics*, 14:71–97, 1995.

[71] P. W. Hemker. A singularly perturbed model problem for numerical computation. *Journal of Computational and Applied Mathematics*, 76:277–285, 1996.

[72] K. B. Nakshatrala, M. K. Mudunuru, and A. J. Valocchi. A numerical framework for diffusion-controlled bimolecular-reactive systems to enforce maximum principles and non-negative constraint. *Journal of Computational Physics*, 253:278–307, 2013.

[73] G. F. Pinder and M. A. Celia. *Subsurface Hydrology.* John Wiley & Sons, Inc., New Jersey, 2006.

[74] S. Chen and G. D. Doolen. Lattice Boltzmann method for fluid flows. *Annual Review of Fluid Mechanics*, 30(1):329–364, 1998.

[75] S. Succi. *The Lattice Boltzmann Equation.* Oxford University Press, Oxford, 2001.

[76] G. Falcucci, S. Ubertini, C. Biscarini, S. Di Francesco, D. Chiappini, S. Palpacelli, A. De Maio, and S. Succi. Lattice Boltzmann methods for multiphase flow simulations across scales. *Communications in Computational Physics*, 1:1–35, 2011.

[77] A. F. Di Rienzo, P. Asinari, E. Chiavazzo, N. I. Prasianakis, and J. Mantzaras. Lattice Boltzmann model for reacting flow simulations. *Europhysics Letters*, 98:34001, 2012.

[78] S. G. Ayodele, F. Varnik, and D. Raabe. Lattice Boltzmann study of pattern formation in reaction-diffusion systems. *Physical Review E*, 83:016702, 2011.

[79] M. Stiebler, J. Tolke, and M. Krafczyk. Advection-diffusion lattice Boltzmann scheme for hierarchical grids. *Computers and Mathematics with Applications*, 55:1576–1584, 2008.

[80] B. Shi and Z. Guo. Lattice Boltzmann model for nonlinear convection-diffusion equations. *Physical Review E*, 79:16701, 2009.

[81] Z. Chai and T. S. Zhao. Lattice Boltzmann model for convection-diffusion equation. *Physical Review E*, 87:63309, 2013.

[82] H. Yoshida and M. Nagaoka. Multiple-relaxation-time lattice Boltzmann model for the convection and anisotropic diffusion equation. *Journal of Computational Physics*, 229:7774–7795, 2010.

[83] R. Huang and H. Wu. A modified multiple-relaxation-time lattice Boltzmann model for convection-diffusion equation. *Journal of Computational Physics*, 274:50–63, 2014.

[84] M. H. Protter and H. F. Weinberger. *Maximum Principles in Differential Equations*. Springer-Verlag, New York, 1999.

[85] C. V. Pao. *Nonlinear Parabolic and Elliptic Equations*. Springer-Verlag, New York, 1993.

[86] K. B. Nakshatrala and A. J. Valocchi. Non-negative mixed finite element formulations for a tensorial diffusion equation. *Journal of Computational Physics*, 228:6726–6752, 2009.

[87] R. Liska and M. Shashkov. Enforcing the discrete maximum principle for linear finite element solutions for elliptic problems. *Communications in Computational Physics*, 3:852–877, 2008.

[88] S. Harris. *An Introduction to the Theory of the Boltzmann Equation.* Dover Publications, Mineola, 2004.

[89] P. L. Bhatnagar, E. P. Gross, and M. Krook. A model for collision processes in gases. I. small amplitude processes in charged and neutral one-component systems. *Physical Review*, 94(3):511, 1954.

[90] X. He and L. Lou. Theory of the lattice Boltzmann method: From the Boltzmann equation to the lattice Boltzmann equation. *Physical Review E*, 56:6811, 1997.

[91] D. d'Humières. Multiple–relaxation–time lattice Boltzmann models in three dimensions. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 360(1792):437–451, 2002.

[92] P. A. Skordos. Initial and boundary conditions for the lattice Boltzmann method. *Physical Review E*, 48(6):4823, 1993.

[93] Q. Zou and X. He. On pressure and velocity boundary conditions for the lattice Boltzmann BGK model. *Physics of Fluids*, 9(6):1591–1598, 1997.

[94] Q. Chen, X. Zhang, and J. Zhang. Improved treatments for general boundary conditions in the lattice Boltzmann method for convection-diffusion and heat transfer processes. *Physical Review E*, 88(3):033304, 2013.

[95] R. Courant, K. Friedrichs, and H. Lewy. ber die partiellen Differenzengleichungen der mathematischen Physik. *Mathematische Annalen*, 100(1):32–74, 1928.

[96] R. Huang and H. Wu. Lattice Boltzmann model for the correct convection-diffusion equation with divergence-free velocity field. *Physical Review E*, 91(3):033302, 2015.

[97] B. Servan-Camas and F. T. C. Tsai. Non-negativity and stability analyses of lattice Boltzmann method for advection–diffusion equation. *Journal of Computational Physics*, 228(1):236–256, 2009.

[98] C. Cercignani. *The Boltzmann Equation and its Applications*. Springer, New York, 1988.

[99] S. Dellacherie. Construction and analysis of lattice Boltzmann methods applied to a 1D convection-diffusion equation. *Acta Applicandae Mathematicae*, 131(1):69–140, 2014.

[100] I. V. Karlin, S. Ansumali, C. E. Frouzakis, and S. S. Chikatamarla. Elements of the lattice Boltzmann method I: Linear advection equation. *Communications in Computational Physics*, 1(4):616–655, 2006.

[101] S. Bekri, J. F. Thovert, and P. M. Adler. Dissolution of porous media. *Chemical Engineering Science*, 50(17):2765–2791, 1995.

[102] B. D. Wood, K. Radakovich, and F. Golfier. Effective reaction at a fluid–solid interface: applications to biotransformation in porous media. *Advances in Water Resources*, 30(6):1630–1647, 2007.

[103] T. L. Van Noorden and I. S. Pop. A Stefan problem modeling crystal dissolution and precipitation. *IMA Journal of Applied Mathematics*, 73(2):393–411, 2008.

[104] M. Shapiro and H. Brenner. Taylor dispersion of chemically reactive species: irreversible first-order reactions in bulk and on boundaries. *Chemical Engineering Science*, 41(6):1417–1433, 1986.

[105] E. M. Murphy and T. R. Ginn. Modeling microbial processes in porous media. *Hydrogeology Journal*, 8(1):142–158, 2000.

[106] C. M. Gramling, C. F. Harvey, and L. C. Meigs. Reactive transport in porous media: A comparison of model prediction with laboratory visualization. *Environmental Science & Technology*, 36(11):2508–2514, 2002.

[107] I. Battiato, D. M. Tartakovsky, A. M. Tartakovsky, and T. Scheibe. On breakdown of macroscopic models of mixing-controlled heterogeneous reactions in porous media. *Advances in Water Resources*, 32(11):1664–1673, 2009.

[108] I. Fatt. The network model of porous media. *Petroleum Transactions*, 207:144–181, 1956.

[109] J. J. Monaghan. Smoothed particle hydrodynamics. *Reports on Progress in Physics*, 68(8):1703, 2005.

[110] X. He and L. Luo. Theory of the lattice Boltzmann method: From the Boltzmann equation to the lattice Boltzmann equation. *Physical Review E*, 56(6):6811, 1997.

[111] C. Cercignani and M. Lampis. Kinetic models for gas-surface interactions. *Transport Theory and Statistical Physics*, 1(2):101–114, 1971.

[112] C. Cercignani. *The Boltzmann Equation and its Applications.* Springer, New York, 1988.

[113] C. Cercignani, R. Illner, and M. Pulvirenti. *The Mathematical Theory of Dilute Gases*, volume 106. Springer Science & Business Media, New York, 2013.

[114] T. D. Scheibe, E. M. Murphy, X. Chen, A. K. Rice, K. C. Carroll, B. J. Palmer, A. M. Tartakovsky, I. Battiato, and B. D. Wood. An analysis platform for multiscale hydrogeologic modeling with emphasis on hybrid multiscale methods. *Groundwater*, 53(1):38–56, 2015.

[115] I. Battiato, D. M. Tartakovsky, A. M. Tartakovsky, and T. D. Scheibe. Hybrid models of reactive transport in porous and fractured media. *Advances in Water Resources*, 34(9):1140–1150, 2011.

[116] T. Sun, Y. Mehmani, and M. T. Balhoff. Hybrid multiscale modeling through direct substitution of pore-scale models into near-well reservoir simulators. *Energy & Fuels*, 26(9):5828–5836, 2012.

[117] A. M. Tartakovsky, D. M. Tartakovsky, T. D. Scheibe, and P. Meakin. Hybrid simulations of reaction-diffusion systems in porous media. *SIAM Journal on Scientific Computing*, 30(6):2799–2816, 2008.

[118] M. T. Balhoff, S. G. Thomas, and M. F. Wheeler. Mortar coupling and upscaling of pore-scale models. *Computational Geosciences*, 12(1):15–27, 2008.

[119] Y. Mehmani and M. T. Balhoff. Bridging from pore to continuum: A hybrid mortar domain decomposition framework for subsurface flow and transport. *Multiscale Modeling & Simulation*, 12(2):667–693, 2014.

[120] Y. Tang, A. J. Valocchi, and C. J. Werth. A hybrid pore-scale and continuum-scale model for solute diffusion, reaction, and biofilm development in porous media. *Water Resources Research*, 51(3):1846–1859, 2015.

[121] P. Albuquerque, D. Alemani, B. Chopard, and P. Leone. Coupling a lattice Boltzmann and a finite difference scheme. In *Computational Science-ICCS 2004*, pages 540–547. Springer, 2004.

[122] I. W. Haslam, R. S. Crouch, and M. Seaïd. Coupled finite element–lattice Boltzmann analysis. *Computer Methods in Applied Mechanics and Engineering*, 197(51):4505–4511, 2008.

[123] M. Astorino, F. Chouly, and A. Quarteroni. A time-parallel framework for coupling finite element and lattice Boltzmann methods. Research report, LMB - CMCS - MOX, 2014.

[124] M. Günther, A. Kvaernø, and P. Rentrop. Multirate partitioned Runge-Kutta methods. *BIT Numerical Mathematics*, 41(3):504–514, 2001.

[125] E. M. Constantinescu and A. Sandu. Extrapolated multirate methods for differential equations with multiple time scales. *Journal of Scientific Computing*, 56(1):28–44, 2013.

[126] A. Lew, J. E. Marsden, M. Ortiz, and M. West. Variational time integrators. *International Journal for Numerical Methods in Engineering*, 60(1):153–212, 2004.

[127] S. Karimi and K. B. Nakshatrala. A monolithic multi-time-step computational framework for first-order transient systems with disparate scales. *Computer Methods in Applied Mechanics and Engineering*, 283:419–453, 2015.

[128] T. Bridges and S. Reich. Multi-symplectic integrators: Numerical schemes for Hamiltonian PDEs that conserve symplecticity. *Physics Letters A*, 284(4):184–193, 2001.

[129] B. Leimkuhler and S. Reich. *Simulating Hamiltonian Dynamics*. Cambridge University Press, Cambridge, U.K., 2004.

[130] K. B. Nakshatrala, K. D. Hjelmstad, and D. A. Tortorelli. A FETI-based domain decomposition technique for time-dependent first-order systems based on a DAE approach. *International Journal for Numerical Methods in Engineering*, 75(12):1385–1415, 2008.

[131] A. Quarteroni and A. Valli. *Domain Decomposition Methods for Partial Differential Equations*. Oxford University Press, Oxford, 1999.

[132] T. Mathew. *Domain Decomposition Methods for the Numerical Solution of Partial Differential Equations*. Springer, Berlin Heidelberg, 2008.

[133] B. I. Wohlmuth. A mortar finite element method using dual spaces for the Lagrange multiplier. *SIAM Journal on Numerical Analysis*, 38(3):989–1012, 2000.

[134] T. Arbogast, G. Pencheva, M. F. Wheeler, and I. Yotov. A multiscale mortar mixed finite element method. *Multiscale Modeling & Simulation*, 6(1):319–346, 2007.

[135] P. L. Lions. On the Schwarz alternating method. i. In *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 1–42. Paris, France, 1988.

[136] X. C. Cai and Y. Saad. Overlapping domain decomposition algorithms for general sparse matrices. *Numerical Linear Algebra with Applications*, 3(3):221–237, 1996.

[137] P. M. Gresho and R. L. Sani. *Incompressible Flow and the Finite Element Method: Advection-Diffusion*, volume 1. John Wiley & Sons, Inc., Chichester, UK, 2000.

[138] W. H. Hundsdrofer and J. G. Verwer. *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations.* Springer-Verlag, New York, USA, 2007.

[139] O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method*. McGraw-Hill, London, 1977.

[140] L. P. Franca, S. L. Frey, and T. J. R. Hughes. Stabilized finite element methods: I. application to the advective-diffusive model. *Computer Methods in Applied Mechanics and Engineering*, 95(2):253–276, 1992.

[141] M. Augustin, A. Caiazzo, A. Fiebach, J. Fuhrmann, V. John, A. Linke, and R. Umla. An assessment of discretizations for convection-dominated convection–diffusion equations. *Computer Methods in Applied Mechanics and Engineering*, 200(47):3395–3409, 2011.

[142] W. Xian and A. Takayuki. Multi-GPU performance of incompressible flow computation by lattice Boltzmann method on GPU cluster. *Parallel Computing*, 37(9):521–535, 2011.

[143] P. Lallemand and L. S. Luo. Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, Galilean invariance, and stability. *Physical Review E*, 61(6):6546, 2000.

[144] S. Karimi and K. B. Nakshatrala. Do current lattice Boltzmann methods for diffusion and diffusion-type equations respect maximum principles and the non-negative constraint? *arXiv:1503.08360*, 2015.

[145] Y. Li, R. Shock, R. Zhang, and H. Chen. Numerical study of flow past an impulsively started cylinder by the lattice-Boltzmann method. *Journal of Fluid Mechanics*, 519:273–300, 2004.

[146] I. V. Karlin, S. Ansumali, C. E. Frouzakis, and S. S. Chikatamarla. Elements of the lattice Boltzmann method i: Linear advection equation. *Communications in Computational Physics*, 1(4):616–655, 2006.

[147] S. Dellacherie. Construction and analysis of lattice Boltzmann methods applied to a 1D convection-diffusion equation. *Acta Applicandae Mathematicae*, 131(1):69–140, 2014.

[148] Q. Zou and X. He. On pressure and velocity boundary conditions for the lattice Boltzmann BGK model. *Physics of Fluids*, 9(6):1591–1598, 1997.

[149] D. P. Ziegler. Boundary conditions for lattice Boltzmann simulations. *Journal of Statistical Physics*, 71:1171–1177, 1993.

[150] D. Yu, R. Mei, L. Luo, and W. Shyy. Viscous flow computations with the method of lattice Boltzmann equation. *Progress in Aerospace Sciences*, 39(5):329–367, 2003.

[151] X. Jiao and M. T. Heath. Common-refinement-based data transfer between non-matching meshes in multiphysics simulations. *International Journal for Numerical Methods in Engineering*, 61(14):2402–2427, 2004.

[152] R. K. Jaiman, X. Jiao, P. H. Geubelle, and E. Loth. Conservative load transfer along curved fluid–solid interface with non-matching meshes. *Journal of Computational Physics*, 218(1):372–397, 2006.

[153] A. de Boer, A. H. Van Zuijlen, and H. Bijl. Review of coupling methods for non-matching meshes. *Computer Methods in Applied Mechanics and Engineering*, 196(8):1515–1525, 2007.

[154] G. Chesshire and W. D. Henshaw. Composite overlapping meshes for the solution of partial differential equations. *Journal of Computational Physics*, 90(1):1–64, 1990.

[155] W. D. Henshaw and D. W. Schwendeman. An adaptive numerical scheme for high-speed reactive flow on overlapping grids. *Journal of Computational Physics*, 191(2):420–447, 2003.

[156] F. Olsson and N. A. Petersson. Stability of interpolation on overlapping grids. *Computers and Fluids*, 25(6):583–605, 1996.

[157] I. Babuška and M. Suri. The optimal convergence rate of the p-version of the finite element method. *SIAM Journal on Numerical Analysis*, 24(4):750–776, 1987.

[158] W. L. Wood. *Practical Time-stepping Schemes*. Clarendon Press, Oxford, 1990.

[159] S. van der Walt, S. C. Colbert, and G. Varoquaux. The NumPy array: A structure for efficient numerical computation. *Computing in Science & Engineering*, 13:22–30, 2011.

[160] A. Logg, K. Mardal, and G. Wells. *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book*. Springer, Berlin Heidelberg, 2012.

[161] F. Nataf and F. Nier. Convergence rate of some domain decomposition methods for overlapping and nonoverlapping subdomains. *Numerische Mathematik*, 75(3):357–377, 1997.

[162] H. Chang, X. Tai, L. Wang, and D. Yang. Convergence rate of overlapping domain decomposition methods for the Rudin–Osher–Fatemi model based on a dual formulation. *SIAM Journal on Imaging Sciences*, 8(1):564–591, 2015.

[163] J. I. Drever. *The Geochemistry of Natural Waters*. Prentice Hall, New Jersey, 1988.

[164] M. M. Benjamin. *Water Chemistry*. Waveland Press Inc., Long Grove, 2002.

[165] K. B. Nakshatrala, M. K. Mudunuru, and A. J. Valocchi. A numerical framework for diffusion-controlled bimolecular-reactive systems to enforce maximum principles and the non-negative constraint. *Journal of Computational Physics*, 253:278–307, 2013.

[166] S. Karimi and K. B. Nakshatrala. A hybrid multi-time-step framework for pore-scale and continuum-scale modeling of solute transport in porous media. *arXiv:1601.02708v2*, 2016.