ON THE APPLICATION OF NUMERICAL CONTINUATION TO LARGE-SCALE DYNAMICAL SYSTEMS

BY

MEHDI SAGHAFI

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Mechanical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2014

Urbana, Illinois

Doctoral Committee:

Professor Harry Dankowicz, Chair
Professor Ian Hiskens, University of Michigan
Associate Professor Prashant Mehta
Associate Professor Matthew West
ABSTRACT

The system approach in dealing with large groups of components with nonlinear interactions has received constantly growing attention in recent years in wide variety of applications from system biology to the size evolution of aerosol particles. These systems exhibit complex behaviors that are not predictable unless the entire system as well as their interactions are taken into account. The complexity of these high-dimensional problems is exacerbated when parameter space of interest is also sizable. The focus of this dissertation is addressing several physical and engineering problems within the context of large-scale dynamical systems through the development as well as implementation of novel computational algorithms based on numerical path-following techniques. For all systems investigated in this study, a suitable boundary value problem is formulated and the numerical continuation method is used for covering the parameter space using the software-package coco.

In this dissertation, we first briefly consider the reachability problem in power system networks in which the uncertainties in the loads require robust design of system parameters to ensure safe operation of the network under undesired perturbation in the system. This analysis is based on an existing MATLAB-code which utilizes a so-called numerical shooting method for the parametric analysis of power systems. A wrapper is developed to make the custom-designed systems in this code compatible with the asynchronous collocation toolbox in coco.

The asynchronous collocation toolbox developed in this dissertation extends the capability of common implementation of collocation methods for differential equations by reducing the number of mesh points required specifically for problems with slow-fast dynamics. This class of dynamical systems appears in many large-scale physical systems in which changes of states are governed by different time scales.
The transient growth of aerosol particles in a humid environment is then investigated in order to explore the effect of some system parameters, such as the initial size distribution and the rate of temperature decay, on the formation of cloud droplets. For a fixed fraction of droplet forming particles in a rising parcel, a suitably formulated boundary value problem is used along with numerical continuation to obtain the solutions in the parameter space of interest. Results obtained from the proposed numerical scheme are compared with the estimated fractions from available criteria in the literature to show the estimation error due to a phenomenon known as kinetic limitation.

In the second part of this dissertation, our objective is to identify and, where possible, resolve singularities that may arise in the discretization of spatiotemporal boundary-value problems governing the steady-state behavior of nonlinear beam structures. Of particular interest is the formulation of nondegenerate continuation problems of a geometrically-nonlinear model of a slender beam, subject to a uniform harmonic excitation, that may be analyzed numerically in order to explore the parameter-dependence of the excitation response. Several methods for breaking both the spatial and temporal equivariance are proposed. We then use these findings to suitably discretize the mixed formulation of the governing equations of a beam in the longitudinal and transverse directions where inertia effects are taken into account. The numerical results corresponding to the free vibrations are compared to a perturbation analysis obtained using the multiple-scales method to show the validity of the numerical scheme. The developed numerical technique is then employed to investigate the geometry as well as parameter dependence of the range of the linear regime in the beam’s forced response.

Finally, the convergence associated with the collocation scheme is investigated in order to rigorously analyze the capabilities of this method. The dissertation ends with a discussion of further development of asynchronous discretization scheme as well as several physical questions to be addressed.
To my mother
ACKNOWLEDGMENTS

As I look back at around four and half years ago when I started the PhD program here at the University of Illinois, the magnitude of changes, mostly for good, is overwhelming. I went from a passionate student whose main hobby was thinking to a person whose main hobby is thinking and has a better idea about what his passions are. This intellectual growth was not possible without my experience during the past several years with all the ups and downs. I am thankful to all the people who formed this experience and, as is common in practice but special in my heart, I would like to take this opportunity to express my gratitude to them. This journey has not ended yet, and hopefully will not very soon, but the experiences therein and the people who made it enjoyable will always remain special to me.

I would like to first thank my advisor, Prof. Harry Dankowicz, for being there when I needed it the most and giving me the liberty I longed for. He has been a constant source of guidance and encouragement and, I have benefited tremendously not only from his vast and deep knowledge of the state of the art science, but also from his insight into the systematic approach in conducting scholarly research. I also thank the members of my doctoral committee: Prof. Matthew West for his openness to lengthy discussions and his generosity in sharing his incredible source of knowledge; Prof. Ian Hiskens for his hospitality during my visits to Ann Arbor as well as his insights into power systems and access to his computational code; and Prof. Prashant Mehta for his valuable comments and advice. My gratitude also goes to the long list of friends, specially my fellow labmates in the Applied Dynamics Laboratory.

I was not fortunate enough to have my father to share this momentous period of my life with. However, I am grateful for being part of an amazing family. My parents, Morteza and Maliheh, instilled in me the joy of free thinking and I am forever indebted to them for giving
me the opportunity to freely shape my thoughts with their constant support and love. I also thank my brothers, Ahmad and Hamed, for giving me the great joy of brotherhood. My sincere thanks go to my wife, Elham, for her love and support throughout these last years. I have enjoyed, and will do in the years to come, the many serious discussions we had about unimportant stuff and colloquial arguments on scientific subjects.

This material is based upon work supported by the National Science Foundation under Grant No. 1016467. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.
# TABLE OF CONTENTS

## CHAPTER 1 INTRODUCTION

1.1 Large-scale dynamical systems ........................................... 1
1.2 Boundary-value problems .................................................. 6
1.3 Numerical continuation ..................................................... 7
1.4 Numerical algorithms ....................................................... 8
1.5 Dissertation outline ....................................................... 12

## CHAPTER 2 CONTINUATION TOOLBOX FOR POWER SYSTEMS

2.1 Introduction ................................................................. 15
2.2 A MATLAB toolbox based on numerical shooting method ............... 16
2.3 Power system toolbox - a boundary-value formulation approach ....... 19
2.4 A test problem ............................................................. 33
2.5 Summary ................................................................. 34

## CHAPTER 3 PARAMETER-DEPENDENT DYNAMICS OF AEROSOLS

3.1 Introduction ................................................................. 35
3.2 Model description .......................................................... 37
3.3 Numerical study ............................................................ 57
3.4 Conclusions ................................................................. 68

## CHAPTER 4 SINGULARITIES IN THE GOVERNING FORMULATION OF BEAM STRUCTURES

4.1 Introduction ................................................................. 71
4.2 A scalar beam model ....................................................... 73
4.3 A mixed formulation ....................................................... 82
4.4 Numerical results .......................................................... 96
4.5 Conclusions ................................................................. 101

## CHAPTER 5 COUPLED DYNAMICS OF BEAM STRUCTURES

5.1 Introduction ................................................................. 103
5.2 Model derivation ........................................................... 105
5.3 Perturbation analysis ....................................................... 110
5.4 Mixed formulation ........................................................ 118
5.5 Performance of the numerical method ................................... 121
5.6 Parametric study of the dynamic range ................................ 122
CHAPTER 1

INTRODUCTION

This dissertation investigates several problems in large-scale dynamical systems where the unifying theme in the corresponding algorithms is the use of numerical continuation within the context of boundary-value problems. For each application, a proper inverse problem is formulated to address specific physical questions. Then, in order to analyze the formulated problem, computational algorithms are developed to overcome the shortcomings of the existing tools. In this chapter, we introduce several core concepts used in this dissertation and briefly discuss the characteristics of the problems studied therein.

1.1 Large-scale dynamical systems

With recent developments of computer capabilities, many large-scale problems that seemed intractable are now easily handled using new advanced techniques [1]. Nowadays, problems with thousands of unknowns appear in many applications, from linear programming to dynamical systems. Interestingly, the size of the problems seems to be increasing even faster than the state of the art techniques for computational analysis. This trend demands for more advanced solution methods that can overcome the size issue of the large-scale systems and provide a reliable solution via a robust mathematical and computer programming method.

The physical applications that are investigated in this dissertation include:

1. Power systems
2. Aerosol particles
3. Mechanical structures
We are interested in the analysis of the behavior of these systems as they evolve over time. In power systems, a large group of electrical as well as mechanical components are connected to each other to form a network. Although the coupling can be sparse, the interaction among different components can be intricate. The difference in the nature of the building components also adds to the complexity of these systems which, collectively, leads to a large system with potentially complex behavior. Such behavior is as much a function of coupling as it is a function of the properties of the individual components. In power networks, the dynamic behavior is not only nonlinear but may also be non-smooth or even discontinuous. The source of such discontinuity is either in the governing equations or the states.

In studying the droplet-forming potential of aerosol particles, we consider a problem of mass transfer. In particular, a large ensemble of aerosol particles is considered in a humid environment where the particles compete to absorb the ambient humidity and grow in size. Each particle may undergo phases of condensation as well as evaporation as this process of mass transfer occurs within a limited source of humidity. Here, again, the coupling is sparse and indirect among particles. However, during each phase the time scale for variation may differ substantially across the population of particles, leading to a complicated system.

Finally, we study the time behavior of mechanical structures when subjected to external forces. The response of such structures is a function of both space and time. The corresponding models are often so complicated that several simplifying assumptions need to be taken into account to make the analysis more amenable to common techniques. Recent applications to mechanical structures in very small scales, however, call for more accurate analyses with fewer model simplifications. As more advanced numerical techniques are employed for the analysis of such structures, more computational resources are required.

The time behavior, i.e. dynamics, of all these systems can be predicted using mathematical models obtained from physical laws. Interestingly, the dynamics of all these systems is governed by differential equations in the forms of Ordinary Differential Equations (ODEs), Partial Differential Equations (PDEs), and Differential-Algebraic Equations (DAEs).

The dynamics of power networks is governed by DAEs. In particular, according to Ref. [2], the differential equations represent the dynamics of generators, controllers, devices, and dynamic loads, while the algebraic equations represent the power balance at the buses in the
network. The discontinuities in the system are caused by state-driven events, i.e., events are triggered when the states reach certain values rather than at some predetermined times.

Partial differential equations also describe the dynamic behavior of the growth of aerosol particles. These equations are then discretized in spatial domain to obtain a system of ordinary differential equations. In our study, a scalar ordinary differential equation governs the dynamics of the diameter of a single particle. An additional scalar ordinary differential equation for the relative humidity couples the dynamics of all the particle diameters.

Finally, the spatio-temporal dynamics of mechanical structures is originally governed by PDEs. Here, we differentiate these equations with respect to two independent variables; time and space. Although the number of unknown displacement fields may be small, proper discretizations of spatial domain may lead to a high-dimensional system of ODEs. In particular, we use the Finite Element Method (FEM) to discretize the original PDEs. The resulting equations are in the form of ODEs in which the size of the system is equal to the number of discretization points in the FEM. To achieve a set of equations that represents the dynamics of the mechanical structures at discrete points close to the original PDEs, large number of discretization points may be required which, in turn, increases the size of the problem. After discretization, the governing equations may be ODEs or DAEs.

The sizes of the problems considered in this dissertation are also affected by the discretization of the differential equations in time. More specifically, a more accurate approximate solution is often obtained when larger number of discretization points are appropriately selected. This leads to a trade-off between accuracy and computational expense. For instance, in the analysis of mechanical structures, the differential equations obtained from the discretization of the original PDEs have relatively small number of unknowns. However, the discretization in time domain can significantly increase the size of the computational problem for proper approximation of the solution.

As discussed in this section, differential equations in the forms of PDEs, ODEs, and DAEs play an important role in the analyses of systems studied in this dissertation as their governing equations. We proceed to review these forms of equations and, particularly, emphasize possible differences in the time-scales of the dependent variables.
1.1.1 Dynamical systems

Many systems evolve in time as a result of coupling as well as the interaction of the system with its environment. These systems are called dynamical systems and can be described by their governing differential equations. In this thesis, the dynamical systems concerned can be described by equations of the form

\[ A \cdot \dot{x} = f(x) \] (1.1)

In Eq. (1.1), \( x \) and \( f \) are \((n \times 1)\) vectors, and overdot denotes differentiation with respect to time, \( t \). Also, \( A \) is an \( n \times n \) constant matrix. The right-hand side of Eq. (1.1) is called the vector field and the equations are autonomous since the vector field is not changing over time. A more general description is in implicit form, i.e.,

\[ h(t, x, \dot{x}) = 0 \] (1.2)

An autonomous system can be obtained if the dependence of \( h \) on time is removed. Equation (1.1) is a special case of Eq. (1.2) if \( h \) is linear in \( \dot{x} \).

In Eq. (1.1), \( A \) is a square matrix and, hence, can be written in Jordan Normal Form. If \( A \) is non-singular, we can obtain Eq. (1.1) in explicit first-order form which is a special form of ODEs, through premultiplication of this equation by the inverse of matrix \( A \). A singularly perturbed set of differential equations is obtained when one (or more) of the eigenvalues in the Jordan form are very close to zero. A simple representation of these equations with dependent variables denoted by \( x \) and \( y \) is as follows.

\[ \dot{x} = f(x, y) \] (1.3)
\[ \epsilon y = g(x, y) \] (1.4)

with \( \epsilon \ll 1 \) a constant. Here, the variables \( x \) and \( y \) show different time scales in their behaviors. In particular, \( y \) is a fast variable, whereas \( x \) is a slow variable. Such time-scale separation, although very common in many physical systems, may pose difficulties in the
numerical treatment of these systems. In fact, if the solution is on the manifold defined by
\( g(x, y) = 0 \), then the fast equation becomes stationary. However, any small perturbation
from this manifold will cause a fast response in \( y \). We may then consider the case in which
\( \epsilon = 0 \) and, \( y \) is completely slaved to \( x \) through \( g(x, y) = 0 \), which results in the Differential-
Algebraic form of equations as
\[
\begin{align*}
\dot{x} &= f(x, y), \\
0 &= g(x, y),
\end{align*}
\]
(1.5)
(1.6)

As was discussed in this chapter, some physical problems that are investigated in this dis-
sertation are either formulated in DAE form using physical laws or mathematically rewritten
in this form for the sake of computational efficiency.

We use the notion of index as a measure for the level of singularity of DAEs. There are
several definitions for the index of DAEs, all of which can be related to each other. For
instance, a differential index is the minimum number of differentiations required to arrive
at a set of differential equations from DAEs. We use this definition in this thesis for the
subsequent analyses. Yet, another useful definition for the index of DAEs is a sensitivity
measure of the solution to perturbation in the equations and is noted as perturbation in-
dex \([3]\). Gear \([4]\) investigated the relationships between these two definitions of indices and
showed
\[
differential \text{ index} \leq \text{perturbation index} \leq differential \text{ index} + 1 \quad (1.7)
\]

From the geometrical perspective, the algebraic equations in DAEs define a manifold on
which the solutions are constrained to lie. The simplest form of DAEs is of index-1 in which
\( \frac{\partial y}{\partial y} \) has a bounded inverse. Following the Inverse Function Theorem, in principle, it is possible
to locally solve for \( y \) in Eq. (1.6) and, substitute in Eq. (1.5) to arrive at a set of first-order
differential equations, however, in practice, the original DAEs are often used in the analysis.
1.2 Boundary-value problems

Given an initial condition at time $t_0$ for a system of differential equations, we can always find a solution on its neighborhood provided that the vector field is continuously differentiable. Numerical methods may be formulated for computing values of the unknowns at discrete points near $t_0$. It is very common to use a *quadrature rule* to find discretization values at these points in which, at each step, one or more prior discretization values are used to find the value at an instance of time. If the desired solution is explicitly defined as function of solutions at prior points then the corresponding numerical scheme is called *explicit*. However, the desired discretization values in some numerical schemes, called *implicit*, is defined as a (nonlinear) function of prior discretization values which requires an additional effort to solve this equation. Although computationally less cumbersome, explicit methods tend to suffer from many numerical issues in their implementations and, hence, are often avoided.

On the other hand, a solution to a set of differential equations may be sought such that it satisfies a particular condition among some solution points. In a simple form, we may want the solution to be equal to a specified value at the end point of its domain, i.e.,

\[
\dot{x} = f(x), \quad 0 \leq t \leq T \tag{1.8}
\]

such that

\[
x(0) = x_0, \quad x(T) = x_1 \tag{1.9}
\]

Equations (1.8,1.9) define a *boundary value problem*. Unlike the initial value problems, a solution does not necessarily exist for this problem.

In the absence of analytical solutions to boundary-value problems, we may seek a numerical solution at discrete points defined in $0 \leq t \leq T$. One powerful approach for this approximation is the utilization of *collocation method*. In this method, which is a subclass of finite-difference methods, an approximate solution is sought such that the differential equations are satisfied at some discrete points, also called the *collocation nodes*. In order to find the value of the derivatives at the collocation nodes, the solutions are often approximated by piecewise polynomials. In particular, the solution domain is divided into several intervals,
within each interval a polynomial of some order is considered to approximate the solution. These polynomials are explicitly defined by their values at some basepoints. Subsequently, their derivatives are evaluated along with the corresponding values at collocation nodes. In addition to these collocation conditions, the satisfaction of boundary conditions are also imposed. Collectively, a set of (nonlinear) algebraic equations are obtained which is then solved using a nonlinear solver, e.g. Newton’s method.

Collocation methods are very efficient for approximating the solutions of moderate-size differential equations. As a result, this method has been successfully used to find approximate solutions of properly formulated boundary-value problems. In practice, we may need to find such approximate solutions under variations in system parameters by employing a numerical continuation method. The formulation for numerical continuation is a set of nonlinear algebraic equation which is appended to the algebraic equations from the collocation conditions. In the next section, we briefly discuss the main ideas behind the numerical continuation method.

### 1.3 Numerical continuation

A large number of mathematical models are characterized by implicitly-defined manifolds. Continuation methods are computational techniques for successive construction of discrete points on such manifolds. Consider, for example, the equation

\[ F(u) = 0, \quad F : \mathbb{R}^a \rightarrow \mathbb{R}^b \]  

(1.10)

where \( a \geq b \geq 1 \), and \( F \) is continuously differentiable. If \( F(u^*) = 0 \) and the Jacobian of \( F \) at \( u^* \) is nonsingular, then Eq. (1.10) defines a local \((b - a)\)-dimensional manifold where the difference \( d = b - a \), i.e. the excess of unknowns to equations, is called the dimensional deficit. In this context, we refer to Eq. (1.10) as a (continuation) zero problem in the vector \( u \) of continuation variables [5].

Among different covering algorithms for one-dimensional covering, the pseudo-arclength continuation method has been the most successful one. In this method, curves are approx-
imately parameterized by the arclength along their tangent line. Many software packages employ this method for manifold covering. This is also the case for several of the algorithms employed by coco, an open-source software package for continuation developed by Harry Dankowicz and Frank Schilder. All the numerical algorithms developed in this dissertation are implemented within this software package. This software can be download from http://sourceforge.net/projects/cocotools.

Although the collocation method has been employed extensively, within the context of numerical continuation, to obtain approximate solutions of differential equations, there are still challenges in their application to large-scale systems. In particular, the method is not efficient for large-scale systems with slow-fast dynamics. Hence, the development of advanced numerical algorithms is required in order to modify the collocation method to deal with such challenges. In the next section, we overview the basics of numerical algorithms and highlight some of the aspects pursued in this dissertation.

1.4 Numerical algorithms

Although large-scale problems possess some of the same characteristic as small systems in terms of coupling between components, new phenomena may occur that pose additional challenges to their study. Qualitative challenges are associated with the detection and analysis of emergent behaviors resulting from the nonlinear interactions of several components of these systems. Such systems are called complex systems in which their behavior is not predictable from an analysis of their components. In the latter type of challenges, however, the emphasis is on computational treatment of large-scale systems. In Ref. [6] these issues are categorized as

- storage: Very large-scale systems can take petabytes of memory. The challenge is in the storage as well as retrieval of the data in a timely manner. It is important to have cost-effective solutions in terms of both the hardware which is going to be used for the storage of data and the managing of the processes required for the storage as well as retrieval of data.
- **computational complexity**: Complexity refers to the rates at which the number of required computations grows with the problem size. In complex large-scale systems, the number of required computational steps may be so large that they exceed available computational resources.

- **accuracy**: Large-scale systems are more likely to be ill-conditioned, i.e., small perturbations in the input will cause large perturbations in the output. This may have an adverse effect on accuracy of such algorithms. This source of sensitivity is a characteristic of the system and not the computational algorithm which is used.

- **global error bounds**: Controlling the upper bound of the global error through a proper norm is a challenge in large-scale systems.

The focus of this dissertation is on the computational complexity and developing methods to reduce the computational resources required to numerically analyze large-scale dynamical system.

### 1.4.1 Complexity of numerical algorithms

The basic measure for arithmetic operations of an algorithm is called a *flop*, which is normally a multiplication followed by a summation. In any numerical algorithm, our wish is to minimize the total number of these operations, also known as **complexity** of an algorithm. In forward simulation problems, in which the solution starts at the initial point and proceeds step by step to the next point using one (or more) previous solutions, the complexity is affected both by the number of mesh points at which a solution is sought and the **tolerance** of an admissible solution which is normally a measure of accuracy. On the other hand, in boundary-value problems, in which the entire domain of the independent variable is evaluated at once, the **memory** limitation is also an important factor when deciding on the number of discretization points in addition to the complexity. The latter class of problems are sometimes referred to as *global methods*.

In general, there are two main classes of techniques to reduce the computational complexity in large-scale problems: reduction methods and partition methods. In the case of reduction
methods, the goal is to reduce the size of the problem to a level tractable by the available computer capabilities. There are two main approaches to reduce the size of a large problem:

- **reducing the number of unknowns**: In this approach, a smaller problem is sought in which the essential mechanisms governing the system behavior are retained. The small problem is often called the *reduced order model*.

- **reducing the complexity of the numerical algorithm**: In this approach, attention is restricted to the computational complexity of the solution procedure and methods are developed to reduce this complexity. In this thesis, we take this approach for developing new computational techniques.

In the case of partition methods, the problem is divided into several sub-problems where each one is treated separately, mostly in parallel, in order to avoid dealing with a large coupled system. With the increased popularity of using GPUs in parallel processing, partition methods offer a low-cost solution to the analysis of large-scale systems.

Partition methods treat the original problem with the speed-up gained though proper division of the original problem. On the other hand, reduction methods deal with a new problem. The reduction methods require that

- errors are comparable to the original system,

- the approach is computationally efficient,

- characteristics of the original computational algorithm are preserved.

Most often, a combination of the partition and reduction methods may be employed in the analysis of large-scale systems. However, in this dissertation, we restrict our attention to development of techniques for reducing the size of computational algorithms.

As the development of computational algorithms that make large-scale problems more tractable is highly desired, these algorithms are basically assessed by how accurately they can approximate an exact solution to the original problem. We proceed to a brief overview of numerical errors as related to the context of this dissertation.
1.4.2 Numerical errors

In the majority of boundary value problems, an analytical solution may not be available and, hence, an approximate solution of the original problem is sought through a proper numerical scheme. As indicated by the term ‘approximate’, errors are introduced in the numerical solution of these problems and the study of these errors as well as their bounds is of significant importance.

Errors in numerical schemes for the solution of differential equations can be categorized into three main types [7]:

1. *input error*

2. *round-off error*

3. *discretization error*

Input errors originate from prior measurements or calculations. Round-off errors originate from the use of floating point numbers and are defined as the difference between the numerical solution and the solution obtained in the case that exact arithmetic were used. Discretization errors, also known as *truncation errors*, are a consequence of the finiteness of the approximation space.

There is a reverse relationship between the round-off error and the discretization error. In a properly discretized problem, the latter would tend to zero as we decrease the mesh size. On the other hand, the round-off error is proportional to $\varepsilon_M$ as well as the mesh size, where $\varepsilon_M$ is called the *machine epsilon* and is the smallest number that if added to 1 will result in a number larger than 1. This also shows that smaller step sizes can be chosen to achieve reasonable round-off error provided that the $\varepsilon_M$ is sufficiently small.

The sources of error discussed in this section are also categorized as *local errors* as they are generated locally. For example, an input error can locally introduce an error in the initial values, while this local perturbation can be propagated throughout the entire solution interval to generate the *global error*.

We end this section with a brief overview of some main concepts in numerical approximation of boundary value problems. We may write all differential equations as a (nonlinear)
operator, i.e.,

\[ \mathbf{L} \mathbf{y}(x) \equiv \mathbf{q}(x), \quad a < x < b \]  

(1.11)

where \( \mathbf{L} \) is a (nonlinear) differential operator. The local truncation error \( \tau_i[\mathbf{y}] \) at mesh points is defined as [7]

\[ \tau_i[\mathbf{y}] := \mathbf{L}_\pi \mathbf{y}(x_i) - \mathbf{L} \mathbf{y}(x_i), \quad 1 \leq i \leq N \]  

(1.12)

where \( x_i \)'s and \( \mathbf{L}_\pi \) are the mesh points and the discretized operator corresponding to mesh \( \pi \), respectively. In other words, the local truncation error is the difference between the residuals of the exact and discretized operators acting on the exact solution evaluated at the mesh points.

Naturally, a useful way of characterizing the discretized operator is to study its convergence to exact operator as the number of mesh points increases. A method is called consistent if the discretized operator \( \mathbf{L}_\pi \) approaches the differential operator \( \mathbf{L} \) in the limit as the mesh size tends to infinity in some suitable norm.

The consistency of a numerical algorithm is required for the convergence of that method. However, in addition to this necessary condition, the method should also be stable to guarantee the convergence. Here, stability refers to the boundedness of the solution to the discretized problem in the presence of small perturbations. Note that stability is a characteristic of a scheme itself and not of the specific boundary value problem.

In the several next chapters of this dissertation, as outlined in the proceeding section, we employ the principles of the numerical analysis of boundary-value problems, as discussed in this chapter, for properly formulated systems to address physical problems in various applications.

1.5 Dissertation outline

In the next several paragraphs, we outline the contributions of each individual chapter.

In Chapter 2, the structure of power system models are discussed and the utilization of grazing phenomena is investigated in reachability analysis of those systems. Moreover, the basic framework of the software package COCO which is used for numerical continuation in
this dissertation is demonstrated through the explanation of a wrapper code that makes an available power system MATLAB code compatible with the format of the asynchronous collocation toolbox. This research was performed in collaboration with Prof. Ian Hiskens and Maxim Markov of the Department of Electrical Engineering and Computer Science at the University of Michigan at Ann Arbor, and partially documented in Ref. [8].

In Chapter 3, we investigate transient changes in the wet diameters of aerosol particles in a humid environment, in order to determine the fraction of particles that absorb sufficient water vapor to form into droplets. A numerical relationship between statistical properties of the initial distribution of aerosol particle sizes and the rate of temperature drop in a rising cloud parcel is obtained, for a fixed fraction of droplet forming particles, using numerical continuation methods applied to an asynchronous discretization of a suitably formulated boundary-value problem. The material in this chapter is collaborative efforts with Prof. Harry Dankowicz and Prof. Matthew West and based on a journal manuscript being prepared for submission.

In Chapter 4, the singularities in the mixed formulation of nonlinear structures are investigated and several approaches are proposed to overcome degeneracies in the resultant nonlinear algebraic equations. In the absence of theoretical bounds, computation is used to estimate convergence rates of the different discretization schemes, in the case of numerical calibration experiments performed on equilibrium and periodic responses for a linear beam, as well as for the full nonlinear models. The material in this chapter is collaborative efforts with Prof. Harry Dankowicz and accepted for publication in the ASME Journal of Computational and Nonlinear Dynamics.

Chapter 5 investigates the modeling of a slender beam in order to obtain the governing equations of the transverse and longitudinal displacements when subjected to external forces. In this model, rotary inertia effects are taken into account and it is shown, through both numerical analysis and perturbation analysis, that an amplitude-independent shift exists in the frequency response of the beam. Similarly, it is shown that a higher-order study reveals the onset of a softening effect for larger amplitudes in the frequency response curve. This chapter proceeds to a numerical formulation suitable for the continuation of the cusp point in the frequency response curve of the beam. This point is used to explore the parameter-
dependence of the range over which the beam behaves in a primarily linear way. The results can be used for designing layered micro/nano-resonators with desired ranges of linear behavior. The material in this chapter is collaborative efforts with Prof. Harry Dankowicz and Prof. Walter Lacarbonara and based on a journal manuscript being prepared for submission.

Finally, Chapter 6 briefly studies the convergence of the asynchronous collocation algorithm. The employed approach considers the boundary value problem as a nonlinear operator to obtain preliminary results on convergence rates.
2.1 Introduction

The energy security of the world relies on a transition to sustainable sources of energy generation. For this reason, renewable resources have attracted significant attention from the energy sector. The complexity of networks that generate such energy, however, presents several technical issues that challenge the successful utilization of such resources. These issues demand for complex designs of power networks in order to insure a robust and reliable performance.

Design of power networks is based on mathematical models of their behaviors. From Ref. [9], these models can be classified as linear/nonlinear, lumped/distributed parameter, static/dynamic, continuous/discrete, deterministic/stochastic. A power network consists of several of these models coupled together in a proper way. The governing equations of the dynamics of these networks are often in the form of differential-algebraic equations. In order to study the dynamics of power systems, forward simulation is often used to find the response of the system over time through proper discretization of the governing equations starting from an initial condition. It is common to use a fixed-time step trapezoidal rule for the discretization of the governing equations [9].

The dynamic analysis aims at investigating the influence of system parameters on the behavior of the network. There are several parameters included in the design of power networks whose values determine a stable and reliable performance of the network. In practice there are two types of problems which are directly influenced by these design parameters. In the first type, the vulnerability of the network due to uncertainties in the loads is investigated. In renewable generation, variability also exists in the generation of energy which significantly
challenges the current approaches for dynamic security assessment of power networks. In the second type, one is concerned with the performance of the network after a disturbance triggers a designated control unit. The control units send a trip signal which short-circuits the affected part of the network. If the design parameters are not properly tuned, this can lead to a cascade of failures in the network with possible catastrophic consequences.

Although brute-force forward simulations are often used for the analysis of power systems, the task of mapping out regions of safe performance requires more advanced tools. The so-called grazing phenomenon has been successfully shown to satisfy this need by setting conditions for defining a surface that separates safe and unsafe regions in parameter space [10]. Numerical continuation can then be employed to cover distinct points on this surface. This approach has been successfully demonstrated in Ref. [11] using a so-called numerical shooting method. Numerical shooting is a method for solving boundary-value problems by performing forward simulation from a given initial condition with a particular end condition in mind. In contrast, the application of a collocation scheme to a boundary-value problem imposes the entire discretization scheme all at once.

Numerical shooting is a powerful tool for locating grazing points, but is limited by issues of numerical stability and accuracy [12]. Hence, a natural choice is using the collocation method in which the entire problem is implicitly formulated. The former approach is the basis of an already existing software in MATLAB developed by Prof. Ian Hiskens at University of Michigan. The underlying theory behind this code is described in Refs. [10, 13]. In Section 2.2, we briefly discuss the structure of this code, and refer to it as the forward simulation code. In Section 2.3, we present the development of a wrapper code which enables the use of the existing library of power component models within a COCO toolbox for a collocation discretization of boundary-value problems. We refer to the latter as the collocation code.

2.2 A MATLAB toolbox based on numerical shooting method

Power systems are an important class of hybrid systems, as they are generally characterized by continuous dynamics as well as discrete events. The continuous dynamics are governed by physical laws that the components must obey. The discrete events correspond to logical
rules of the system. One approach for modeling these systems is to describe them by a set of differential, switched algebraic and state-reset (DSAR) equations (e.g. [14, 15]). This can be written as

\[ \dot{x} = f(x, y) \]  
\[ 0 = g^{(0)}(x, y) \]  
\[ 0 = \begin{cases} 
  g^{(i-)}(x, y) & y_{d,i} < 0 \\
  g^{(i+)}(x, y) & y_{d,i} > 0 
\end{cases} \quad i = 1, \ldots, d \]  
\[ x^+ = h_j(x^-, y^-) \quad y_{e,j} = 0 \quad j \in \{1, \ldots, e\} \]

where

\[ x = \begin{bmatrix} x \\ z \\ \lambda \end{bmatrix}, \quad f = \begin{bmatrix} f \\ 0 \\ 0 \end{bmatrix}, \quad h_j = \begin{bmatrix} x \\ h_j \\ \lambda \end{bmatrix} \]

and

- \( x \) are the continuous dynamic states
- \( z \) are discrete dynamic states
- \( y \) are algebraic states
- \( \lambda \) are parameters

The notations \( ' + ' \) and \( ' - ' \) denote the values of states just after or prior to a reset event, respectively. In this model, the differential variables are considered to be continuous and, the discontinuities are applied only to algebraic variables through the reset function, \( h_j \). There are two types of events in this model. One is switching of the algebraic equations which is triggered by \( y_d \). The other event is a state reset which is triggered by \( y_e \).

In the forward simulation code, the integration of this model is carried out through the application of the trapezoidal rule between the events. In the forward simulation code, forward integration of the model between events is governed by the following discretization
\[
\begin{align*}
    x^{k+1} &= x^k + \frac{\eta}{2} \left( f(x^k, y^k) + f(x^{k+1}, y^{k+1}) \right) \\
    0 &= g(x^{k+1}, y^{k+1})
\end{align*}
\]

where the superscript, \( k \), denotes the instant corresponding to time \( t_k \) and, \( \eta = t_{k+1} - t_k \) is the integration time step. Since this integration rule is implicit, an implicit set of nonlinear equations will be obtained and solved using an iterative Newton scheme.

An event does not generally occur at a mesh point, hence, extra conditions should be imposed to detect it. In the forward simulation code, an event is assumed to have occurred when a change in sign of the associated algebraic variables occurs between two consecutive mesh points. The last point is then ignored and, while \( \eta \) is treated as an unknown, the step is repeated with the additional condition \( y_i = 0 \) appended.

2.2.1 Implementation

The structure of the forward simulation code is modular, i.e., each system is composed of several subsystems that are connected together. In each subsystem, some algebraic variables are used as interface variables. These interface variables are then related through simple algebraic equations to link subsystems to each other. This modular structure allows the independent development of component models. Indeed, a library of models has been developed by Prof. Hiskens which can be used for constructing a large variety of power networks. Each model evaluates the differential (i.e. vector field) and algebraic equations as well as their Jacobians.

For any power network, the structure of the network should be provided in a specific format and stored in a file to be called by the forward simulation code. In particular, a cell structure should be provided where in each row we have

\[
'id', 'model', d_{ini}, a_{ini}, p_{ini}
\]

where 'model' is the file name of a specific model in the library, 'id' is an identification name we assign to this model in our structure, and, \( d_{ini}, a_{ini}, \) and \( p_{ini} \) are row vectors
of real numbers that set the initial values of the differential, algebraic, and parameter values, respectively. After this information is provided for all the models, a different cell structure defines the connections among models. As was mentioned earlier, interface variables are always algebraic variables. Each connection is set by

'\text{id}1', \text{idx}1, '\text{id}2', \text{idx}2, '\text{id}3', \text{idx}3, \ldots

This defines a simple connection equation. In particular, each connection is defined by a set of pairs. Each pair, e.g. 'id1', idx1, corresponds to the identification name of the model and the index of an algebraic variable, respectively. Denoting the value of this algebraic variable by $c_1$, we have for this connection

$$c_1 \pm c_2 \pm c_3 \pm \ldots = 0 \quad (2.8)$$

where the minus sign is used when the index of the variable has negative sign.

Although the modularity of the forward simulation code facilitates the development of power networks, its hierarchical structure makes this code more complicated. In the next section, we discuss the development of a wrapper code that calls the models in the existing library with appropriate arguments within the context of asynchronous collocation and, translates the output arguments such that a boundary-value problem is formed in COCO.

2.3 Power system toolbox - a boundary-value formulation approach

In this section, the general framework of the collocation code as well as the actual implementation is discussed. In this code, the wrapper calls the relevant models in the power systems library. An initial solution guess, as is required for initiation of the continuation procedure, is obtained through forward simulation of the corresponding system.
2.3.1 Initial solution guess

Similar to all numerical continuation problems, an initial solution in the vicinity of the solution manifold of the continuation problem is required to start the algorithm. In order to provide COCO with such a solution in the power system toolbox, we perform a forward simulation in the forward simulation code whose output contains the time solution of both differential and algebraic variables as well as an ordered set of indices corresponding to the events occurred during the simulation. The simulation is performed through the following function call.

\[ [x, y, -, -, time, evs] = \text{simulate}(h, final, 1, [], [], [], plt); \]

Here, \( h \) is the fixed step size for the numerical integration, \( \text{final} \) is the final time of integration, and \( \text{plt} \) indicates the variables to be plotted. For example, \( \text{plt} = [0 \ 0 \ 8 \ 7 \ 5] \) indicates that the output plot has its origin at \((0,0)\) and shows the corresponding values of the 7-th differential variable in the 8-th model over time while the plot is updated in every 5 steps. Note that a negative value corresponds to the indices of algebraic variables. The output arguments, \( x \) and \( y \) are the corresponding values of the differential and algebraic variables, respectively, of all models and \( \text{evs} \) stores the state of event variables. Note that the skipped input and output arguments in this call correspond to the sensitivity analysis performed by the forward simulation code but not used in the collocation code.

2.3.2 Implementation in COCO

Although an asynchronous discretization is not always required in the context of power system problems, the wrapper is developed so as to support this capability. The wrapper, in general, calls a hybrid system toolbox, which, in its own, calls the asynchronous collocation toolbox for the discretization and the imposition of collocation conditions of the governing DAEs. This toolbox accepts the vector field in the implicit form, where, in the power system problem, it is included in the MATLAB file called powersystem.m. This file evaluates the vector field as well as the algebraic connections corresponding to all models in a single segment along a multi-segment trajectory.
function [data, y] = powersystem(xx, yy, pp, inds, data)
% (c) Mehdi Saghafi, 2014

sdiff = sum(data.no_diff);
salg = sum(data.no_alg);

y1 = zeros(sum(data.NTST(data.async_ind(1:sdiff)))*data.NCOL,1);
y2 = zeros(sum(data.NTST(data.async_ind((sdiff+1):(sdiff+salg))))*data.NCOL,1);

mode = data.seg2mode(data.mode);

rang1 = 0;
rang2 = 0;

diff_inds = [0 cumsum(data.no_diff)];
alg_inds = sum(data.no_diff) + [0 cumsum(data.no_alg)];
par_inds = [0 cumsum(data.no_pars)];

for i = 1:data.no_of_models
    x_ind = inds((diff_inds(i)+1):diff_inds(i+1));
y_ind = data.fun_ind(data.g_ind{i});

    x_1 = xx((diff_inds(i)+1):diff_inds(i+1),x_ind);
    x_2 = xx((diff_inds(i)+1):diff_inds(i+1),y_ind);
    y_1 = xx((alg_inds(i)+1):alg_inds(i+1),x_ind);
    y_2 = xx((alg_inds(i)+1):alg_inds(i+1),y_ind);

    param_x = reshape(pp((par_inds(i)+1):par_inds(i+1)), data.par_shape{i}(1),...
        data.par_shape{i}(2));
    param_y = reshape(pp((par_inds(i)+1):par_inds(i+1)), data.par_shape{i}(1),...
        data.par_shape{i}(2));

    ev = data.unique_events(mode, :);

    o = feval(data.modtyp{i},x_2,y_2,1,ev(data.ev_range{i}),{param_y},0,i);
    f = feval(data.modtyp{i},x_1,y_1,1,ev(data.ev_range{i}),(param_x),1,i);
    g = feval(data.modtyp{i},x_2,y_2,1,ev(data.ev_range{i}),(param_y),4,i);

    % data.event_order{i} = data.alg_inds(i) + o(3:length(o));

    y1_ind = sub2ind(size(yy),(diff_inds(i)+1):diff_inds(i+1),x_ind);
    y1(range1+1:range1+length(f)) = cell2mat(yy(y1_ind')) - f;
    y2(range2+1:range2+length(g)) = g;

    range1 = range1 + length(f);
    range2 = range2 + length(g);
end

for k1 = 1:size(data.connec,1)
    tmp = cell2mat(xx(abs(data.connec{k1,:})),data.fun_ind(data.conn_ind(k1))');
    sign1 = sign(cell2mat(data.connec(k1,:)));
    tmp = repmat(sign1, size(tmp,1), 1);
    h = sum(tmp,2);
    y2(range2+1:range2+length(h)) = h;
    range2 = range2 + length(h);
end

y = [y1;y2];
end

Here, the input arguments indicate that the basepoint values of the all variables and their corresponding first derivatives are provided separately, denoted by xx and yy, respectively. The third input argument, pp, is an $n_p \times 1$ vector of parameter values, where $n_p$ is the
number of system parameters. The last input, \texttt{data}, is a structure that carries some local information to the vector field. Finally, \texttt{inds} is an \((n_a + n_d) \times 1\) vector where \(n_a\) and \(n_d\) denote the number of algebraic and differential variables in the power system, respectively. This vector contains the information for the asynchronous discretization. For instance, if the \(i\)-th value of \texttt{inds} is equal to \(k \in \{1, \ldots, p\}\), and \(p\) is the total number of distinct mesh groups, then the \(k\)-th mesh is used for the discretization of the \(i\)-th variable. More details on the implementation of the asynchronous collocation algorithm is provided in Appendix B. The \texttt{data} structure has several fields which are briefly described as follows. Note that \(n_{\text{model}}\) denotes the number of models used in the network.

- **\texttt{no\_diff}:** An \(n_{\text{model}} \times 1\) vector, in which the \(i\)-th element is the number of differential variables in the \(i\)-th model.

- **\texttt{no\_alg}:** An \(n_{\text{model}} \times 1\) vector, in which the \(i\)-th element is the number of algebraic variables in the \(i\)-th model.

- **\texttt{no\_pars}:** An \(n_{\text{model}} \times 1\) vector, in which the \(i\)-th element is the number of parameters in the \(i\)-th model.

- **\texttt{NTST}:** A \(p \times 1\) vector, in which the \(i\)-th element is the number of collocation mesh intervals in the \(i\)-th mesh group.

- **\texttt{N\text{COL}:** Number of collocation nodes (scalar).

- **\texttt{async\_ind}:** An \((n_d + n_a) \times 1\) vector in which the first \(n_d\) elements correspond to differential variables and, the remaining elements, equal to \(n_a\), correspond to algebraic variables. Each element is an integer in \(\{1, \ldots, p\}\) which indicates the designated mesh group for that variable.

- **\texttt{mode}:** A scalar integer value in \(\{1, \ldots, n_{\text{seg}}\}\) in which \(n_{\text{seg}}\) is the number of segments in the multi-segment boundary-value problem.

- **\texttt{seg2mode}:** An \(n_{\text{seg}} \times 1\) vector in which the \(i\)-th element indicates the type of event associated with the \(i\)-th segment.
• **no_of_models**: Number of models used in the network.

• **fun_ind**: An \((n_d + n_a) \times 1\) vector in which the first \(n_d\) elements correspond to differential equations and, the remaining elements, equal to \(n_a\), correspond to algebraic equations. Each element is an integer in \(\{1, \ldots, p\}\) which indicates the designated mesh group for that equation.

• **g_ind**: An \(n_{\text{model}} \times 1\) cell array in which the \(i\)-th element is a vector of indices corresponding to the algebraic equations.

• **par_shape**: An \(n_p \times 1\) cell array in which the \(i\)-th element is the size dimensions of the \(i\)-th parameter. It is used because in the collocation code the parameter values are stored as a concatenated vector, while the models in the forward simulation code require that parameters be stored in specific dimensions.

• **unique_events**: An \(n_{\text{ev}} \times m_{\text{ev}}\) matrix in which \(n_{\text{ev}}\) and \(m_{\text{ev}}\) are the number of unique events and the number of event variables among all models, respectively. Each element in this matrix takes a value equal to either 1 or \(-1\). This value corresponds to the state of the associated event variable.

• **modtyp**: An \(n_{\text{model}} \times 1\) cell array in which the \(i\)-th element is a string reference to the \(i\)-th model.

• **ev_range**: An \(n_{\text{model}} \times 1\) cell structure in which the \(i\)-th element is a vector of integers that indicate the indices of the event variables as stored in unique_events. If an element is empty, it means that no event variable exists in the corresponding model.

• **connec**: A cell array with \(n_c\) rows in which \(n_c\) is the number of connecting equations. The number of columns of this cell array is equal to the highest number of terms in a connecting equation. Each row defines a connecting equation and the absolute value of each element corresponds to the index of the algebraic variable as discussed in the previous section. Since the number of terms is not identical among connecting equations, some elements in this array might be empty.
- **conn_ind**: An \( n_c \times 1 \) vector that contains the indices of the connecting equations.

The `for`-loop in `powersystem.m` goes through all the models in the system. The corresponding indices of the differential variables, algebraic variables, and the model parameters of all models are stored in `diff_ind`, `alg_ind`, and `par_ind`, respectively, through the assignments

```matlab
diff_ind = [0 cumsum(data.no_diff)];
alg_ind = sum(data.no_diff) + [0 cumsum(data.no_alg)];
par_ind = [0 cumsum(data.no_pars)];
```

Within the asynchronous collocation toolbox, both `xx` and `yy` are \((n_a + n_d) \times p\) cell arrays, where the \((i,j)\) component has the basepoint values of the \(i\)-th variables at the \(j\)-th mesh group. The information of the mesh used in the variables of each model is extracted from `data` structure into `x_ind` and `y_ind` associated to the differential and algebraic variables, respectively, through the following assignments.

```matlab
x_ind = ind((diff_ind(i)+1):diff_ind(i+1));
y_ind = data.fun_ind(data.g_ind{i});
```

Here, it is assumed that the equation mesh corresponding to differential equations is assigned to be equal to the variable mesh with first order derivative in that equation. However, the equation mesh corresponding to algebraic equations is not trivial and the user can assign it through `fun_ind` in the `data` structure. By default, if the equation mesh is not provided, the variable mesh is assigned to equations, i.e., the mesh corresponding to the \(i\)-th variable is assigned to the \(i\)-th equation. The desired values of both differential and algebraic variables are then extracted from `xx`, at the proper equation mesh, through the following assignments.

```matlab
x_1 = xx((diff_ind(i)+1):diff_ind(i+1),x_ind);
x_2 = xx((diff_ind(i)+1):diff_ind(i+1),y_ind);
y_1 = xx((alg_ind(i)+1):alg_ind(i+1),x_ind);
y_2 = xx((alg_ind(i)+1):alg_ind(i+1),y_ind);
```

Similarly, the corresponding parameter values are extracted from `pp` as follows.

```matlab
param_x = reshape(pp((par_ind(i)+1):par_ind(i+1)), data.par_shape{i}{1},... data.par_shape{i}{2});
param_y = reshape(pp((par_ind(i)+1):par_ind(i+1)), data.par_shape{i}{1},... data.par_shape{i}{2});
```

Note that parameter values are stored in a large vector, `pp`, in the collocation code, however, they are reshaped according to the specific dimensions for each model in order to be compatible with the models in the forward simulation code.
In the current implementation of this problem, prior knowledge of a fixed event history (signature) is assumed, and the event history in each segment is extracted into `ev` through the assignment

```
    ev = data.unique_events(mode, :);
```

Following the format of the forward simulation code, the DAEs are in the semi-implicit form, and their right-hand side of the differential and algebraic equations are evaluated and stored in `f` and `g`, respectively, through the following assignments.

```
f = feval(data.modtyp{i},x_1,y_1,1,ev(data.ev_range{i}},{param_x},1,i);
g = feval(data.modtyp{i},x_2,y_2,1,ev(data.ev_range{i}},{param_y},4,i);
```

Here, the syntax follows the native format of the forward simulation code. The MATLAB function `feval` is used to evaluate the function whose handle is given in the first argument, i.e., `data.modtyp{i}`. The second and third arguments are the values of the differential and algebraic variables, respectively, at the collocation nodes. Next, a value equal or larger than zero is passed to indicate that the evaluation of the autonomous equations is not for the purpose of initialization (in which a negative value should be used). The fifth input argument then passes the corresponding event values for the current model. After passing the parameter values through the sixth input argument, the next input argument takes a value between 0 to 6 such that

0: Reads some stored values in models for initialization

1: Evaluates the vector field of the differential equations.

2: Evaluates the Jacobian of the vector field with respect to the differential variables.

3: Evaluates the Jacobian of the vector field with respect to the algebraic variables.

4: Evaluates the algebraic equations.

5: Evaluates the Jacobian of the algebraic equations with respect to the differential variables.

6: Evaluates the Jacobian of the algebraic equations with respect to the algebraic variables.
Here, the values are all calculated within a specific model. Finally, the last input argument takes the index of the current model. The residual of the fully implicit form is then obtained through the subtraction from the basepoint values of the derivative stored in $yy$ through the following assignments.

```matlab
y1_ind = sub2ind(size(yy),(diff_inds(i)+1):diff_inds(i+1),x_ind);
y1(range1+1:range1+length(f)) = cell2mat(yy(y1_ind')) - f;
y2(range2+1:range2+length(g)) = g;
```

These values are evaluated for all models and stored, vertically concatenated, in vectors $y_1$ and $y_2$ corresponding to the differential and algebraic equations, respectively.

Similar to the evaluation of the right-hand sides of the differential and algebraic equations, each model is also called with flag equal to 0 for reading values related to the indices in the model through the following assignment.

```matlab
o = feval(data.modtyp{i},x_2,y_2,1,ev(data.ev_range{i}),(param),0,i);
```

As an example, the output $o$ can take the following value

```matlab
o = [5 16 [12 13 16 17]]
```

The first, second, and the remaining elements constitute three parts of this output, respectively. In this example, the output indicates that there are 5 differential equations and 16 algebraic equations in the corresponding model. Moreover, the third part, i.e. $[12 13 16 17]$ indicates that there exists four event variables in this model that correspond to the 12-th, 13-th, 16-th, and 17-th algebraic variables in the model. This order of event variables is fixed throughout the model, hence, any association to the third event variable, as an example, corresponds to the 16-th algebraic variable in this model. This order of events is then stored through the following assignment.

```matlab
data.event_order{i} = data.alg_inds(i) + o(3:length(o));
```

Finally, the algebraic relations that connect algebraic variables among models are imposed through extraction from field `connec` in data structure. The corresponding residuals are appended to the residuals of the algebraic equations and the final output of `powersystem.m` is formed as

```matlab
y = [y1;y2];
```
The hybrid system toolbox along with the asynchronous collocation toolbox adds the zero problem of the power system vector field through the assignment

```matlab
opts = adaphs_isol2sol(opts, fid, @powersystem, segs, pars, @powersystem_bc, ...
                      'dbcdx', @powersystem_bc_DFDX);
```

Here, `adaphs_isol2sol` calls the hybrid system toolbox where the structure `segs` contains the initial solution guess of all segments passed to the toolbox together with the parameter values, `pars`, and the handles of the boundary conditions function and its Jacobian. The outcome of the toolbox is then stored in `opts` which is a structure that contains the constructed numerical continuation problem.

The preceding collocation conditions have a dimensional deficit of \( n_{seg} (n_a + n_d + 1) \) after the imposition of continuity condition of mesh intervals within each segment (see Appendix B and [5]), in which \( n_{seg} \) denotes the total number of segments. In the original implementation of this system, there exist reset conditions which trigger events at the end of each segment. Since the occurrence of events is assumed to be fixed both in type and order, these conditions are used along with the jump functions defined for each event to obtain \( n_{seg} (n_a + n_d + 1) - (n_a + n_d) \) additional conditions. The remaining \( (n_a + n_d) \) conditions are the initial conditions for the first segment. These conditions are imposed as the boundary conditions in `powersystem_bc.m` as demonstrated below.

```matlab
1 function fbc = powersystem_bc(x0,x1,pp,T,mode,data)
2  % (c) Mehdi Saghafi, 2014
3
4 bd_p = pp(sum(data.no_pars)+1:sum(data.no_pars)+sum(data.no_diff));
5 gp_pidx = sum(data.no_pars)+sum(data.no_diff)+1;
6
7 numdiff = sum(data.no_diff);
8 range2 = 0;
9
diff_inds = [0 cumsum(data.no_diff)];
10 alg_inds = sum(data.no_diff) + [0 cumsum(data.no_alg)];
11 par inds = [0 cumsum(data.no_pars)];
12
13 switch mode
14   case num2cell(1:length(data.seg2mode)-1)
15     fbc1 = x0(1:numdiff) - x1(1:numdiff);
16     fbc2 = zeros(sum(data.no_alg),1);
17     for i = 1:data.no_of_models
18       x_ind = (diff_inds(i)+1):diff inds(i+1);
19       y_ind = (alg inds(i)+1):alg inds(i+1);
20       param = reshape(pp([par inds(i)+1:par inds(i+1)], data.par_shape{i}));
21       ev = data.unique_events(data.seg2mode(mode+1), :);
22      end
23
24   case length(data.seg2mode)
25     fbc1 = zeros(sum(data.no_alg),1);
26     fbc2 = zeros(sum(data.no_alg),1);
27     for i = 1:data.no_of_models
28       x_ind = (diff inds(i)+1):diff inds(i+1);
29     y_ind = (alg inds(i)+1):alg inds(i+1);
30     param = reshape(pp([par inds(i)+1:par inds(i+1)], data.par_shape{i}));
31     ev = data.unique_events(data.seg2mode(mode+1), :);
32    end
33
34 case > length(data.seg2mode)
35    for i = 1:length(data.seg2mode) - 1
36      fbc1 = fbc1;
37      fbc2 = fbc2;
38      for j = 1:length(data.seg2mode(i+1))
39        x_ind = (diff_inds(i)+1):diff inds(i+1);
40        y_ind = (alg inds(i)+1):alg inds(i+1);
41        param = reshape(pp([par inds(i)+1:par inds(i+1)], data.par_shape{i}));
42        ev = data.unique_events(data.seg2mode(i+1), :);
43      end
44    end
45
46 case 'dbcdx'
47     for i = 1:length(data.seg2mode)
48       fbc1 = fbc1;
49       fbc2 = fbc2;
50      for j = 1:length(data.seg2mode(i+1))
51         x_ind = (diff_inds(i)+1):diff inds(i+1);
52         y_ind = (alg inds(i)+1):alg inds(i+1);
53         param = reshape(pp([par inds(i)+1:par inds(i+1)], data.par_shape{i}));
54         ev = data.unique_events(data.seg2mode(i+1), :);
55      end
56    end
57
58 case 'all'
59    for i = 1:length(data.seg2mode)
60      fbc1 = fbc1;
61      fbc2 = fbc2;
62      for j = 1:length(data.seg2mode(i+1))
63        x_ind = (diff_inds(i)+1):diff inds(i+1);
64        y_ind = (alg inds(i)+1):alg inds(i+1);
65        param = reshape(pp([par inds(i)+1:par inds(i+1)], data.par_shape{i}));
66        ev = data.unique_events(data.seg2mode(i+1), :);
67      end
68    end
69
70   end
71
72 % (c) Mehdi Saghafi, 2014
73 save
```

27
Here, the input arguments for the \( i \)-th segment are as follows. \( x_0 \) and \( x_1 \) are \((n_d + n_a) \times 1\) vectors of the first and last basepoints in the \((i + 1)\)-th and \(i\)-th segments, respectively. Note that \( x_0 \) in the last segment denote the first basepoints in the first segment. The third input argument, \( pp \), denotes the vector of parameter values. Finally, \( mode \) and \( data \) are the index of the segment and the data structure, respectively.

As was mentioned earlier in this chapter, the differential variables are always considered to be continuous in this implementation. This condition is imposed through the following assignment.

\[
fbc1 = x_0(1:numdiff) - x_1(1:numdiff);
\]

The algebraic variables, however, are not necessarily continuous and we impose the condition that the algebraic equations be satisfied at the first basepoints of all segments except the first one. This condition is imposed through the following assignment.

\[
g = \text{feval(data.modtyp{\( i \)}, x_0x, x_0y, 1, ev(data.ev.range{\( i \)}), params, 4, i);
\]

The loop goes through all the models and store the residuals in \( fbc2 \). Similarly, this condition is imposed for the connecting equations and the residuals are appended to \( fbc2 \). The end condition of all segments, except the last one, is then imposed by their corresponding event conditions as follows.

\[
fbc3 = x_1(data.event.order{\( data.Model(mode) \}){\( data.Order(mode) \)});
\]
The initial conditions of the first segment as well as the end condition of the last segment were not covered in the previous imposition of boundary conditions. These conditions are implemented as

$$fbc = [ps\_init(x0,pp)\ ;\ graze1(x1,pp,gr\_pidx)];$$

Here, \texttt{ps\_init} calls a function which imposes the initial conditions at the first basepoints of the first segment and returns a \((n_d + n_a) \times 1\) vector of residuals. The end condition of the last segment is then imposed by \texttt{graze1} which returns a scalar residual. The code for imposing the initial boundary condition is given below.

```matlab
1 function y = ps\_init(x0,pp)
2 % (c) Mehdi Saghafi, 2014
3 bd\_p = pp(sum(data.no\_pars)+1:sum(data.no\_pars)+sum(data.no\_diff));
4 range1 = 0;
5 range2 = 0;
6
diff\_inds = [0 cumsum(data.no\_diff)];
7 alg\_inds = sum(data.no\_diff) + [0 cumsum(data.no\_alg)];
8 par\_inds = [0 cumsum(data.no\_pars)];
9
y2 = zeros(sum(data.no\_alg),1);
10 for i = 1:data.no\_of\_models
11 x\_ind = (diff\_inds(i)+1):diff\_inds(i+1);
12 y\_ind = (alg\_inds(i)+1):alg\_inds(i+1);
13 param = reshape(pp((par\_inds(i)+1):par\_inds(i+1)), data.par\_shape{1});
14 ev = data.unique\_events(data.seg2mode{1}, :);
15 x0x\_1 = repmat(num2cell(x0(x\_ind)), 1, length(data.f\_ind{1}));
16 x0y\_1 = repmat(num2cell(x0(y\_ind)), 1, length(data.f\_ind{1}));
17 x0x\_2 = repmat(num2cell(x0(x\_ind)), 1, length(data.g\_ind{1}));
18 x0y\_2 = repmat(num2cell(x0(y\_ind)), 1, length(data.g\_ind{1}));
19 params = {param,bd\_p(x\_ind)};
20 f = feval(data.modtyp{i},x0x\_1,x0y\_1,-1,ev(data.ev\_range{1}),params,1,i);
21 g = feval(data.modtyp{i},x0x\_2,x0y\_2,-1,ev(data.ev\_range{1}),params,4,i);
22 y1(range1+1:range1+length(f),1) = f;
23 y2(range2+1:range2+length(g),1) = g;
24
25 range1 = range1 + length(f);
26 range2 = range2 + length(g);
27
28 for k1 = 1:size(data.connec,1)
29 tmp = x0(abs([data.connec{k1,:}]));
30 sign1 = sign(cell2mat(data.connec{k1,:}));
31 sign1 = repmat(sign1, size(tmp,1), 1);
32 tmp = tmp.* sign1;
33 y2(range2+k1,1) = sum(tmp,2);
34
35 y = [y1;y2];
36
37 end
38
39 end
```

29
This code is similar to the previous ones in which the collocation conditions were imposed. The main difference here is that the fourth input argument of the power system models is a negative number in order to call the governing equations associated with the initialization of the system. Finally, the end condition of the last segment, here grazel, can be chosen to be a particular grazing condition depending on the specific boundary-value formulation of interest.

The Jacobian of the boundary-value formulation is required in the continuation scheme we use for the analysis of power systems. Although it is possible to evaluate such Jacobian using a finite-difference scheme, we provide an analytic form of the Jacobian to improve the efficiency of the computational analysis. The corresponding Jacobian of powersystem.m with respect to basepoint values and parameters are evaluated separately in powersystem_DFDX.m and powersystem_DFDP.m, respectively, as shown below.

```matlab
function [data, dfode1, dfode2] = powersystem_DFDX(xx, yy, pp, inds, data)
% (c) Mehdi Saghafi, 2014

mode = data.seg2mode(data.mode);

diff_inds = [0 cumsum(data.no_diff)];
alg_inds = sum(data.no_diff) + [0 cumsum(data.no_alg)];
par_inds = [0 cumsum(data.no_pars)];

Fx = cell(data.no_of_models,1);
Fy = cell(data.no_of_models,1);
Gx = cell(data.no_of_models,1);
Gy = cell(data.no_of_models,1);

for i = 1:data.no_of_models
    x_ind = inds((diff_inds(i)+1):diff_inds(i+1));
    y_ind = data.fun_ind(data.g_ind{i});
    x_1 = xx((diff_inds(i)+1):diff_inds(i+1),x_ind);
    x_2 = xx((diff_inds(i)+1):diff_inds(i+1),y_ind);
    y_1 = xx((alg_inds(i)+1):alg_inds(i+1),x_ind);
    y_2 = xx((alg_inds(i)+1):alg_inds(i+1),y_ind);
    param_x = reshape(pp((par_inds(i)+1):par_inds(i+1)), ...
                      data.par_shape{i}(1),data.par_shape{i}(2));
    param_y = reshape(pp((par_inds(i)+1):par_inds(i+1)), ...
                      data.par_shape{i}(1),data.par_shape{i}(2));
    ev = data.unique_events(mode, :);
    fx = feval(data.modtyp{i},x_1,y_1,1,ev(data.ev_range{i}),{param_x},2,i);
    fy = feval(data.modtyp{i},x_1,y_1,1,ev(data.ev_range{i}),{param_x},3,i);
    gx = feval(data.modtyp{i},x_2,y_2,1,ev(data.ev_range{i}),{param_y},5,i);
    gy = feval(data.modtyp{i},x_2,y_2,1,ev(data.ev_range{i}),{param_y},6,i);

    Fx{i} = -fx;
    Fy{i} = -fy;
    Gx{i} = gx;
    Gy{i} = gy;
```

30
end

Fx = blkdiag(Fx{:});
Fy = blkdiag(Fy{:});
Gx = blkdiag(Gx{:});
Gy = blkdiag(Gy{:});
dftmp = [Fx,Fy;Gx,Gy];
[rows,cols,vals] = find(dftmp);
rtmp = max(rows);
for k1 = 1:size(data.connec,1)
    tmp = [data.connec{k1,:}];
    reps = data.NTST(data.conn.ind(data.conn.ind{k1}))*data.NCOL;
    rows = [rows;repmat((rtmp+1:reps)’, [length(tmp) 1])];
    cols = [cols;repmat(repmat(repmat([1 length(tmp)]),abs(tmp)),1 length(tmp))];
    vals = [vals;repmat(repmat([1 length(tmp)]),sign(tmp))];
end
dfode1 = sparse(rows,cols,vals);
dfptmp = arrayfun(@(x) sparse(ones(x,1)), ...
    data.NTST(data.async.ind(1:sum(data.no.diff)))*data.NCOL, 'UniformOutput', false);
dfptmp = blkdiag(dfptmp{:});
[m,n,v] = find(dfptmp);
dfode2 = sparse(m,n,v,size(dfode1,1),size(dfode1,2));

function [data, dfode] = powersystemDFDP(xx, yy, pp, inds, data)
% (c) Mehdi Saghafi, 2014
bdp = pp(sum(data.no_pars)+1:sum(data.no_pars)+sum(data.no_diff));
mode = data.seg2mode(data.mode);
diff_inds = [0 cumsum(data.no_diff)];
alg_inds = sum(data.no_diff) + [0 cumsum(data.no_alg)];
par_inds = [0 cumsum(data.no_pars)];
Fp1 = cell(data.no_of_models,1);
Fp2 = cell(data.no_of_models,1);
Gp1 = cell(data.no_of_models,1);
Gp2 = cell(data.no_of_models,1);
for i = 1:data.no_of_models
    x_ind = inds((diff inds(i)+1):diff inds(i+1));
y_ind = data.fun.ind(data.g.ind{i});
    x_l = xx((diff inds(i)+1):diff inds(i+1),x_ind);
    x_r = xx((diff inds(i)+1):diff inds(i+1),y_ind);
    y_l = xx((alg inds(i)+1):alg inds(i+1),x_ind);
    y_r = xx((alg inds(i)+1):alg inds(i+1),y_ind);
    param = reshape(pp((par inds(i)+1):par inds(i+1)), ...
        data.par.shape{i}(1),data.par.shape{i}(1));
    ev = data.unique_events(mode, :);
    params = {param,bdp(x_ind)};
    fp = feval(data.modtyp{i},x_l,y_l,1,ev(data.ev_range{i}),params,10,i);
    gp = feval(data.modtyp{i},x_r,y_r,1,ev(data.ev_range{i}),params,11,i);
    Fp1{i} = -fp{1};
    Fp2{i} = -fp{2};
    Gp1{i} = gp{1};
In powersystem_DFDX.m, Fx and Gx are the Jacobians of the right-hand side of the differential and algebraic equations, respectively, with respect to basepoint values corresponding to differential variables. Similarly, Fy and Gy are the corresponding Jacobians with respect to algebraic variables. Since, in the collocation code, the first-order differential equations are rewritten such that the differentiated variables are subtracted by the vector fields, Jacobians evaluated in the forward simulation code are multiplied by a negative sign in lines 36 and 37. The Jacobian of the connecting equations is then evaluated in the second for-loop with values 1 and -1 in proper locations, as per the format of these equations. Here, repel is a MATLAB function, whose encoding is given in the appendix. In the call repel(reps,vals), reps and vals are vectors with identical sizes. The i-th element of vals is repeated r_i times to form a subvector, where r_i is the i-th element of reps. These subvectors are then concatenated, in their original order, to form the output vector of repel. This function is demonstrated in Appendix C. Finally, dfptmp is the coefficient matrix in front of the time derivatives of the state variables in the governing equations.

In powersystem_DFDP.m, the Jacobian of the vector field and the algebraic equations are evaluated using flags 10 and 11, respectively. Here, the parameter values are divided into two parts and stored in the cell elements of p. The first element contains the original parameters of the forward simulation code, while the second element contains the parameters introduced in the modified version of models used in the collocation code.

The phenomenon of grazing corresponds to tangential intersection of a state-space trajectory with a zero level surface of an event function h : \( \mathbb{R}^{n_d+n_a} \to \mathbb{R} \). In addition to the condition that trajectories end at the event surface, the tangentiality condition at grazing
point is given by

\[ h_x(x, y) \cdot \dot{x} + h_y(x, y) \cdot \dot{y} = 0 \]  \hspace{1cm} (2.9)

Here, \( \dot{x} \) is known from the DAEs while \( \dot{y} \) is obtained through differentiation of the algebraic equations. Note that the latter is possible since the governing DAEs of power systems are index-1.

In order to cover the subset of the event surface where trajectories are tangential, the boundary-value formulation should satisfy both grazing conditions. While one of these conditions is already imposed in the code for boundary conditions, i.e. `powersystem_bc.m`, we add a constraint to the problem in which the second grazing condition is imposed. Provided that the initial guess satisfies the grazing condition, numerical continuation can be performed at these codimension-2 families in order to cover the grazing surface, i.e., the manifold on \( h \) where tangentiality occurs.

### 2.4 A test problem

In order to show the feasibility of the continuation paradigm using the wrapper code discussed in this chapter as well as its implementation in COCO, we bring here the results for a parametric analysis performed on the IEEE 14-bus system. The details of this system is described in Ref. [10]. The aim of the analysis is to explore the relationship between values of real power demand at different buses that cause the apparent impedance trajectory to graze the protection hypersurface. More details on this analysis are published in a conference paper (Ref. [8]).

In this problem, a total of 46 models were used to construct the network. Also, the number of differential and algebraic variables were 177 and 391, respectively. The time period of the initial solution is 0.4081 seconds which is divided into 20 segments for events occurring at their end points. This event history, as was mentioned earlier, is kept fixed throughout the continuation. During the short time span of each segment, except the last one, all variables exhibit slow behavior. Hence, we use only one time interval in the collocation discretization approximated by third degree polynomials, except for the last segment in
which four intervals are used. Consequently, although the toolbox provides such capability, the asynchronous discretization was not employed in this problem.

We start from a solution that satisfies the tangentiality condition and continue the solution by varying the real power demand at one of the buses. Continuation is performed until a solution that satisfies the performance constraint is detected. To continue the corresponding family of grazing trajectories, we fix the bus load and impose the zero-crossing of the performance constraint. We then obtain a one-dimensional solution manifold by allowing the real power demand at two of the buses to vary.

2.5 Summary

This study describes the outcome of coupling the continuation framework to an existing library of power system components. This approach supports trajectory discretization methods based on asynchronous collocation, discussed in Chapter 3 and Appendix B, that enable scaling up to the number of states typical of practical power systems. The coupled framework allows user-driven exploration of switching, threshold and instability-related grazing phenomena, and hence provides tools for parametric uncertainty evaluation and vulnerability assessment of networks.
3.1 Introduction

The presence of clouds in the atmosphere can drastically influence the albedo by hindering the normal radiative heat exchanges between the earth and the atmosphere. In fact, the influence of human-made aerosol on the optical properties of clouds is a major source of uncertainty in climate simulations. For this reason, the study of the formation of clouds is of great importance, and, in particular, the role of aerosol particles as cloud condensation nuclei (CNN) has been the subject of extensive studies [16, 17, 18, 19, 20]. These particles originate from different sources [21] including sea salt, sulfate particles, etc., and can expedite the initiation of the condensation of water vapor down to a relatively low humidity. In general, all mechanisms through which the climate is affected by the influence of aerosols on droplet number concentration are referred to as the aerosol indirect effect on climate [22, 23, 24, 25].

For a fixed adiabatic control volume, the supply of water vapor is limited and aerosol particles compete for its absorption. If the relative humidity exceeds the so-called supersaturation points of an individual particle, then the particle starts growing. It will continue to do so to form a droplet unless a decrease in the relative humidity causes the absorbed water to evaporate.

The fraction of particles that form a cloud droplet is of great practical importance and is often studied based on the physical properties of the particles and their environment. Such particles are called activated. In order to avoid long forward simulations for very
large-scale systems of particles, a parametrization is commonly sought to find an estimate of the maximum supersaturation and the critical supersaturation of all particles based on the main contributing parameters of the system [26, 27, 28, 29, 30, 31], for a review see [32]. The fraction of activated particles is then estimated through some simple threshold criteria on the size of particles and the associated critical values. These simplifying assumptions are sometimes acceptable, however, there are instances where they overestimate the number of activated particles possibly to a significant extent. This overestimation is caused by a phenomenon called kinetic limitation [33, 34]. An example of this phenomenon occurs when a particle does not grow fast enough to form a cloud droplet. In this case, the growth of the aerosol particle is interrupted and the particle begins to shrink in size.

The overestimation of activated particles is a result of ignoring the kinetic limitation. This shortfall of common parameterization methods requires a more careful analysis of the influences of the kinetic limitation. In the literature, however, most of the aerosol activation schemes neglect this effect [32]. Only few studies (see, e.g.,Refs. [35, 36, 37]) try to address a class of kinetically limited systems to avoid overestimation of the cloud droplets due to a certain kinetic mechanism known as inertial mechanism (see section 3.2.4).

It is the purpose of this work to propose a relatively low-complexity computational framework for the investigation of the influence of the kinetic limitation phenomena. Towards this end, we propose a boundary-value formulation for a large set of particles in which the particle diameters grow in time. The solutions to this formulation has a fixed fraction of particles that form droplets. In order to use numerical continuation to study the parameter dependence of this formulation, proper discretization of variables in the time domain is required. This poses another challenge to the feasibility of this formulation as the large problem size makes the number of mesh points excessively large. This complexity issue is even further exacerbated considering several regions in the time domain where at least one of the particles undergoes sudden changes in its growth. An asynchronous collocation algorithm is, therefore, devised in which mesh points in time domain do not necessarily coincide among different variables.

Asynchronous discretization schemes has been used in finite-element problems (where they are called subcycling) to have different time steps for different spatial domains to overcome
the efficiency issue in large scale problems with stiff elements [38, 39, 40, 41]. Even earlier, numerical integration methods were proposed that used different time-steps for different variables [42, 43, 44] (see Refs. [45, 46, 47] for some recent works). These methods are based on the principle that a system can be partitioned such that different schemes are applied to different partitions. This may ensure an efficient and stable solution strategy [48].

In section 3.2, we first describe the coupled differential equations that govern the growth dynamics. We proceed by showing several characteristics of both autonomous and non-autonomous models for the growth of the particle diameters. The effect of kinetic limitation is then discussed following the introduction of an ad-hoc parametrization for estimating the fraction of activated particles. We proceed, in section 3.3, to perform an extensive numerical study of the growth dynamics.

3.2 Model description

The equations governing time-dependent variations in the size of cloud-condensation nuclei may be derived by considering a control volume containing a fixed number of aerosol particles with identical compositions [49, Chapter 17]. Such a model assumes that the effects of emissions, dilution, and coagulation between particles are negligible and that the condensation/evaporation of water onto/from individual particles is coupled to the availability of water vapor represented by the relative humidity. We will consider adiabatic evolution of the control volume with a fixed total water content. In Sections 3.2.1 and 3.2.2 we will assume constant ambient temperature and pressure for the control volume, while in Sections 3.2.3, 3.2.4, and 3.3 we will model cloud-formation via a constant-speed updraft, following [50] and [51]. This corresponds to a linear temperature profile and constant pressure [49, Section 17.3.2].

Let $D$ denote the wet diameter of an aerosol particle and denote the environmental saturation ratio (i.e., the relative humidity) and the ambient temperature by $S$ and $T$, respectively. Following [50], for fixed $T$, variations in $D$ beyond the (initial) dry diameter $D_{\text{dry}}$ are then
governed by a differential equation of the form

\[ \dot{D} = r(D) (S - q(D; D_{\text{dry}})) , \]  

(3.1)

where

\[ r : D \mapsto \frac{\alpha_0}{D + \alpha_1} \]  

(3.2)

and

\[ q : (D; D_{\text{dry}}) \mapsto \frac{D^3 - D_{\text{dry}}^3}{D^3 + (\kappa - 1) D_{\text{dry}}^3} e^{\beta/D} . \]  

(3.3)

for some positive constants \( \alpha_0, \alpha_1, \beta, \) and the aerosol hygroscopicity (i.e., capacity to absorb water vapor) \( \kappa. \) Physical expressions for \( \alpha_0, \alpha_1, \) and \( \beta \) are given below in equations (3.26)–(3.28), while \( \kappa \) is a parameter that depends on the aerosol constituents. From Petters and Kreidenweis [17], we find that typical aerosol species correspond to nonnegative values of \( \kappa \) limited from above by around 1.4. In the analysis below, we assume that this bound holds throughout.

### 3.2.1 General properties

The following initial characterizations of the function \( q(D; D_{\text{dry}}) \) provide a foundation for the analysis of the aerosol size dynamics.

**Lemma 1.** The function \( q(D; D_{\text{dry}}) \) has a unique local maximum for some critical diameter greater than \( D_{\text{dry}}. \)

**Proof.** The function \( q(D; D_{\text{dry}}) \) equals 0 for \( D = D_{\text{dry}} \) and converges to 1 from above as \( D \to \infty. \) Indeed, the sign of the partial derivative \( \partial_D q(D; D_{\text{dry}}) \) equals that of the polynomial

\[ p(D; D_{\text{dry}}) := 3D^4D_{\text{dry}}^3 \kappa - (D^3 - D_{\text{dry}}^3) (D^3 - D_{\text{dry}}^3 + D_{\text{dry}}^3 \kappa) \beta , \]  

(3.4)

which is positive when \( D = D_{\text{dry}} \) and negative for sufficiently large \( D. \) It follows that there exists at least one local maximum for some critical diameter greater than \( D_{\text{dry}}. \) The total number of local extrema of \( q(D; D_{\text{dry}}) \) for \( D > D_{\text{dry}} \) equals the number of roots of
the polynomial $p(D_{\text{dry}} + \delta; D_{\text{dry}})$ with $\delta > 0$. By Descartes’ rule, an upper bound for this number is given by the number of sign changes in the sequence

$$1, 4 - \tilde{\beta}, 6\kappa - \tilde{\beta}(\kappa + 3), 12\kappa - \tilde{\beta}(18 + \kappa), \kappa - 5\tilde{\beta}, -1$$

(3.5)

for $\tilde{\beta} > 0$. For $\kappa < 1.4$, this number equals 1 and the claim follows.

An alternative proof of this result is given after equation (8) in [52], but the proof above is a precursor for the results below.

**Corollary 1.** *The maximum of $q(D; D_{\text{dry}})$ is greater than 1.*

We denote the value of $D$ at the local maximum of $q(D; D_{\text{dry}})$ by $D_{\text{crit}}$ and refer to the value $q(D_{\text{crit}}; D_{\text{dry}})$ as the *critical saturation ratio* associated with the dry diameter $D_{\text{dry}}$. Since this value is greater than 1, the excess $q(D_{\text{crit}}; D_{\text{dry}}) - 1$ is called the *critical supersaturation* and the diameter $D_{\text{crit}}$ is referred to as the corresponding *supersaturation point*. The graph of $q(D; D_{\text{dry}})$ as a function of $D$ is known as a *Köhler curve*. Several such curves are shown in Fig. 3.1.

![Köhler curves](image)

**Figure 3.1:** Köhler curves for different values of $D_{\text{dry}}$. The red curve shows the loci of critical saturation points.
Lemma 2. No two Köhler curves intersect.

Proof. The claim follows from the observation that

\[ \partial_{D_{\text{dry}}} q(D; D_{\text{dry}}) = -\frac{3D^3D_{\text{dry}}^2\kappa}{(D^3 + D_{\text{dry}}^3(\kappa - 1))^2} e^{3/D} \]  

(3.6)

is everywhere negative.

Corollary 2. The critical supersaturation is a decreasing function of \( D_{\text{dry}} \).

Further information about the shape of individual Köhler curves may be gleaned from the following lemma.

Lemma 3. Suppose that \( q(D_1; D_{\text{dry}}) = q(D_2; D_{\text{dry}}) \) for \( D_1 \) and \( D_2 \) to the left and right, respectively, of \( D_{\text{crit}} \). Then the slope of the graph of \( q(D; D_{\text{dry}}) \) at \( D = D_1 \) is greater than the negative of the slope at \( D = D_2 \).

Proof. For \( \kappa < 1.4 \), an analysis based on Descartes’ rule shows that there is a unique inflection point along every Köhler curve and that this occurs to the right of the critical diameter. The claim follows from the additional observation that the curvature is negative for \( D = D_{\text{dry}} \), converges to 0 from above as \( D \to \infty \), and (by Descartes) has a unique local extremum for \( D > D_{\text{dry}} \).

Finally, information about variations in the shape across the family of Köhler curves follows from the following lemma.

Lemma 4. The slope of the graph of \( q(D; D_{\text{dry}}) \) for a fixed value of \( q(D; D_{\text{dry}}) \) is a decreasing function of \( D_{\text{dry}} \) to the left of \( D_{\text{crit}} \) and an increasing function of \( D_{\text{dry}} \) to the right of \( D_{\text{crit}} \).

Proof. The substitution \( D = D_{\text{dry}} \) in \( q(D; D_{\text{dry}}) \) yields a constant function of \( D_{\text{dry}} \) provided that

\[ \frac{d}{dD_{\text{dry}}} D(D_{\text{dry}}) = -\frac{\partial_{D_{\text{dry}}} q(D(D_{\text{dry}}); D_{\text{dry}})}{\partial_D q(D(D_{\text{dry}}); D_{\text{dry}})}. \]  

(3.7)

Now consider the rate of change of \( \partial_D q(D(D_{\text{dry}}); D_{\text{dry}}) \) with respect to \( D_{\text{dry}} \), obtained by the use of the chain rule and substitution of Eq. (3.7), and suppose that \( \kappa < 1.4 \). Using
Descartes’ rule, we conclude that the sign of this quantity is opposite to that of the slope of $q (D; D_{\text{dry}})$ evaluated at $D = D_{\text{eq}}$, and the claim follows.

**Corollary 3.** The magnitude of the slope of $q (D; D_{\text{dry}})$ for a fixed value of $q (D; D_{\text{dry}})$ is a decreasing function of $D_{\text{dry}}$ for all $D > D_{\text{dry}}$ away from $D_{\text{crit}}$.

**Corollary 4.** Suppose that $D_{1,\text{dry}} < D_{2,\text{dry}}$ and consider $D_1 < D_{1,\text{crit}}$ and $D_2 > D_{2,\text{crit}}$, such that

$$q (D_1; D_{1,\text{dry}}) = q (D_2; D_{2,\text{dry}}).$$

(3.8)

It follows that

$$r (D_1) \partial_D q (D_1; D_{1,\text{dry}}) > - r (D_2) \partial_D q (D_2; D_{2,\text{dry}}).$$

(3.9)

### 3.2.2 Autonomous dynamics

For fixed $S$, it is clear from Eq. (3.1) that equilibrium is obtained for $D = D_{\text{eq}}$, where the *equilibrium diameter* $D_{\text{eq}}$ must satisfy

$$S = q (D_{\text{eq}}; D_{\text{dry}}).$$

(3.10)

Since $\partial_D q (D; D_{\text{dry}})$ is positive for $D < D_{\text{crit}}$ and negative for $D > D_{\text{crit}}$, the equilibrium diameters $D_{\text{eq}} < D_{\text{crit}}$ are asymptotically stable, whereas those with $D_{\text{eq}} > D_{\text{crit}}$ are unstable. For $D_{\text{eq}} = D_{\text{crit}}$, we obtain a saddle-node equilibrium. The direction of the flow in $D$ (for different fixed values of $S$) is indicated in Fig. 3.2. In particular, we see unbounded growth in $D$ for $D$ greater than the unstable equilibrium diameter when $1 < S \leq q (D_{\text{crit}}; D_{\text{dry}})$ and for all $D$ when $S > q (D_{\text{crit}}; D_{\text{dry}})$. In contrast, for $0 < S \leq 1$, the unique equilibrium diameter is globally attracting.

From a mass balance relation [49, Section 10.2], it follows that for fixed $T$, changes to the wet diameter of aerosol particles inside the adiabatic volume correspond to variations in the relative humidity that satisfy the differential equation

$$\dot{S} = - \frac{3 \gamma}{N_p} \sum_{i=1}^{N_p} D_i^2 \dot{D}_i$$

(3.11)
Figure 3.2: For fixed values of $S$ corresponding to horizontal slices through this diagram, the arrows indicate the direction of flow of $D$. Green circles indicate stable equilibria and the red circle indicates the unstable equilibrium. Beyond a particular value of $S$ unbounded growth occurs. The blue curve (Köhler curve) is the isocline for which $\dot{D} = 0$. The dotted lines show two cases at which $S = 1$ (bottom) and $S = q(D_{\text{crit}}; D_{\text{dry}})$.

for some positive constant $\gamma$, where we sum over a total of $N_p$ particles competing for the available supply of water vapor. A physical expression for $\gamma$ is given below in equation (3.25).

The negative feedback on the value of $S$ associated with growth of the wet diameter of any of the aerosol particles makes unbounded growth impossible and thus sets the stage for a possible cessation or reversal of growth.

From Eq. (3.11), it follows that

$$S - S_{\text{dry}} = -\frac{\gamma}{N_p} \sum_{i=1}^{N_p} (D_i^3 - D_{i,\text{dry}}^3),$$

where $S_{\text{dry}}$ is the relative humidity prior to any condensation onto the aerosol particles. It follows that the variations in wet diameter for $N_p$ cloud condensation nuclei are governed by the closed system of differential equations

$$\dot{D}_i = r(D_i) \left( S_{\text{dry}} - \frac{\gamma}{N_p} \sum_{j=1}^{N_p} (D_j^3 - D_{j,\text{dry}}^3) - q(D_i; D_{i,\text{dry}}) \right),$$

(3.13)
for $i = 1, \ldots, N_p$.

Consider, at first, the case of $N_p = 1$ and the single differential equation

$$
\dot{D} = r(D) \left( S_{\text{dry}} - \gamma (D^3 - D_{\text{dry}}^3) - q(D; D_{\text{dry}}) \right),
$$

(3.14)

where we have omitted the trivial subscript. Equilibrium is obtained for $D = D_{\text{eq}}$ provided that

$$
S_{\text{dry}} = \gamma (D_{\text{eq}}^3 - D_{\text{dry}}^3) + q(D_{\text{eq}}; D_{\text{dry}}).
$$

(3.15)

**Lemma 5.** For a given initial dry diameter $D_{\text{dry}}$, there exists a critical value $\gamma^*$, such that all equilibria of the dynamical system in Eq. (3.14) are globally attractive provided that $\gamma > \gamma^*$ (see Fig. 3.3). For $\gamma < \gamma^*$, there exists two critical values of $S_{\text{dry}}$, denoted by $S_{\text{dry}}^*$ and $S_{\text{dry}}^{**}$, where $1 < S_{\text{dry}}^* < S_{\text{dry}}^{**}$, such that

- for $0 < S_{\text{dry}} < S_{\text{dry}}^*$, there exists a unique globally attractive equilibrium with $D_{\text{eq}} < D_{\text{crit}}$;

- for $S_{\text{dry}}^* < S_{\text{dry}} < S_{\text{dry}}^{**}$, there exist two locally attractive equilibria with $D_{\text{eq}} < D_{\text{crit}}$ and $D_{\text{eq}} > D_{\text{crit}}$, respectively, separated by an unstable equilibrium with $D_{\text{eq}} > D_{\text{crit}}$.

- for $S_{\text{dry}} > S_{\text{dry}}^{**}$, there exists a unique globally attractive equilibrium with $D_{\text{eq}} > D_{\text{crit}}$.

**Proof.** An equilibrium $D = D_{\text{eq}}$ of the dynamical system in Eq. (3.14) is asymptotically stable when

$$
\gamma > -\frac{\partial dq(D_{\text{eq}}; D_{\text{dry}})}{3D_{\text{eq}}^2}
$$

(3.16)

and unstable when the opposite inequality holds. The expression on the right-hand side is negative for $D_{\text{eq}} < D_{\text{crit}}$, positive for $D_{\text{eq}} > D_{\text{crit}}$, and converges to 0 as $D_{\text{eq}} \to \infty$. It follows that it must attain a local maximum for some $D_{\text{eq}} > D_{\text{dry}}$. As in the proof to Lemma 1, by Descartes’ rule, the total number of local extrema for $D_{\text{eq}} > D_{\text{dry}}$ is bounded from above by
the number of sign changes in the sequence

\[ 1, 48 + (4\kappa + 6)\bar{\beta} - \kappa\bar{\beta}^2, 168 + (4\kappa + 6)\bar{\beta} - (\kappa + 6)\bar{\beta}^2, \]
\[ 1008\kappa + (8\kappa^2 - 30\kappa - 108)\bar{\beta} - (\kappa^2 + 36\kappa + 27)\bar{\beta}^2, \]
\[ 1260\kappa + (2\kappa^2 - 60\kappa - 432)\bar{\beta} - (30\kappa + 81)\bar{\beta}^2, \]
\[ 168\kappa - (7\kappa + 126)\bar{\beta} - (2\kappa + 18)\bar{\beta}^2, \]
\[ 504\kappa - (14\kappa + 756)\bar{\beta} - (2\kappa + 81)\bar{\beta}^2, \]
\[ 72\kappa - (\kappa + 234)\bar{\beta} - 18\bar{\beta}^2, 2\kappa - 20\bar{\beta} - \bar{\beta}^2, -1 \]

for \( \bar{\beta} > 0 \). For \( \kappa < 1.4 \), this number equals 1 and the claim follows.

Figure 3.3: Left and right panels illustrate different stability conditions for a particle when \( \gamma < \gamma^* \) and \( \gamma > \gamma^* \), respectively (see Lemma 5).

The critical values \( S^{\text{dry}}_{\text{dry}} \) and \( S^{\text{dry}*}_{\text{dry}} \) correspond to saddle-node bifurcations, where the branch of unstable equilibria meets each of the branches of stable equilibria. In particular, \( S^{\text{dry}}_{\text{dry}} \to 1 \) and \( S^{\text{dry}*}_{\text{dry}} \to q(D_{\text{crit}}; D_{\text{dry}}) \) as \( \gamma \to 0 \).

**Corollary 5.** The value \( \gamma = \gamma^* \) corresponds to a cusp bifurcation, at which the two saddle-node bifurcations merge.

In the one-particle case, the unbounded growth obtained for sufficiently large values of \( S_{\text{dry}} \) when \( \gamma = 0 \) is halted when \( \gamma > 0 \) by the finiteness of the supply of water vapor. We
consider next the case of \( N_p = 2 \) and the two coupled differential equations

\[
\dot{D}_1 = r(D_1) \left( S_{\text{dry}} - \frac{\gamma}{2} (D_1^3 - D_{1,\text{dry}}^3) - \frac{\gamma}{2} (D_2^3 - D_{2,\text{dry}}^3) - q(D_1; D_{1,\text{dry}}) \right) \quad (3.17a)
\]

\[
\dot{D}_2 = r(D_2) \left( S_{\text{dry}} - \frac{\gamma}{2} (D_1^3 - D_{1,\text{dry}}^3) - \frac{\gamma}{2} (D_2^3 - D_{2,\text{dry}}^3) - q(D_2; D_{2,\text{dry}}) \right) \quad (3.17b)
\]

and assume without loss of generality that \( D_{1,\text{dry}} < D_{2,\text{dry}} \). Equilibrium is obtained at \( D_1 = D_{1,eq} \) and \( D_2 = D_{2,eq} \) provided that \( q(D_{1,eq}; D_{1,\text{dry}}) = q(D_{2,eq}; D_{2,\text{dry}}) \), in which case the corresponding value of \( S_{\text{dry}} \) may be obtained, for example, from

\[
S_{\text{dry}} = \frac{\gamma}{2} (D_{1,eq}^3 - D_{1,\text{dry}}^3) + \frac{\gamma}{2} (D_{2,eq}^3 - D_{2,\text{dry}}^3) + q(D_{1,eq}; D_{1,\text{dry}}) . \quad (3.18)
\]

The two families of equilibria are shown in Fig. 3.4 in which only one of the families (shown in the left panel) is viable.

![Figure 3.4](image)

Figure 3.4: For two particles, there exist two families of equilibria where only one of them is viable (left panel). In the left panel, the equilibrium is at \( D_{1,eq} \) (green circle) and \( D_{2,eq} \) (one of the black circles). In the right panel, the equilibrium is at \( D_{1,eq} \) (red circle) and \( D_{2,eq} \) (one of the black circles). Equilibrium is obtained at \( D_1 = D_{1,eq} \) and \( D_2 = D_{2,eq} \) provided that \( q(D_{1,eq}; D_{1,\text{dry}}) = q(D_{2,eq}; D_{2,\text{dry}}) \), in which case the corresponding value of \( S_{\text{dry}} \) may be obtained from Eq. 3.18.

From Eq. (3.6) it follows that there exist two disjoint families of equilibria with \( D_{1,eq} < D_{1,\text{crit}} \) and \( D_{1,eq} > D_{1,\text{crit}} \), respectively. From the earlier discussion about the ordering of particle sizes, we conclude that only the former of these two families may be reached in forward time
from the initial dry diameters, independently of the value of $S_{\text{dry}}$.

The necessary condition for the existence of the equilibria were given in the discussion prior to Eq. 3.18. We now proceed to present a sufficient condition for the equilibrium in the following lemma.

Figure 3.5: For $N_p = 2$, depending on the values of $\gamma$ and $S_{\text{dry}}$, three distinct equilibria can appear where one of them is unstable.

**Lemma 6.** For given initial dry diameters $D_{1,\text{dry}}$ and $D_{2,\text{dry}}$, there exists a critical value $\gamma^*$, such that all equilibria of the dynamical system in Eqs. (3.17a-3.17b) that may be reached from the initial dry diameters are globally attractive provided that $\gamma > \gamma^*$. For $\gamma < \gamma^*$, there exists two critical values of $S_{\text{dry}}$, denoted by $S_{\text{dry}}^*$ and $S_{\text{dry}}^{**}$, where $1 < S_{\text{dry}}^* < S_{\text{dry}}^{**}$, such that

- for $0 < S_{\text{dry}} < S_{\text{dry}}^*$, there exists a unique globally attractive equilibrium with $D_{2,\text{eq}} < D_{2,\text{crit}}$;
- for $S_{\text{dry}}^{**} < S_{\text{dry}}$, there exists a unique globally attractive equilibrium with $D_{2,\text{eq}} > D_{2,\text{crit}}$;
Figure 3.6: Phase planes corresponding to different arrangements of equilibrium points. In panel (a), three equilibrium points exist where only two of them are stable. In panel (b), only one equilibrium point exists close to the dry diameters of both particles and the equilibrium is stable. Finally, as shown in panel (c), an equilibrium exists far from the dry diameter of one of the particles. Note that the green line separates the regions with $D_{2,\text{Dry}} > D_{1,\text{Dry}}$ from $D_{1,\text{Dry}} > D_{2,\text{Dry}}$ where only the phase paths in the former are viable and shown in blue. The black curves are $S = q(D_1; D_{1,\text{Dry}})$ or $S = q(D_2; D_{2,\text{Dry}})$.

- for $S_{\text{Dry}}^* < S_{\text{Dry}} < S_{\text{Dry}}^{**}$, there exist two locally attractive equilibria with $D_{2,\text{eq}} < D_{2,\text{crit}}$ and $D_{2,\text{eq}} > D_{2,\text{crit}}$, respectively, separated by an unstable equilibrium with $D_{2,\text{eq}} > D_{2,\text{crit}}$.

Proof. Let $q'_i := \partial D q (D_i; D_{1,\text{Dry}})$ and $r_i := r (D_i; \text{eq})$. Using the result of Corollary 4 and the Hurwitz criterion for the negative stability of the Jacobian of the vector field corresponding to Eqs. (3.17a-3.17b), local asymptotic stability of an equilibrium is obtained provided that the determinant

$$r_1 r_2 \left( \frac{3}{2} D_{1,\text{eq}}^2 q'_2 \gamma + q'_1 \left( q'_2 + \frac{3}{2} D_{2,\text{eq}}^2 \gamma \right) \right)$$

(3.19)
is positive, while the equilibrium is unstable when this quantity is negative. Equivalently, local asymptotic stability follows when

\[
\frac{\gamma q_1'}{3D_{1,eq}^2 \gamma + 2q_1'} > -\frac{q_2'}{3D_{2,eq}^2}
\]  

(3.20)

which is always satisfied when i) \(D_{2,eq} < D_{2,\text{crit}}\), ii) for sufficiently large \(\gamma\), independently of \(D_{2,eq}\), and iii) for sufficiently large \(D_{2,\text{dry}} > D_{2,\text{crit}}\) for fixed \(\gamma\). For a given \(D_{2,eq} > D_{2,\text{crit}}\), it is possible to select \(\gamma\), such that equality holds in Eq. (3.20), corresponding to a saddle-node bifurcation. For the same value of \(\gamma\), the inequality is satisfied in both of the limits \(D_{2,eq} \rightarrow D_{2,\text{crit}}\) and \(D_{2,eq} \rightarrow \infty\). It follows that a second saddle-node bifurcation must exist along the branch of equilibria for \(D_{2,eq} > D_{2,\text{crit}}\).

Figure 3.7: For three particles, there exist four families of equilibria where only one of them is viable (top-left panel). The analysis is analogous to Eq. 3.18 which, for \(N_p = 3\), gives, for example, \(S_{\text{dry}} = \frac{\gamma}{2} (D_{1,eq}^3 - D_{1,\text{dry}}^3) + \frac{\gamma}{2} (D_{2,eq}^3 - D_{2,\text{dry}}^3) + \frac{\gamma}{2} (D_{3,eq}^3 - D_{3,\text{dry}}^3) + q (D_{1,eq}; D_{1,\text{dry}}).\)

48
Figure 3.5 shows two cases in which there exists one (left panel) and three (right panel) equilibria. In Fig. 3.6 the phase plane corresponding to three different cases with $N_p = 2$ are shown.

In the general case of $N_p$ particles, it is again true that $q(D_{i,eq}; D_{i,dry})$ must be identical for all $i = 1, \ldots, N_p$, and that the corresponding value of $S_{dry}$ may be obtained immediately after selection of $\gamma$. Although there exist $2^{N_p-1}$ families of equilibria, the only branch that is reachable from the initial dry diameters again corresponds to $D_{i,eq} < D_{i,crit}$ for $i = 1, \ldots, N_p - 1$. Figure 3.7 shows the case with $N_p = 3$ in which among four families of equilibria only one of them is viable. The corresponding determinant of the Jacobian of the vector field is found to equal

$$\prod_{i=1}^{N_p} r_i \prod_{j=1}^{N_p} q_j' \left( 1 + \frac{3\gamma}{N_p} \sum_{k=1}^{N_p} \frac{D_{k,eq}^2}{q_k'} \right).$$

An analysis similar to that in the proof of Lemma 6 may be used to establish the existence of two saddle-node bifurcations along the equilibrium branch for sufficiently small values of $\gamma$.

The preceding analysis shows that regardless of the number of particles, a certain arrangement of particle sizes occurs in an equilibrium state. In particular, the largest particle, corresponding to the largest dry diameter, takes a relatively large equilibrium diameter while all other particles reach the equilibrium at small diameters close to their corresponding dry diameters.

### 3.2.3 Nonautonomous dynamics

For the general study of the formation of droplets from cloud condensation nuclei, it is necessary to account for temporal variations in temperature, for example, associated with vertical motion in the atmosphere [51]. In this case, the time dependence $S(t)$ of the relative
humidity is related to the distribution of wet diameters through the equation

$$\frac{d}{dt} (P(T)S) = \frac{-3\gamma}{N_p} \sum_{i=1}^{N_p} D_i^2 \dot{D}_i$$

(3.22)

for some temperature-independent constant $\gamma > 0$, or its integral

$$P(T)S - P(T_{dry})S_{dry} = \frac{-\gamma}{N_p} \sum_{i=1}^{N_p} \left(D_i^3 - D_{i,dry}^3\right),$$

(3.23)

where $P(T)$ represents the saturation vapor pressure, and $S_{dry}$ and $T_{dry}$ are the relative humidity and ambient temperature associated with the original emission of dry aerosol particles. For a given temperature variation, we again obtain a closed system of $N_p$ differential equations for the variations in the aerosol wet diameters by solving Eq. (3.23) for $S$ and substituting into Eq. (3.1), where now $\alpha_0$, $\alpha_1$, and $\beta$ are functions of temperature. As in the case of the autonomous dynamics, unbounded particle growth is impossible, since the magnitude of the right-hand side of Eq. (3.23) cannot exceed $P(T_{dry})S_{dry}$. Prior to this, $S$ must fall below 1, ensuring reversal of growth for sufficiently large particle diameters.

It follows from Eqs. (3.1) and (3.6) that, all other things being equal, $\dot{D}$ increases with dry diameter. Particles must thus remain ordered in size for all time according to their initial dry diameters, independently of the time-dependence of the relative humidity $S$ or the ambient temperature $T$. Any characterization of the particle population in terms of bounds on the wet diameters at a certain time, or on the nature of their time histories, is therefore equivalent to an ordered partition of the initial size distribution. As an example, we may loosely speak of a cloud condensation nucleus as activated if it exhibits run-away growth toward the formation of a cloud droplet. It follows that, at any moment in time, the set of activated nuclei are those whose dry diameters exceed some lower bound. Equivalently, all the particles smaller than a nonactivated particle are nonactivated and all the particles larger than an activated particle are also activated.

We let the saturation vapor pressure be represented by the monotonically increasing func-
tion
\[ P(T) := 611.2 \exp \left( 7.45 \ln(10) \frac{T - 273.15}{T - 38} \right), \]  
(3.24)

where \( T \) is in K and \( P(T) \) is in Pa. The dependence of the relative humidity on variations in aerosol diameter is then governed by the parameter
\[ \gamma := \frac{\pi \rho_w c_{\text{dry}} RT_{\text{dry}}}{6 M_w}, \]  
(3.25)

where \( \rho_w \) is the water density, \( M_w \) is the molecular weight of water, \( R \) is the universal gas constant, and \( c_{\text{dry}} \) denotes the number concentration of aerosol particles per unit volume. Note that for fixed \( c_{\text{dry}} \), increasing \( N_p \) will result in an increase in \( V_{\text{comp}} \) and vice versa.

Finally, we parameterize the aerosol dynamics by
\[ \alpha_0(T) := \frac{4 M_w D_v}{\rho_w RT} P^0(T), \]  
(3.26)
\[ \alpha_1(T) := 2 D_v \sqrt{\frac{2 \pi M_w}{RT}}, \]  
(3.27)
\[ \beta(T) := \frac{4 M_w \sigma_w}{\rho_w RT}, \]  
(3.28)

where \( \sigma_w \) is the water surface tension and
\[ D_v := 0.211 \cdot 10^{-4} \left( \frac{T}{273} \right)^{1.94} \]  
(3.29)
is the diffusivity coefficient in m\(^2\)/s for \( T \) in K. The rate of change of \( q(D, T; D_{\text{dry}}) \) (now a function of \( T \) on account of Eq. (3.28)) with respect to \( T \) is everywhere negative. It follows that variations in \( T \) parameterize a foliation of Köhler curves. In particular, the critical supersaturation ratio is a decreasing function of temperature. Unless otherwise stated, Table 3.1 shows the numerical values of the model parameters used in all subsequent computations.

The transient growth of the aerosol particles may now be obtained by explicit substitution of a time profile for the temperature. While for slow enough variations in \( T \), one expects the dynamics to be guided by those found for the system described by Eq. (3.13) (albeit with
Table 3.1: Numerical values of the model parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density of water</td>
<td>( \rho_w )</td>
<td>1000</td>
<td>kg m(^{-3})</td>
</tr>
<tr>
<td>Aerosol number concentration</td>
<td>( c_{\text{dry}} )</td>
<td>( 10^{10} )</td>
<td>m(^{-3})</td>
</tr>
<tr>
<td>Molecular weight of water</td>
<td>( M_w )</td>
<td>( 18 \times 10^{-3} )</td>
<td>kg mol(^{-1})</td>
</tr>
<tr>
<td>Universal gas constant</td>
<td>( R )</td>
<td>8.314472</td>
<td>J mol(^{-1}) K(^{-1})</td>
</tr>
<tr>
<td>Surface tension of water</td>
<td>( \sigma_w )</td>
<td>0.073</td>
<td>J m(^{-2})</td>
</tr>
<tr>
<td>Hygroscopicity</td>
<td>( \kappa )</td>
<td>0.5</td>
<td></td>
</tr>
</tbody>
</table>

A redefined \( \gamma \), this is no longer true in the event of a rapidly rising aerosol layer. As an example, a steady updraft motion of the aerosol layer corresponds, to first approximation, to a linear decrease of the temperature in time. This results in a decreasing saturation vapor pressure and, all other factors being equal, an increase in the relative humidity and the critical supersaturation ratios. For as long as the relative humidity exceeds the value of \( q(D, T; D_{\text{dry}}) \) of a particle of a given dry diameter \( D_{\text{dry}} \), the corresponding aerosol diameter \( D \) grows at a nonzero rate, until such a time that competition for the available water vapor results in a decrease of the relative humidity below the corresponding Köhler curve.

Inspired by Pinsky et al. [53, 54], suppose that the distribution of wet particle diameters relaxes rapidly (independently of the initial distribution and without concomitant changes to the relative humidity, such that \( S(0) \lesssim S_{\text{dry}} = 1 \)) to one consisting of (i) a monodisperse aerosol size distribution with \( pN_p \) activated particles of identical size \( D \) significantly larger than the corresponding dry diameters, and (ii) a distribution of \((1 - p)N_p\) particles whose diameters have equilibrated at values approximately equal to the corresponding dry diameters. In this case, we may expand Eq. (3.22) to obtain

\[
\dot{S} = \frac{1}{P(T)} \left( -3p\gamma D^2 D - \partial_T P(T) TS \right),
\]

(3.30)

For \( D \gg D_{\text{dry}} \), we may approximate \( q(D, T; D_{\text{dry}}) \) in Eq. (3.1) by 1:

\[
\dot{D} = \frac{\alpha_0(T)}{D} (S - 1)
\]

(3.31)
where we also neglect the correction associated with $\alpha_1$. Furthermore, Eq. (3.23) yields
\[ D = \left( \frac{P(T_{\text{dry}}) - P(T)S}{p\gamma} \right)^{\frac{1}{3}}. \] (3.32)

Substitution in Eq. (3.30) then results in the scalar, nonautonomous differential equation
\[ \dot{\chi} = -\frac{1}{P(T)} \left( 3p\frac{\dot{p}}{\dot{\gamma}}\gamma\alpha_0(T)H^{\frac{2}{3}}\chi - \partial_T P(T)\dot{T}(1 - \chi) \right) \] (3.33)

where $\chi := 1 - S$,
\[ H = P(T_{\text{dry}}) - P(T) + P(T)\chi, \] (3.34)
and $T = T_{\text{dry}} - \zeta t$, for constant $\zeta$. In the limit that $S(0) \uparrow 1$, it follows that $\dot{S}(0) > 0$. In this case, we define $\dot{S}$ as the value of $S$ at the first moment that $\dot{S} = \dot{\chi} = 0$, and let $\dot{T}$ denote the temperature when this condition is reached.

We make plausible the existence of such a maximum in the limit as $S(0) \uparrow 1$ (and, consequently, as $H(0) \downarrow 0$) by replacing $\alpha_0(T)$ with a positive constant $\alpha_0$. In this case, suppose that $\dot{\chi}$ is negative and bounded away from 0 for all time, such that $-\chi(t) > kt$ for some $k > 0$. Then, since
\[ \frac{dH}{dt} = -3p\frac{\dot{p}}{\dot{\gamma}}\gamma\alpha_0 H^{\frac{2}{3}}\chi, \] (3.35)

it follows that
\[ H(t)^{1/3} \sim \sqrt{\left( \frac{1}{\int_0^t -2\chi(\tau) \, d\tau} \right)} \] (3.36)

By the assumptions on $\chi$, the right-hand side is bounded from below by $\sqrt{kt}$. It follows that the first term in Eq. (3.33) must eventually exceed, in magnitude, the value of the second term in this equation, in contradiction with the assumptions on $\chi$, and the claim follows.

It follows from Eq. (3.33) that
\[ \dot{\chi}' = \frac{1}{P} \left( 3p\frac{\dot{p}}{\dot{\gamma}}\gamma\alpha_0 H^{\frac{1}{3}}\dot{\chi} - \partial_T P(1 - \dot{\chi}) \right) \] (3.37)

where $\dot{\chi}(T_{\text{dry}} - \zeta t) := \chi(t)$. It follows that the dynamics of $\dot{\chi}$ is parameterized by the ratio
\( \frac{p^2}{\zeta} \) governing the balance between depletion of the ambient humidity due to the growth of aerosol particles and the increase in the humidity due to a decreasing saturation vapor pressure, but not by either \( p \) or \( \zeta \) separately. Each value of this ratio thus corresponds to a unique pair of values of \( \hat{S} \) and \( \hat{T} \), as illustrated in Fig. 3.8. The two curves included in this figure correspond to projections of the one-dimensional solution manifold to the discretized boundary-value problem given by a collocation approximation of Eq. (3.37) with \( \hat{\chi}(T_{\text{dry}}) = \hat{\chi}'(\hat{T}) = 0 \) in terms of an unknown piecewise-polynomial approximant of \( \hat{\chi} \) on a fixed mesh with 15 equal-sized intervals and polynomial degree 5, and for unknown \( \hat{T} \). The results show dramatic variations in the values of \( \hat{S} \) and \( \hat{T} \) for very small values of the ratio \( \frac{p^2}{\zeta} \), corresponding to the sought-after maximum of \( S \) being reached only after a significant change in temperature. In the extreme case, for which \( \hat{S} \) and/or \( \hat{T} \) lie outside of physically meaningful ranges, the validity of the model formulation is naturally put in question. Consequently, with Fig. 3.8 in mind, we restrict attention below to values of the ratio \( \frac{p^2}{\zeta} > 1 \).

![Figure 3.8](image)

Figure 3.8: Both \( \hat{S}_v \) and \( \hat{T} \) deviate dramatically from their physically meaningful ranges as \( \frac{p^2}{\zeta} \to 0 \). Here, the circle denotes the initial solution guess to the boundary-value problem obtained by imposing the additional conditions (3.40-3.42) with \( \zeta = 0.01 \), \( E[D_{\text{dry}}] = 7 \times 10^{-8} \), and \( V[D_{\text{dry}}] = 1 \times 10^{-14} \).

The fraction \( p \) is related to the initial probability distribution \( f(D_{\text{dry}}) \) of dry aerosol
diameters through the equality

\[ p = \int_{D_{\text{dry}}^*}^{\infty} f(D_{\text{dry}}) \, dD_{\text{dry}}, \quad (3.38) \]

for some cutoff diameter \( D_{\text{dry}}^* \). For example, in terms of the log-normal density

\[ f_{\text{l.n.}}(D_{\text{dry}}) := \frac{1}{D_{\text{dry}}\sqrt{2\pi\sigma}} e^{-\frac{(\ln D_{\text{dry}} - \mu)^2}{2\sigma^2}}, \quad (3.39) \]

it follows that

\[ p = \int_{D_{\text{dry}}^*}^{\infty} f_{\text{l.n.}}(D_{\text{dry}}) \, dD_{\text{dry}} = \frac{1}{2} \left( 1 + \text{erf} \left( \frac{\mu - \ln D_{\text{dry}}^*}{\sqrt{2\sigma}} \right) \right). \quad (3.40) \]

For later reference, we note the corresponding expressions for the expected value \( E[D_{\text{dry}}] = e^{\mu + \frac{\sigma^2}{2}} \) and variance \( V[D_{\text{dry}}] = e^{2\mu + \sigma^2} \left( e^{\sigma^2} - 1 \right) \).

We finally arrive at a closed nonlinear boundary-value problem for the time history \( S(t) \) and the unknown fraction \( p \) of activated particles by imposing a condition on \( D_{\text{dry}}^* \) that captures the nature of the initial inflationary relaxation phase. Activation of an aerosol particle of a given dry diameter, as defined by an apparent inflationary phase of growth of the wet aerosol diameter, requires that the relative humidity exceed the critical saturation ratio at the instant that the wet diameter coincides with the corresponding supersaturation point. If we approximate \( q(D,T;D_{\text{dry}}) \) by its restriction to \( T = \hat{T} \), then a necessary, but not sufficient, condition for this to hold is that the maximum relative humidity \( \hat{S} \) exceed the critical saturation ratio \( q(D_{\text{crit}}, \hat{T}; D_{\text{dry}}) \). We rely on this condition in a first approximation of the fraction of activated particles for given values of \( \gamma \) and the rate of decrease of the ambient temperature. Specifically, we assume that \( D_{\text{dry}}^* \) is implicitly defined by the pair of scalar equations

\[ q(D_{\text{crit}}, \hat{T}; D_{\text{dry}}^*) = \hat{S}, \quad (3.41) \]

\[ \partial_D q(D_{\text{crit}}, \hat{T}; D_{\text{dry}}^*) = 0. \quad (3.42) \]
In this case, the dimensional deficit of the nonlinear boundary-value problem (i.e., the excess of the number of unknowns to the number of equations) equals 0, i.e., at least one model parameter must be allowed to vary in order to obtain a one-dimensional family of solutions through a given solution.

The results of numerical parameter continuation along solution families of the nonlinear boundary-value problem given by Eqs. (3.37), (3.40-3.42), and \( \dot{\chi}(T_{\text{dry}}) = \dot{\chi}'(\dot{T}) = 0 \) are shown in Fig. 3.9. Here, orthogonal collocation was again used to discretize the governing differential equation, with a corresponding piecewise-polynomial approximation of \( \dot{\chi}(T) \) on a uniform mesh with 15 intervals and polynomial degree 5, and with initial solution guess found during the continuation run that generated Fig. 3.8. Figure 3.9 considers the case with additional imposition of fixed values \( p \) for variable expected value \( E[D_{\text{dry}}] \) and variance \( V[D_{\text{dry}}] \).

![Figure 3.9: Several values of \( p \), as shown in the colorbar, are fixed to find the variations of \( E[D_{\text{dry}}] \) and \( V[D_{\text{dry}}] \).](image-url)
3.2.4 Kinetic limitation

The simplifying assumptions made in the heuristic derivation above tend to overestimate the number of activated particles, as they fail to account for the possibility that a particle, whose critical supersaturation ratio is less than the maximum relative humidity, nevertheless reaches a maximum diameter without undergoing an inflationary growth. This phenomenon is termed *kinetic limitation* [34] and particles affected by this phenomenon are said to be *kinetically limited*. As the error associated with this effect may be significant, it is important to evaluate its influence on predictions of the fraction of activated particles.

There are at least two relevant mechanisms that fall under the umbrella of kinetic limitation, for which we use the terminology in [34]. In the “evaporation mechanism”, the relative humidity drops below the critical saturation ratio of a given particle before the critical diameter $D_{\text{crit}}$ is reached (see the left panel of Fig. 3.10), resulting in the immediate deactivation of the particle. In the “deactivation mechanism”, shown in the right panel of Fig. 3.10, the relative humidity exceeds the critical saturation ratio when the particle diameter reaches its critical value, but falls below the corresponding Köhler curve shortly thereafter, after which evaporation again results in a shrinking diameter. A third mechanism [34], referred to as the *inertial mechanism*, described particles of very large dry diameter and correspondingly very low critical supersaturation and large critical diameter. In this case, even though the critical diameter may not be reached, growth may continue for such a long time that, for practical purposes, the particles may be considered activated. The latter mechanism is out of the scope of our study and will not be considered here.

3.3 Numerical study

In lieu of the heuristic analysis in the previous section, we seek a systematic computational framework for the study of the dependence of the fraction of particles affected by kinetic limitation on the statistical properties of the initial distribution of dry diameters.
Figure 3.10: Kinetic limitation: while the maximum relative humidity along the projected trajectory (green curves) exceeds the critical saturation ratio along the corresponding $D$-isocline (blue curves), growth is reversed as the trajectory reaches the isocline on either side of the critical diameter.

3.3.1 Forward simulation

In the continuum limit $N_p \to \infty$, the particle distribution can be specified by a number concentration density function $n(D_{\text{dry}})$, so that $n(D_{\text{dry}})dD_{\text{dry}}$ is the number concentration of particles with dry diameters in the range $[D_{\text{dry}}, D_{\text{dry}} + dD_{\text{dry}})$. Then Eq. (3.22) becomes

$$\frac{d}{dt}(P(T)S) = -\frac{3\gamma}{c_{\text{dry}}} \int_0^\infty D(D_{\text{dry}}, t)^2 \frac{\partial}{\partial t} D(D_{\text{dry}}, t) n(D_{\text{dry}}) dD_{\text{dry}}, \quad (3.43)$$

where $D(D_{\text{dry}}, t)$ is a function that gives the wet diameter at time $t$ of particles with dry diameter $D_{\text{dry}}$, and thus evolves pointwise according to Eq. (3.1).

This continuum description can be discretized by an equally-weighted sampling of $N_p$ particles from $n(D_{\text{dry}})$ and approximating (3.43) by the corresponding Riemann sum, yielding Eq. (3.22).

As an example, we analyze numerically the dynamics of a collection of $N_p$ particles with dry diameters corresponding to a uniform discretization of the cumulative probability density of the log-normal density in Eq. (3.39). Specifically, we assume that the dry diameter $D_{i,\text{dry}}$
of $i$-th particle satisfies the equation

$$F_{i.n.}(D_{i,\text{dry}}) = \varepsilon + \frac{1 - 2\varepsilon}{N_p - 1} (i - 1)$$

(3.44)

for some $\varepsilon > 0$, where

$$F_{i.n.}(D_{\text{dry}}) = \int_0^{D_{\text{dry}}} f_{i.n.}(D_{\text{dry}}) \, dD_{\text{dry}} = \frac{1}{2} \left( 1 - \text{erf} \left( \frac{\mu - \ln D_{\text{dry}}}{\sqrt{2}\sigma} \right) \right) .$$

(3.45)

In the numerical results reported throughout the remainder of the study, $N_p = 3000$, $S_{\text{dry}} = 0.98$, $T_{\text{dry}} = 290$ K, and $\varepsilon = 1/3000$.

Consider, specifically, a transient numerical simulation with $E[D_{\text{dry}}] = 7 \times 10^{-8}$, $V[D_{\text{dry}}] = 10^{-14}$, $\zeta = 0.01$, and $N_p = 500$ integrated over 100 units of time. Let the initial conditions $S(0)$ and $D_i(0)$ be given by the relative humidity $S_{eq}$ and distribution of diameters $D_{i,eq}$ corresponding to the asymptotically stable equilibrium below the critical supersaturation point obtained for constant temperature $T = T_{\text{dry}}$, the given distribution of dry diameters, and the corresponding relative humidity $S_{\text{dry}}$. It follows that

$$S_{eq} - S_{\text{dry}} + \frac{\gamma}{P(T_{\text{dry}})} \frac{1}{N_p} \sum_{i=1}^{N_p} (D_{i,eq}^3 - D_{i,\text{dry}}^3) = 0$$

(3.46)

and

$$S_{eq} - q(D_{i,eq}, T_{\text{dry}}; D_{i,\text{dry}}) = 0$$

(3.47)

for $i = 1, \ldots, N_p$.

As seen from the resultant time histories in Fig. 3.11, a subset of the particles, namely those of sufficiently large dry diameter, exhibits an inflationary growth phase (around $t = 37$), after which the corresponding diameters continue to grow throughout the extent of the simulation. Particles with intermediate-size dry diameters are kinetically limited and fail to become activated, in spite of a brief inflationary phase. Finally, particles of sufficiently small dry diameters fail to enter an inflationary growth as the relative humidity never exceeds the corresponding critical saturation ratio. In the simulation, the 415th particle consists of particles, for which the relative humidity does not exceed the corresponding critical saturation
ratio. Similarly, the 425th particle consists of those particles of largest dry diameter whose corresponding wet diameter reaches a maximum during the simulation. By the dynamically consistent ordering of diameters, it follows that particles in particles 416 through 425 are all kinetically limited.

![Figure 3.11: Time evolution of particles sizes. The red curve represents the 415th particle, while the blue and green curves represent particles 424 and 425, respectively. All curves in cyan are affected by kinetic limitation.](image)

The time history \( S(t) \) for the relative humidity shown in Fig. 3.11 captures the collective influence on the availability of water vapor from the variations in the distribution of wet diameters approximated by the discretized particle model. We can use \( S(t) \) (and the associated variations in temperature \( T \)) as an exogenous signal to drive the time history \( D_0(t) \) of a test particle with dry diameter \( D_{0,\text{dry}} \) according to Eq. (3.1). Note that \( D_{0,\text{dry}} \) can take any value and is not ordered with respect to the values \( D_{i,\text{dry}} \). A result of such an analysis is shown in Fig. 3.12(a). Here, the vertical axes represent the wet diameter \( D_{0,\text{max}} \) and the elapsed time \( t_{0,\text{max}} \) at an instant of vanishing right-hand side in Eq. (3.1), corresponding to a turning point in the growth of the corresponding wet diameter. These values coincide with the corresponding critical values \( D_{i,\text{max}} \) and elapsed times \( t_{i,\text{max}} \) in the discretized particle
model, provided that \( D_{0,\text{dry}} = D_{t,\text{dry}} \). As is evidenced by the figure, \( D_{0,\text{max}} \) and \( t_{0,\text{max}} \) depend sensitively on changes to the dry diameter \( D_{0,\text{dry}} \) close to the transition to particles with inflationary growth (which fail to reach such an instant during the 100 units of integration time).

An alternative representation of the data in Fig. 3.12(a) is given in Fig. 3.12(b), where we observe a similar sensitivity in graphing the ratio \( t_{0,\text{max}}/t_{425,\text{max}} \) versus the ratio \( \hat{D}_0/\hat{D}_{425} \) of wet diameters at the instant that \( \dot{S} = 0 \). We generalize this observation to an arbitrary discretization of the initial distribution of dry diameters (and the corresponding time histories) by letting \( \lambda \) denote the slope at \((1,1)\) of the graph of \( t_{0,\text{max}}/t_{\text{last},\text{max}} \) versus \( \hat{D}_0/\hat{D}_{\text{last}} \). Here, the subscript \( \text{last} \) represents the index of the largest particle to reach a maximum during the simulation. In particular, we argue that the discretized model resolves the transition between kinetically limited particles and fully activated particles with satisfactory detail provided that \( \lambda \gg 1 \).

![Graphs showing particle diameter and time ratios](image)

Figure 3.12: Diameter of particles at their deactivation points undergoes a very steep change in a narrow region corresponding to dry diameters of particles that separate activated particles from others.

As shown in Fig. 3.13, the number of activated particles converges as we increase the number of particles, i.e., a finer discretization of the size distribution. If \((N_k,p_k)\) are the data points, we expect that \( p_k - p^* = C N_k^r \) for some \( C \) and \( r \), where \( p^* \) is the value that \( p_k \) is converging to. We obtain \( p^* = 0.1427 \), \( C = 0.08017 \), and \( r = -0.3327 \).
Figure 3.13: Fraction of activated particles ($p$) converges to a certain value as we increase the number of particles ($N_p$). The dashed line shows the converging value of $p$. The right panel shows the logarithmic convergence of error.

3.3.2 A boundary-value problem

Inspired by the time histories found using direct numerical simulation, we seek a boundary-value formulation such that every point on the corresponding solution manifold corresponds to the time histories of particles with the same fraction of activated, but not kinetically limited particles. We consider two alternative heuristic criteria that each characterize an activated particle and explore the predicted fraction of kinetically limited particles under variations in the statistical moments of the initial distribution of dry aerosol diameters. In particular, given the discretization of the initial distribution of dry diameters in Eq. (3.44), let $p$ denote the fraction of activated particles exhibiting runaway growth throughout the duration of the relevant time histories. For an a priori value of $p$, it follows that the last subscript represents a known bin index.

Now consider a three-stage segmentation of the continuous time histories of the relative humidity $S$, the wet aerosol diameters $D_i$, and the wet diameter $D_0$ of a test particle with $D_{0,\text{dry}} = D_{\text{last, dry}} - \varepsilon$ for some $\varepsilon > 0$. The first segment is characterized by initial conditions for $S$ and $D_i$, for $i = 0, \ldots, N_p$ coincident with the equilibrium values $S_{\text{eq}}$ and $D_{i,\text{eq}}$ (below the critical saturation point) obtained from Eq. (3.46) and Eq. (3.47), where the latter equation
applies also to $i = 0$. The first segment ends after an elapsed time $\hat{t}$ at a local maximum of $S(t)$. Similarly, the second and third segments are assumed to terminate after total elapsed times $t_{0,\text{max}}$ and $t_{\text{last, max}}$, at points where $S$ equals $q(D_0, T; D_{0,\text{dry}})$ and $q(D_{\text{last}}, T; D_{\text{last, dry}})$, respectively.

Let $\hat{S}$, $\hat{T}$, and $\hat{D}_i$, for $i = 0, \ldots, N_p$, denote the relative humidity, ambient temperature, and particle diameters at $t = \hat{t}$. Then, a particle of dry radius $D_{i,\text{dry}}$ is said to be activated according to the first criterion if $\hat{S}$ exceeds the critical saturation ratio $q(D_{\text{crit}}, \hat{T}; D_{i,\text{dry}})$.

Alternatively (cf. [32]), a particle is said to be activated according to the second criterion if its diameter $\hat{D}_i$ exceeds the critical supersaturation point $D_{\text{crit}}$ associated with the corresponding dry diameter $D_{i,\text{dry}}$ and $T = \hat{T}$.

Finally, in order to ensure that the discretization resolves the transition between kinetically limited particles and fully activated particles with satisfactory detail, we impose the additional condition that the slope $\lambda$ defined in the previous section, obtained here from a finite-difference approximation, equal some given large positive number $\lambda^*$. Collectively, the corresponding boundary conditions constitute algebraic constraints on the unknown time histories of the form

$$
\begin{cases}
S(0) - S_{\text{dry}} + \frac{\gamma}{P(T_{\text{dry}})N_p} \sum_{i=1}^{N_p} (D_i^3(0) - D_{i,\text{dry}}^3) \\
S(0) - q(D_0(0), T_{\text{dry}}; D_{0,\text{dry}}) \\
\vdots \\
S(0) - q(D_{N_p}(0), T_{\text{dry}}; D_{N_p,\text{dry}}) \\
\hat{S}(\hat{t}) \\
S(t_{0,\text{max}}) - q(D_0(t_{0,\text{max}}), T(t_{0,\text{max}}); D_{0,\text{dry}}) \\
S(t_{\text{last, max}}) - q(D_{\text{last}}(t_{\text{last, max}}), T(t_{\text{last, max}}); D_{\text{last, dry}}) \\
\hat{D}_{\text{last}}(t_{\text{last, max}} - t_{0,\text{max}}) - \lambda^* t_{\text{last, max}} (\hat{D}_{\text{last}} - \hat{D}_0)
\end{cases} = 0 \quad (3.48)
$$
3.3.3 Numerical Results

In order to obtain a one-parameter family of solutions to the boundary-value formulation in the previous section, we apply path following techniques to a suitable discretization of the system of ordinary differential equations, using the Matlab-based computational continuation core coco. Specifically, we approximate each of the unknown time histories in terms of continuous, piecewise-polynomial functions, and require that these satisfy the boundary conditions, as well as the governing differential equations at a collection of discrete collocation nodes (the so-called collocation conditions). Finally, we ensure that the number of unknown continuation variables exceeds the number of corresponding algebraic equations by one.

Given the natural segmentation of the boundary-value problem, we treat each of the segments separately and regain the original transient time histories by imposing continuity across the segment boundaries. On each segment, we shift and scale the time variable in order to obtain a system of differential equations on the interval \([0, 1]\), parameterized by the interval duration, per the following relationship

\[
\frac{dx(t)}{dt} = f(x(t)), \quad t \in [t_b, t_e] \implies \frac{dx(t(\tau))}{d\tau} = (t_e - t_b) f(x(t(\tau))), \quad \tau \in [0, 1]
\]  

(3.49)

for \(t(\tau) = t_b + \tau (t_e - t_b)\). On each segment, each scalar-valued, piecewise-polynomial approximant is naturally associated with a partition of the time domain in terms of a collection of mesh nodes and mesh intervals. We typically parameterize the approximant on each such interval in terms of a Lagrange interpolant through unknown values on a set of base points. The number of base points exceeds the polynomial degree by one.

An asynchronous mesh is characterized by more than one collection of mesh nodes across the set of approximants. When the union of all mesh nodes across the different approximants equals the mesh nodes for each of the approximants, we speak of a synchronous mesh. A synchronous mesh is said to be uniform, if the mesh intervals are all equal in length, and non-uniform otherwise. A mesh is adaptive if it is updated during continuation to ensure that an estimate of the discretization errors remains bounded below a critical threshold. Such updates may include changes to the location of individual nodes (\(r\)-type), as well as the number of mesh nodes or the size of the intervals (\(p\)-type and \(h\)-type, respectively).
The imposition of collocation conditions on the polynomial approximants requires the use of Lagrange interpolation in order to obtain algebraic conditions on the unknown base point values. In the case of a synchronous mesh, there exists a linear, and variable-independent relationship between the values of the polynomial functions and their derivatives at the collocation nodes and base points, respectively. Mesh updates may be reduced to discrete changes to scaling constants in the collocation conditions, without any change to this linear relationship. In contrast, in an asynchronous mesh, each scalar-valued approximant is associated with a distinct such linear relationship. In this case, any updates to the individual meshes requires recomputing the coefficients of the corresponding linear maps.

It is clear that the choice of mesh affects both the accuracy of the discretization, as well as the computational cost associated with applying linear and nonlinear solvers to the discretized boundary-value problem. In the case of a sufficiently high-dimensional dynamical system, it is quite conceivable that the choice may affect the feasibility of the analysis, as the required intermediate storage exceeds available machine memory. Ultimately, the choice of mesh must be guided by the observed nature of the relevant time histories. Large differences between the characteristic times scales for individual state variables at the same instant, between the characteristic time scales for a single state variable at different instants, or between the characteristic time scales for a single variable at the same instant but for different points on the solution manifold to the boundary-value problem suggest the use of asynchronous, non-uniform, and adaptive meshes, respectively.

For the analysis reported below of the aerosol boundary-value problem, we use an asynchronous, non-uniform, adaptive mesh, in order to account for dissimilar growth rates and significant sensitivity in the behavior of kinetically limited particles to changes in the system parameters. The detailed implementation of the mesh selection and adaptation algorithm is discussed in Appendix B. Here, the state variables associated with the discretized distribution of particle diameters are further discretized in time according to three distinct meshes. The first of these describes particles whose diameters remain close to their initial diameters and exhibit relatively slow dynamics. The second mesh describes particles whose diameters exhibit more rapid variability, e.g., a sudden reduction in size following the termination of initial growth. Finally, the third mesh describes particles with relative large dry diameters,
which exhibit intermediate (and persistent) growth rates. As mentioned in Appendix B, an optimal grouping of variables is achieved when the accumulated error of each variable is a sufficient condition for group assignment. However, the kinetically limited variables in the second and third segments do not satisfy this condition and, hence, we assign a separate group for all kinetically limited particles. Moreover, due to the high sensitivity of the time at which small regions of fast behavior occur with respect to the varying parameters, the mesh in these two groups are selected uniformly, such that for each group \( h = \max_i h_i \), where \( h_i \) are the mesh interval sizes obtained from the mesh selection algorithm, and \( h \) is the single mesh interval size used for the uniform mesh partition. Note that in each adaptation step, the first kinetically limited particle is identified and used for the selection of groups. In all cases, we use a separate mesh for the piecewise-polynomial approximant of the relative humidity (noting that this is affected by all the different time scales present in the dynamics of the particle diameters).

We proceed to implement the asynchronous collocation scheme described in the Appendix B with \( m = 4 \) and \( TOL = 10^{-10} \). For each of the three cases reported below, an initial solution is obtained using forward simulation with MATLAB for \( V[D_{\text{dry}}] = 1 \times 10^{-14} \) and \( E[D_{\text{dry}}] = 7 \times 10^{-8}, 8 \times 10^{-8}, 9 \times 10^{-8} \), and the assignment of individual particle time histories to one of the three meshes, as well as the number and location of the corresponding mesh nodes, are updated adaptively every five continuation steps. In each case, a one-parameter family of solutions to the discretized boundary-value problem results by allowing two of the remaining model parameters to vary.

As an example, the left panel in Fig. 3.14 shows the projection of the solution manifolds onto the \( E[D_{\text{dry}}] - V[D_{\text{dry}}] \) plane in the case that \( \zeta = 0.01 \) and for \( p \) equal to 0.147, 0.162, and 0.176, respectively. Unexpectedly, these curves are straight lines going through the origin with slopes equal to \( 4.87 \times 10^{-7}, 3.76 \times 10^{-7}, \) and \( 3.14 \times 10^{-7} \), respectively. This suggests that if there exists a monotonically decreasing function that relates the fraction \( p \) of activated particles to the corresponding slope, then the value of \( p \) may be predicted directly from the mean and variance of the original log-normal distribution. The right panel of Fig. 3.14 shows the difference between the predicted fraction of activated particles using each of the heuristic criteria compared to the curve with \( p = 0.147 \) in the left panel of Fig. 3.14. As
expected, in this case as well as the other two cases, both heuristic criteria overestimate the number of activated particles, with a larger fraction of particles activated according to the first criterion than those activated according to the second criterion. In the parameter ranges shown in Fig. 3.14, the predicted values using the second criterion achieve a local maximum while the prediction according to the first criterion increases monotonically with $E[D_{\text{dry}}]$.

Figure 3.14: Bifurcation diagrams for fixed fractions of particles that form cloud droplets. The right panel shows the difference between the predicted values and $p = 0.147$.

For $p = 0.147$, the panels in Fig. (3.15) show the relationships between the rate of temperature decay $\zeta$ and the mean or variance, respectively, of the initial diameter distribution along the corresponding solution manifold. As seen in panel (a), for fixed $V[D_{\text{dry}}]$, $\zeta$ increases with $E[D_{\text{dry}}]$ in order to maintain a fixed fraction of activated particles. In contrast, as seen in panel (b), for fixed $E[D_{\text{dry}}]$, $\zeta$ decreases with increasing values of $V[D_{\text{dry}}]$. The difference between the corresponding predictions of the fraction of activated particles according to each of the two heuristic criteria and the actual fraction are shown in panels (c) and (d). We note significant differences, associated with kinetically limited particles, between these predictions and the actual value of $p$. Indeed, the prediction error incurred by the first heuristic criterion grows with $E[D_{\text{dry}}]$ (and the associated reduction in the rate of temperature decay), whereas that incurred by the second heuristic criterion improves after an initial brief increase.
Figure 3.15: Panels (a) and (b) show the bifurcation curves for fixed values of $V[D_{dry}]$ and $E[D_{dry}]$, respectively. Panels (c) and (d) show the difference between the corresponding predicted fractions of activated particles and its actual value.

Finally, we note that when asynchronous collocation algorithm is utilized in this problem, the total number of unknown basepoint values are around $1.5 \times 10^5$ compared to the synchronous mesh in which around $1 - 1.5 \times 10^6$ basepoints exist. In the least efficient case where a uniform and synchronous mesh is used, the total number of basepoints in this problem can be as high as $1.5 \times 10^7$ which justifies the use of a nonuniform and asynchronous mesh in this problem.

### 3.4 Conclusions

As discussed in the introduction, this study aimed at addressing the commonly neglected, but sometimes significant, effect of kinetic limitation in the estimation of cloud particle activation. The boundary-value formulation proposed in this work is not based on simplifying assumptions in order to be capable of accurately account for kinetically limited particles. Although the formulation was a powerful tool for studying the parameter-dependence of ki-
netic limitation, the discretization of the equations through collocation method leads to an excessively large problem that cannot be handled using in-house hardware facilities. Consequently, a discretization algorithm was introduced in which the associated mesh partitions for each variable can be selected independently. This asynchronous collocation method drastically reduced the size of the problem, making the proposed boundary-value formulation a feasible candidate for accurate study of large-scale systems with slow-fast dynamics.

The developed asynchronous collocation scheme in this work comes with an overhead mostly due to the interpolation of base-point values at desired collocation points. Although the interpolating operators are generated once in every mesh adaptation step, which reduces the overhead, several criteria must be considered in utilizing the asynchronous collocation scheme. Perhaps the most natural question to answer is if the asynchronous collocation can indeed increase the efficiency of the solution strategy considering the overhead of the method. Although the answer to this question is often apparent, like in this study, in some cases were the size of the problem is not too big or the difference in the behavior of variables is not very distinct, it might be advised that both synchronous and asynchronous schemes be used initially to select the most efficient one.

The next step is grouping the variables based on their behaviors into synchronous subgroups. The grouping formulation derived in this work performs optimally only under certain conditions as discussed in Appendix B. This demands for some *a priori* knowledge of the behavior of solutions to arrive at a grouping that minimizes the size of the problem for a desired tolerance. Further reduction in the overhead of the algorithm can be achieved if the interpolation at collocation points is limited to be conducted only for variables that are coupled in the equations. Although this selection was straightforward in the equations we studied in this study, this distinction might be cumbersome to be done manually in general, hence, demanding for an automated algorithm that parse through the equations and find the required interpolation for coupled variables.

Finally, in the lack of rigorous error analysis of the asynchronous collocation scheme, we used a mesh selection strategy that treat each variable (or group) individually. Although this approach seems to be adequate for the asynchronous collocation, more sophistication can be employed in the mesh selection of asynchronous discretization as the error bounds of
groups can be defined independently.
CHAPTER 4

SINGULARITIES IN THE GOVERNING FORMULATION OF BEAM STRUCTURES

4.1 Introduction

Parameter\(^1\) continuation affords an efficient computational methodology for exploring categories of system response by largely eliminating the need for costly simulations. In [56], such techniques were illustrated in the context of the response of a suitably constrained beam to harmonic excitation providing for a study of two-parameter bifurcation curves and sensitivity analysis. In [56], the methodology was further investigated for a family of beam models, demonstrating the coupling of parameter continuation (using a MATLAB-based computational continuation core (COCO) developed by Dankowicz and Schilder [5]) and the commercial multiphysics simulation package COMSOL. As an example, Fig. 4.1 shows several frequency-response curves obtained using an in-house finite-element discretization of the hinged-hinged boundary-value problem under variations in the excitation frequency and for different strengths of the geometric nonlinearity. The curves illustrate the characteristic hardening effect of the nonlinearity, which becomes further pronounced as the value of \(\alpha\) is increased. As a result of hardening, a range of values of the excitation frequency is found, over which two stable, periodic responses coexist with a single unstable steady-state oscillation.

A passing note in [56] remarked on singularities observed in the formulation of a continuation problem for the case of a clamped-clamped beam. These singularities were found to be associated with translational symmetries in a finite-element discretization of a mixed form version of the governing integro-partial differential equation. It is the objective of

\(^1\)The material in this chapter is collaborative efforts with Prof. Harry Dankowicz and accepted for publication in the ASME Journal of Computational and Nonlinear Dynamics [55].
the present study to provide a further in-depth analysis of these singularities, to investigate their occurrence also in alternative formulations that support continuation in the case of the clamped-clamped boundary conditions, as well as to propose ways of resolving the degeneracy in the mixed-form continuation problem.

To this end, in this chapter, we present distinct discretizations of equivalent forms of the so-called Mettler equation, governing the transverse oscillations of a slender beam while accounting for geometric nonlinearities associated with internal tension. In these cases, candidate finite-element discretizations of the original scalar equation and of a mixed-form representation are employed to arrive at a corresponding continuation problem. The remainder of the study then treats the mixed-form representations and the resolution of singularities mentioned above. In particular, it is shown that degenerate continuation problems arise from spatial and temporal group symmetries of the governing differential-algebraic boundary-value problems obtained from a spatial finite-element discretization or their further temporal dis-
cretization using collocation. We proceed to propose a resolution to these singularities, e.g.,
by means of symmetry reduction, or by breaking the symmetries arising from the discretiza-
tion. Numerical experiments are then performed to characterize the convergence rates of
the different discretization schemes, both in the case of a linear beam model for which exact
solutions are available, and for the fully nonlinear model using Richardson extrapolation.

4.2 A scalar beam model

4.2.1 Problem definition

We consider the flexural transverse motion of a slender beam of length \( l \), cross-sectional
area \( A \), area moment of inertia \( I \), volume density \( \rho \), and Young’s modulus \( E \), relative to
an undeformed straight configuration between two stationary ends, as represented by the
displacement \( v(s, t) \), scaled by \( l \), and as a function of a longitudinal coordinate \( s \in \Omega := [0, 1] \),
scaled by \( l \), and time \( t \), scaled by \( l^2 \sqrt{\rho A/EI} \). From [57], it follows that the effects on the
beam dynamics of linear elasticity and external viscous damping, geometric nonlinearities
due to mid-plane stretching, and a transverse excitation may be captured by the scalar,
nonlinear, integro-partial differential equation

\[
v_{tt} + v_{ssss} + cv_t - \frac{\alpha}{2} \mathcal{T} v_{ss} = f\tag{4.1}
\]

where

\[
\mathcal{T} = \int_{\Omega} v_s^2 ds \tag{4.2}
\]

models the nondimensionalized longitudinal tension, \( f(s, t) \) is the known external load den-
sity per unit length, scaled by \( l^3/EI \), and partial derivatives of \( v \) are denoted by subscripts
representing the corresponding independent variables and the order of differentiation. Here,
the linear damping coefficient \( c \) equals \( 2\zeta\omega_1 \), where \( \zeta \) is the damping ratio and \( \omega_1 \) is the first
natural frequency of the corresponding Euler-Bernoulli beam in the absence of the geometric
nonlinearity, and the nondimensional parameter

\[ \alpha := \frac{Al^2}{I} \]  

(4.3)

is the square of the slenderness ratio.

In the preceding publication [56] and throughout this study, our main concern is with periodic, steady-state motion of the beam, with period \( T := \frac{2\pi}{\omega} \), in response to a uniform, harmonic excitation \( f(s, t) = \Lambda \cos \omega t \) (as suggested in Fig. 4.2), with excitation amplitude \( \Lambda \) and excitation frequency \( \omega \), and for two distinct sets of boundary conditions, namely the case of a clamped-clamped beam, for which \( v \) and \( v_s \) both vanish at \( s = 0 \) and \( s = 1 \), as well as that of a hinged-hinged beam, for which \( v \) and \( v_{ss} \) both vanish at \( s = 0 \) and \( s = 1 \). For small excitation amplitudes, where the effects of the nonlinearity are moderate, these solutions may be obtained using standard perturbation methods. For larger excitation amplitudes, and especially when the effects of the nonlinearity become pronounced, numerical path-following techniques applied to various discretizations of the boundary-value problem may be employed to study families of periodic responses, parameterized by individual or collections of system parameters.

![Figure 4.2: Schematic of the beam with clamped-clamped boundary condition.](image)
4.2.2 Finite-difference discretization

The derivatives in the governing partial-differential equation (4.1) may be approximated in terms of finite differences based on a rectangular grid of nodal points on the domain \([0, 1] \times [0, T]\). Specifically, let

\[
v(i,j) := v \left( \frac{i - 1}{N}, \frac{J - 1}{N} \right)
\]

(4.4)

for \(i = 1, \ldots, N + 1\) and \(j = 1, \ldots, N + 1\), denote the nodal values on a rectangular mesh that is uniform in each of the two independent variables. Then, the corresponding discrete set of algebraic equations is given by

\[
\delta_{tt}^{(i,j)} + \delta_{ssss}^{(i,j)} + c\Delta_t^{(i,j)} - \frac{\alpha}{2}\delta_{ss}^{(i,j)}T^{(i)} = \Lambda \cos \omega t
\]

(4.5)

for \(i = 2, \ldots, N\) and \(j = 3, \ldots, N - 1\), where \(\delta\) and \(\Delta\) denote the central and forward difference approximations, respectively, and

\[
T^{(i)} := \sum_{k=2}^{N+1} \frac{(\Delta_s^{(i,k)})^2 + (\Delta_s^{(i,k-1)})^2}{2N}
\]

(4.6)

Here, subscripts represent different derivative orders and superscripts identify the corresponding nodal points or, in the case of \(T^{(i)}\), a time slice. For example,

\[
\delta_{ssss}^{(i,j)} = \frac{v(i-2,j) - 4v(i-1,j) + 6v(i,j) - 4v(i+1,j) + v(i+2,j)}{h^4}
\]

(4.7)

where \(h = 1/N\) is the mesh size. In this notation, the spatial boundary conditions become

\[
v_{i,1} = v_{i,2} = v_{i,N} = v_{i,N+1} = 0, \ i = 1, \ldots, N + 1
\]

(4.8)

in the case of the clamped-clamped beam, and

\[
v_{i,1} = v_{i,3} - 2v_{i,2} = v_{i,N+1} = v_{i,N-1} - 2v_{i,N} = 0, \ i = 1, \ldots, N + 1
\]

(4.9)
in the case of the hinged-hinged beam. Periodicity is here imposed by requiring that

\begin{align*}
  v(1, j) &= v(N + 1, j) \quad (4.10) \\
  v(2, j) - v(1, j) &= v(N + 1, j) - v(N, j) \quad (4.11)
\end{align*}

for \( j = 3, \ldots, N - 1 \). For each set of spatial boundary conditions, the finite-difference discretization thus results in a nonlinear algebraic system of \((N + 1)^2\) equations in \((N + 1)^2\) unknowns.

The set of solutions to this system of equations may be parameterized by the excitation amplitude \( \Lambda \) (written below as a multiple of \( \Lambda_0 \)) and excitation frequency \( \omega \). An atlas covering the corresponding two-dimensional solution manifold, or any of the one-dimensional embedded manifolds obtained by single-parameter variations, may now be generated using the MATLAB-based computational continuation core COCO [58].

In order to locate an initial point on this solution manifold, we consider the solution to the linear Euler-Bernoulli beam, when subjected to small excitation amplitudes at low frequencies where the spatial shape of the response is dominated by the first bending mode of the beam. Under these conditions, an approximate steady-state solution may be written in the following separated form

\[ v(t, s) = A(t) \phi(s) \quad (4.12) \]

in terms of the first spatial eigenfunction of the corresponding Euler-Bernoulli beam \( \phi(s) \) and the time-periodic amplitude

\[ A(t) = a \cos(\omega t + \varphi) \quad (4.13) \]

For example, in the case of clamped-clamped boundary condition, we have

\[ \phi(s) = \cos \sqrt{\omega_1} s - \cosh \sqrt{\omega_1} s + \frac{\cosh \sqrt{\omega_1} - \cos \sqrt{\omega_1}}{\sin \sqrt{\omega_1} - \sinh \sqrt{\omega_1}} \left( \sin \sqrt{\omega_1} s - \sinh \sqrt{\omega_1} s \right) \quad (4.14) \]
where \( \omega_1 \approx 22.3733 \) is the lowest root of the characteristic equation

\[
\cosh \sqrt{\omega} \cos \sqrt{\omega} = 1 \tag{4.15}
\]

Substitution of Eqn. (4.12) into (4.1) with \( \alpha = 0 \), followed by multiplication by \( \phi(s) \) and integration over the spatial domain then yields an ordinary differential equation, to which the method of undetermined coefficients may be applied to obtain

\[
a = \frac{k_2 \Lambda}{\sqrt{\omega^4 + (c^2 - 2k_1) \omega^2 + k_1^2}} \tag{4.16}
\]

\[
\varphi = \arccos \left( \frac{k_1 - \omega^2}{\sqrt{\omega^4 + (c^2 - 2k_1) \omega^2 + k_1^2}} \right) \tag{4.17}
\]

where

\[
k_1 = \int_0^1 \phi(s) \phi_{ssss}(s) \, ds \tag{4.18}
\]

\[
k_2 = \int_0^1 \phi(s) \, ds \tag{4.19}
\]

We illustrate the application of the finite-difference discretization to the continuation of the periodic steady-state responses for the hinged-hinged beam for different orders of discretization in Fig. 4.3 (cf. Fig. 4.1). An example space-time reconstruction of a periodic response is shown in Fig. 4.4. As seen in Fig. 4.3, a large discretization order is necessary to get reasonable agreement between the predictions of the continuation method and results obtained using finite-element methods. Indeed, further increases in the discretization order result in the analysis reaching memory limits imposed by the software implementation. These prevent successful continuation in the case of larger excitation amplitudes, limiting the usefulness of this discretization.

4.2.3 Finite-element discretization

Following the finite-element paradigm and the Bubnov-Galerkin method [59], we seek functions \( v(s,t) \) (from a space of twice-differentiable trial functions with continuous first deriva-
Figure 4.3: Frequency response of the beam for $\Lambda = \Lambda_0$ obtained from the finite-difference method (dashed lines) when compared to the left-most curve from Fig. 4.1 (solid line).

tives relative to $s$ and that satisfy the relevant boundary conditions) for which the integral

$$
\int_{\Omega} \left( w_{tt} + w_{ss} v_{ss} + c w_t v_t - \frac{\alpha}{2} w T v_{ss} - w \Lambda \cos \omega t \right) ds = 0
$$

vanishes for all $t$ and for arbitrary functions $w(s)$ (from a space of twice-differentiable test functions with continuous first derivatives and that satisfy the same boundary conditions as the trial functions).

Let $M$ and $p$ be two positive integers such that $p - 1 > 2/M$. This inequality condition is used to ensure that the number of internal and external boundary conditions do not exceed the total number of unknowns in the equations. For test functions $w(s)$ in the form of continuously differentiable, piecewise-polynomial functions in space with arbitrary base-point values on a spatial mesh given by the uniform partition

$$
0 = s_1 < \cdots < s_{M+1} = 1
$$
Figure 4.4: Periodic solution for $\Lambda = \Lambda_0$ and $\omega = 22.9456$ obtained by applying continuation to the finite-difference discretization.

of the domain $\Omega$, and the uniform subpartition

$$0 = \xi_1 < \cdots < \xi_{p+1} = 1$$

(4.22)

of the $i$-th interval $\Omega_i$ in terms of the local variable

$$\xi := \frac{s - s_i}{s_{i+1} - s_i}$$

(4.23)

we seek a solution $v(s, t)$ to the weak formulation in Eq. (4.20) in the form of a continuously differentiable, piecewise-polynomial function in space with time-dependent, periodic, base-point values on the same spatial mesh. For $i = 1, \ldots, M$, we now write

$$v(s, t)|_{\Omega_i} = v^{(i)}(\xi, t) := \sum_{l=1}^{p+1} L_i(\xi) v^{(i)}(\xi_l, t) = L \cdot x^{(i)}(t)$$

(4.24)
and
\[ w(s)\big|_{\Omega_i} = w^{(i)}(\xi) := \sum_{l=1}^{p+1} L_l(\xi) w^{(i)}(\xi_l) = \mathcal{L} \cdot y^{(i)} \] (4.25)

where \(x^{(i)}\) and \(y^{(i)}\) are column matrices containing the base-point values associated with \(\Omega_i\), and \(\mathcal{L}\) is a row matrix consisting of the Lagrangean shape functions

\[ L_k(\xi) := \prod_{l=1,l\neq k}^{p+1} \frac{\xi - \xi_l}{\xi_k - \xi_l}, \quad k = 1, \ldots, p + 1 \] (4.26)

The internal boundary conditions

\[ v^{(i)}(1,t) - v^{(i+1)}(0,t) = v^{(i)}_\xi(1,t) - v^{(i+1)}_\xi(0,t) = 0 \] (4.27)

and

\[ w^{(i)}(1) - w^{(i+1)}(0) = w^{(i)}_\xi(1) - w^{(i+1)}_\xi(0) = 0 \] (4.28)

for \(i = 1, \ldots, M - 1\), then guarantee that \(v(s,t)\) and \(w(s)\) are continuously differentiable. Finally, we obtain the external boundary conditions

\[ v^{(1)}(0,t) = v^{(1)}_\xi(0,t) = v^{(M)}(1,t) = v^{(M)}_\xi(1,t) = 0 \] (4.29)

and

\[ w^{(1)}(0) = w^{(1)}_\xi(0) = w^{(M)}(1) = w^{(M)}_\xi(1) = 0 \] (4.30)

for the clamped-clamped beam, and similarly for the hinged-hinged beam.

Substitution of the piecewise-polynomial form of the trial and test functions into Eq. (4.20) and integration now yields an equation of the form

\[ y^T \cdot \eta(\ddot{x}, \dot{x}, x, t) = 0 \] (4.31)

where \(x\) and \(y\) collect the base-point values for \(v(s,t)\) and \(w(s)\), respectively, and \(\eta\) is linear in the components of \(\ddot{x}\). Notably, the elements of \(y\) are not independent, as they must satisfy the boundary conditions as well as the requirement on continuity of the test functions and
their first derivatives across element boundaries. These conditions form a linear system of constraint equations of the form

\[ C \cdot y = 0, \quad (4.32) \]

in terms of a constant \((2M + 2) \times M(p + 1)\) matrix \(C\) of maximal rank. There exists a \(M(p + 1) \times (Mp - M - 2)\) matrix \(P\) whose columns span the nullspace of \(C\), such that

\[ y = P \cdot \tilde{y} \quad (4.33) \]

for some column matrix \(\tilde{y}\) with independent components. It follows that

\[ P^T \cdot \eta(\ddot{x}, \dot{x}, x, t) = 0 \quad (4.34) \]

By the imposition of the boundary conditions as well as conditions of continuity of the trial functions and their first spatial derivatives across interval boundaries, it follows that

\[ C \cdot x = 0 \quad (4.35) \]

and, consequently, that

\[ x = P \cdot \tilde{x} \quad (4.36) \]

for some column matrix \(\tilde{x}\) with independent components. Substitution in Eq. (4.34) then yields a system of \(M(p - 1) - 2\) implicit, second-order differential equations in the collection of \(M(p - 1) - 2\) unknown components of \(\tilde{x}\). Notably, the matrix \(\ddot{x}\) appears linearly in the resulting equations with a nonsingular coefficient matrix.

Further discretization in time is now required in order to enable numerical parameter continuation. We rely here on a discretization i) of the time-dependent components of \(\tilde{x}\) using continuous, piecewise-polynomial functions of degree \(m\) parameterized by equidistributed base points across \(N\) uniform intervals, and ii) of a first-order form of the governing implicit differential equations by requiring that these be satisfied by the polynomial approximants at a set of collocation nodes in each of the \(N\) intervals, corresponding to the Gauss-Legendre quadrature nodes [56]. In the case of uniform harmonic excitation, we rely on an autonomous
representation of the forced system, through the introduction of a pair of additional state
variables with a unique limit cycle motion on the unit circle, as also described in [56]. Finally,
to ensure a unique phase of the periodic steady-state response, we rescale the time variable
by the period $T$, treat the latter as unknown, and impose the additional condition that
$v_t\left(\frac{1}{2}, 1\right) = 0$, i.e., that the displacement of the midpoint reach a local extremum with respect
to time at $t = 1$.

The set of solutions to the discretized system of equations may now be parameterized
by the excitation amplitude $\Lambda$ (written below as a multiple of $\Lambda_0$) and excitation frequency
$\omega$. An atlas covering the corresponding two-dimensional solution manifold, or any of the
one-dimensional embedded manifolds obtained by single-parameter variations, may now be
generated using coco.

4.3 A mixed formulation

4.3.1 Problem definition

In the weak form of the scalar integro-partial differential equation (4.1), the presence of a
second-order spatial derivative necessitates the use of continuously differentiable test and
trial functions. An alternative approach [56, 60, 61] is to rewrite the equation in the form

$$u_{2,t} + u_{5,s} + cu_2 - \frac{\alpha}{2} u_4 T = \Lambda \cos \omega t$$  \hspace{1cm} (4.37)

where $T = \int_{\Omega} u_3^2 \, ds$, and

$$u_1 = v$$ \hspace{1cm} (4.38)
$$u_2 = v_t$$ \hspace{1cm} (4.39)
$$u_3 = v_s$$ \hspace{1cm} (4.40)
$$u_4 = v_{ss}$$ \hspace{1cm} (4.41)
$$u_5 = v_{sss}$$ \hspace{1cm} (4.42)
and to consider the corresponding mixed weak form

\[ 0 = \int_\Omega (u_{1,t} - u_2) w_1 \, ds \]  
(4.43)

\[ 0 = \int_\Omega (u_{2,t} + u_{5,s} + cu_2 - \frac{\alpha}{2} u_4 T - \Lambda \cos \omega t) w_2 \, ds \]  
(4.44)

\[ 0 = \int_\Omega (u_{1,s} - u_3) w_3 \, ds \]  
(4.45)

\[ 0 = \int_\Omega (u_{3,s} - u_4) w_4 \, ds \]  
(4.46)

\[ 0 = \int_\Omega (u_{4,s} - u_5) w_5 \, ds \]  
(4.47)

for some suitably chosen set of test functions \( w_1, w_2, w_3, w_4, \) and \( w_5. \) In the corresponding finite-element discretization, trial functions in \( H^1 \) may be employed, as the highest order of differentiation is one. In addition, the mixed form also allows for lower-degree polynomial approximants, since differentiation reduces their degree by at most one. Finally, for both sets of boundary conditions on \( v, \) the corresponding boundary conditions on the fields \( u_i, \) \( i = 1, \ldots, 5 \) are of Dirichlet form. For example, in the case of the clamped-clamped beam, we find that \( u_1, u_2, \) and \( u_3 \) must all vanish on the boundaries.

We again let \( M \) denote a positive integer and consider the uniform partition

\[ 0 = s_1 < \cdots < s_{M+1} = 1. \]  
(4.48)

On the \( i \)-th interval \([s_i, s_{i+1}],\) we write

\[ u_k(s, t) = u_k^{(i)}(\xi, t) := \sum_{l=1}^{p+1} \mathcal{L}_l(\xi) u_k^{(i)}(\xi_l, t) \]  
(4.49)

\[ w_k(s) = w_k^{(i)}(\xi) := \sum_{l=1}^{p+1} \mathcal{L}_l(\xi) w_k^{(i)}(\xi_l) \]  
(4.50)

where \( \xi \) is given by the transformation (4.23), the shape functions \( \mathcal{L}_l \) are given in (4.26), and \( \{\xi_l\}_{l=1}^{p+1} \) is a uniform partition of the interval \([0, 1].\) We proceed by collecting the time-dependent base-point values \( u_k^{(i)}(\xi_l, t) \) and the constant base-point values \( w_k^{(i)}(\xi_l) \) in the
time-dependent $5M(p + 1) \times 1$ column matrix

$$x := \left( x_1^T \ x_2^T \ x_3^T \ x_4^T \ x_5^T \right)^T$$ \hspace{1cm} (4.51)

and the constant $5M(p + 1) \times 1$ column matrix

$$y := \left( y_1^T \ y_2^T \ y_3^T \ y_4^T \ y_5^T \right)^T,$$ \hspace{1cm} (4.52)

respectively, with the first $M(p + 1)$ elements $x_1$ and $y_1$ corresponding to the first field, the next $M(p + 1)$ elements $x_2$ and $y_2$ to the second field, and so on. As an example, the base-point values of the $k$-th trial function in the $i$-th element are obtained by the matrix product $T_k^{(i)} \cdot x$, where $T_k^{(i)}$ is a $(p + 1) \times 5M(p + 1)$ rectangular matrix consisting of zeros except for a $(p + 1) \times (p + 1)$ identity matrix after the $(M(k - 1) + (i - 1))(p + 1)$ first rows, and similarly for the corresponding test function. In particular,

$$u_k^{(i)}(\xi, t) = \mathcal{L}(\xi) \cdot T_k^{(i)} \cdot x, \quad w_k^{(i)}(\xi) = \mathcal{L}(\xi) \cdot T_k^{(i)} \cdot y$$ \hspace{1cm} (4.53)

where $\mathcal{L}$ again denotes the row matrix of Lagrangean shape functions.

Now let $\mathcal{C}_1$ denote a maximal rank $(M + 1) \times M(p + 1)$ matrix, such that

$$\mathcal{C}_1 \cdot x_k = 0$$ \hspace{1cm} (4.54)

corresponds to the imposition of vanishing boundary conditions and continuity across all elements of the $k$-th trial function. Similarly, let $\mathcal{C}_2$ denote a maximal rank $(M - 1) \times M(p + 1)$ matrix, such that

$$\mathcal{C}_2 \cdot x_k = 0$$ \hspace{1cm} (4.55)

corresponds only to the imposition of continuity across all elements of the $k$-th trial function. It follows that

$$C \cdot x = 0$$ \hspace{1cm} (4.56)
for the maximal rank \((5M+1) \times 5M(p+1)\) matrix

\[
C = C_{h-h} := \text{diag} \left( \tilde{C}_1, \tilde{C}_1, \tilde{C}_2, \tilde{C}_1, \tilde{C}_2 \right)
\] (4.57)

in the case of hinged-hinged boundary conditions and

\[
C = C_{c-c} := \text{diag} \left( \tilde{C}_1, \tilde{C}_1, \tilde{C}_2, \tilde{C}_1, \tilde{C}_2 \right)
\] (4.58)

in the case of clamped-clamped boundary conditions.

We may choose the columns of the \(M(p+1) \times (Mp-1)\) and \(M(p+1) \times (Mp+1)\) matrices \(\tilde{P}_1\) and \(\tilde{P}_2\), such that they span the nullspaces of \(\tilde{C}_1\) and \(\tilde{C}_2\), respectively, and such that the number of initial zeros increases with each successive column. In each case, it follows as a consequence that the boundary and continuity conditions are satisfied provided that

\[
x = P \cdot \tilde{x}
\] (4.59)

in terms of the \((5Mp-1) \times 1\) column matrix

\[
\tilde{x} := \begin{pmatrix} \tilde{x}_1^T & \tilde{x}_2^T & \tilde{x}_3^T & \tilde{x}_4^T & \tilde{x}_5^T \end{pmatrix}^T
\] (4.60)

and the rectangular matrix

\[
P = P_{h-h} := \text{diag} \left( \tilde{P}_1, \tilde{P}_1, \tilde{P}_2, \tilde{P}_1, \tilde{P}_2 \right)
\] (4.61)

in the case of hinged-hinged boundary conditions and

\[
P = P_{c-c} := \text{diag} \left( \tilde{P}_1, \tilde{P}_1, \tilde{P}_2, \tilde{P}_2 \right)
\] (4.62)

in the case of clamped-clamped boundary conditions. In each case, we note that the \((p + 1) \times (5Mp - 1)\) matrix \(T_k^{(i)} \cdot P\) consists of zeros except for a (possibly truncated) identity matrix after the \(-1 + (i-1)p\) first columns.

In lieu of a theoretical analysis identifying the appropriate boundary conditions for the
test functions \( w_1, w_2, w_3, w_4, \) and \( w_5, \) we consider below the cases where

i) \( w_1, w_2, \) and \( w_4 \) vanish at the boundaries for the hinged-hinged beam

ii) \( w_1, w_2, w_3 \)

iii) \( w_1, w_2, w_5, \)

respectively, vanish on the boundaries for the clamped-clamped beam. In each case

\[
y = P \cdot \tilde{y},
\]

where the entries of the \((5M - 1) \times 1\) column matrix

\[
\tilde{y} := \begin{pmatrix} \tilde{y}_1^T & \tilde{y}_2^T & \tilde{y}_3^T & \tilde{y}_4^T & \tilde{y}_5^T \end{pmatrix}^T
\]

are independent and \( P \) is block-diagonal with diagonal elements given by \( \tilde{P}_1 \) and \( \tilde{P}_2, \) as appropriate.

Let

\[
W_1 := \int_0^1 L(\xi) \cdot \frac{dL(\xi)}{d\xi} d\xi \quad (4.65)
\]

\[
W_2 := \int_0^1 L(\xi) \cdot L(\xi) d\xi \quad (4.66)
\]

and define the Kronecker products

\[
\tilde{W}_1 := I_M \otimes W_1 \quad (4.67)
\]

and

\[
\tilde{W}_2 := I_M \otimes W_2 \quad (4.68)
\]

and the matrix products

\[
A_{i,j} := \tilde{P}_i^T \cdot \tilde{W}_1 \cdot \tilde{P}_j \quad (4.69)
\]
\[ B_{i,j} := \tilde{P}^T_i \cdot \tilde{W}_2 \cdot \tilde{P}_j \] (4.70)

for \( i, j = 1, 2 \). Since Eqs. (4.45-4.47) must hold for all members of the space of test functions, it follows that

\[ 0 = M A_{2,1} \cdot \tilde{x}_1 - B_{2,2} \cdot \tilde{x}_3 \] (4.71)
\[ 0 = M A_{1,2} \cdot \tilde{x}_3 - B_{1,1} \cdot \tilde{x}_4 \] (4.72)
\[ 0 = M A_{2,1} \cdot \tilde{x}_4 - B_{2,2} \cdot \tilde{x}_5 \] (4.73)

in the case of hinged-hinged boundary conditions with \( w_1, w_2, \) and \( w_4 \) vanishing on the boundaries;

\[ 0 = M A_{1,1} \cdot \tilde{x}_1 - B_{1,1} \cdot \tilde{x}_3 \] (4.74)
\[ 0 = M A_{2,1} \cdot \tilde{x}_3 - B_{2,2} \cdot \tilde{x}_4 \] (4.75)
\[ 0 = M A_{2,2} \cdot \tilde{x}_4 - B_{2,2} \cdot \tilde{x}_5 \] (4.76)

in the case of clamped-clamped boundary conditions with \( w_1, w_2, \) and \( w_3 \) vanishing on the boundaries; and

\[ 0 = M A_{2,1} \cdot \tilde{x}_1 - B_{2,1} \cdot \tilde{x}_3 \] (4.77)
\[ 0 = M A_{2,1} \cdot \tilde{x}_3 - B_{2,2} \cdot \tilde{x}_4 \] (4.78)
\[ 0 = M A_{1,2} \cdot \tilde{x}_4 - B_{1,2} \cdot \tilde{x}_5 \] (4.79)

in the case of clamped-clamped boundary conditions with \( w_1, w_2, \) and \( w_5 \) vanishing on the boundaries.

From the Appendix A.2, we obtain the invertibility of \( B_{1,1} \) and \( B_{2,2} \). We may thus solve for \( \tilde{x}_3, \tilde{x}_4, \) and \( \tilde{x}_5 \) in terms of \( \tilde{x}_1 \) in the first two cases considered above. Substitution into the equations obtained from (4.43-4.44) then yields a system of ordinary differential equations in \( \tilde{x}_1 \) and \( \tilde{x}_2 \). Alternatively, if we retain the algebraic conditions on \( \tilde{x} \), the resultant system
of differential-algebraic equations is of index one.

In contrast, in the case of clamped-clamped boundary conditions with \( w_1, w_2, \) and \( w_5 \) vanishing on the boundaries, it is clearly not possible to eliminate \( \tilde{x}_3, \tilde{x}_4, \) and \( \tilde{x}_5 \) in terms of \( \tilde{x}_1 \), since the coefficient matrix \( B_{1,2} \) is rectangular with at least a two-dimensional nullspace. In this case, we must carry the algebraic conditions on \( \tilde{x} \) together with the equations obtained from (4.43-4.44), resulting in a system of differential-algebraic equations of index two.

Even in the case that \( \tilde{x}_3, \tilde{x}_4, \) and \( \tilde{x}_5 \) may be eliminated in terms of \( \tilde{x}_1 \) for clamped-clamped boundary conditions, a further complication arises. In particular, it follows from the analysis in the Appendix A.1 that the square matrix \( A_{1,1} \) is skewsymmetric and, consequently, singular when either \( M \) or \( p \) is even. In this case, the corresponding one-dimensional nullspace is spanned by the column matrix \( \tilde{v} \) given explicitly in the Appendix A.1. It follows that Eqs. (4.74-4.76) are invariant under arbitrary translations of \( \tilde{x}_1 \) along the vector \( \tilde{v} \). Furthermore, since the only remaining occurrence of \( u_1 \) in Eqs. (4.43-4.47) is in terms of its partial derivative with respect to time in Eq. (4.43), it follows that the reduced system of ordinary differential equations (or the corresponding index-one system of differential-algebraic equations) is invariant under time-independent translations of \( \tilde{x}_1 \) along \( \tilde{v} \).

In the next section, we investigate further the two cases of clamped-clamped boundary conditions, having established in the preceding analysis special circumstances that need to be accounted for during parameter continuation for either formulation.

### 4.3.2 Temporal ghost modes

We find it convenient to write the system of differential-algebraic equations obtained from the weak mixed form, after elimination of boundary and continuity conditions on the trial functions and time rescaling, in the form

\[
\begin{pmatrix}
B_{1,1} & 0 & 0 \\
0 & B_{1,1} & 0 \\
0 & 0 & 0
\end{pmatrix} \cdot \dot{\tilde{x}} = TR(\tilde{x})
\] (4.80)
where $T$ denotes the period of oscillation and we recall the invertibility of the constant 
$(Mp - 1) \times (Mp - 1)$ matrix $B_{1,1}$.

In order to enable continuation of steady-state, periodic trajectories of this system of 
differential-algebraic equations, we again approximate the unknown time-dependent base-
point values across a uniform partition

$$0 = t_1 < \cdots < t_{N+1} = 1$$

of $[0, 1]$ in $N$ intervals by continuous, periodic, piecewise-polynomial functions of degree $m$
for all components of $\tilde{x}$, and impose Eq. (4.80) at a set of $m$ collocation nodes in each time
interval corresponding to the Gauss-Legendre quadrature nodes, i.e., the roots of the $m$-th
degree shifted Legendre polynomial $Q_m(\eta)$ on $[0, 1]$ composed with the affine transformation

$$\eta = N (t - t_i)$$

for the corresponding $i$. At this point, assuming a known period of oscillation, it is straightforward to show that the continuation problem is, in fact, singular [62], since it supports a one-parameter family of shifts in the phase of the response. Our objective is now to modify the continuation problem to guarantee a nondegenerate formulation.

To this end, we again let the period of oscillation $T$ be unknown and append a phase
condition on the solution trajectory corresponding to a vanishing value of the piecewise-
polynomial approximant of $u_2$ at $s = 1/2$ and $t = 1$ (consistent, in a strong formulation with
the terminal end of the trajectory corresponding to a moment of extremal displacement of
the point at the beam’s middle). In the case of a system of ordinary differential equations,
it can be shown that the introduction of this condition generically resolves the singularity
observed in the previous paragraph. In the differential-algebraic context relevant to the
mixed formulation of the beam dynamics, however, this may not be the case (cf. [63]).

Consider, in particular, the case of the index-one system of differential-algebraic equations
and let $\chi_{\text{app}}(t)$ denote a continuous, periodic, piecewise-polynomial approximant on $[0, 1]$ of
any component of the vector of unknowns $\tilde{x}(t)$ whose derivative does not appear in the
differential-algebraic equations. Then, with all other components unchanged, \( \chi_{\text{app}}(t) + \delta \chi(t) \) satisfies the differential-algebraic equations at the collocation nodes, provided that \( \delta \chi(t) \) also vanishes at the collocation nodes. For \( \delta \chi(t) \) given by a continuous, piecewise-polynomial function of degree \( m \) on each of the \( N \) intervals, it follows that \( \delta \chi \) must equal a multiple of \( Q_m(N(t - t_i)) \) for each \( i \). By the symmetry properties of the Legendre polynomials, it follows that

\[
\delta \chi(0) = (-1)^{mN} \delta \chi(1)
\]  

and thus that the periodicity condition is satisfied by \( \delta \chi(t) \) provided that \( mN \) is even. We conclude that the continuation problem is singular if \( m \) or \( N \) is even (see Fig. 4.5).

In the case that \( m \) or \( N \) is even, it is clear that the time-discretization of the continuation problem must be modified in order to support nondegenerate continuation. In particular, we must ensure that a nontrivial, periodic, continuous, piecewise-polynomial function that vanishes on each of the collocation nodes fails to satisfy the continuation problem. We may achieve this, for example, by replacing the condition that the components of \( \tilde{x}_3, \tilde{x}_4, \) and \( \tilde{x}_5 \) be periodic with the condition that the algebraic equations be satisfied at \( t = 0 \). In the case that the differential-algebraic equations are index one, it follows that this condition is satisfied only if \( \delta \chi(0) = 0 \), i.e., for

\[
\delta \chi \equiv 0.
\]  

As an alternative, we may reformulate the discretization of the differential-algebraic equations using polynomials for \( \tilde{x}_3, \tilde{x}_4, \) and \( \tilde{x}_5 \) of degree \( m - 1 \), one degree lower than that used for \( \tilde{x}_1 \) and \( \tilde{x}_2 \), and omitting the corresponding continuity and periodicity conditions.

Neither of these approaches suffices to regularize the discretization of the index-two differential-algebraic system of equations obtained for the clamped-clamped boundary conditions with \( w_1, w_2, w_5 \) vanishing on the boundaries. In this case, the imposition of the algebraic equations at \( t = 0 \) in lieu of the periodicity condition does not imply a unique solution, since to each continuous, piecewise-polynomial solution \( \tilde{x}_{\text{app}}(t) \) to this collocation problem, there corresponds a two-parameter family of solutions

\[
\tilde{x}_{\text{app}}(t) + \left( \begin{array}{cccc} 0 & 0 & 0 & \delta \tilde{x}_5(t)^T \end{array} \right)^T
\]  

90
where the continuous, piecewise-polynomial function $\delta \tilde{x}_5(t)$ vanishes at the collocation nodes and equals an arbitrary vector in the nullspace of $B_{1,2}$ at $t = 0$.

Consider again, as an alternative, the discretization of the unknown functions $\tilde{x}_3(t)$, $\tilde{x}_4(t)$, and $\tilde{x}_5(t)$ in terms of piecewise-polynomial functions of degree $m - 1$ and parameterized by their values at the $m$ collocation nodes. Let $\delta \tilde{x}_5(t)$ now denote such a piecewise-polynomial function that lies in the nullspace of $B_{1,2}$ for all $t$, written in terms of a time-dependent coefficient multiplied by a nullvector of $B_{1,2}$. We proceed to show that for a suitable choice of this time-dependent coefficient, there exists a continuous, piecewise-polynomial function $\delta \tilde{x}_2(t)$ of degree $m$ on each interval, such that

$$\tilde{x}_{\text{app}}(t) + \begin{pmatrix} 0 & \delta \tilde{x}_2(t)^T & 0 & 0 & \delta \tilde{x}_5(t)^T \end{pmatrix}^T$$

satisfies the differential-algebraic equations at the collocation nodes and the periodicity condition in $\tilde{x}_1$ and $\tilde{x}_2$.

To this end, consider the first three terms in Eq. (4.44) and the corresponding finite-
element discretization

\[ B_{1,1} \cdot (\dot{\tilde{x}}_2 + c\tilde{x}_2) - A_{1,2} \cdot \tilde{x}_5 \] (4.87)

This expression is unchanged by the addition of \( \delta \tilde{x}_2(t) \) to \( \tilde{x}_2(t) \) and, simultaneously, of \( \delta \tilde{x}_5(t) \) to \( \tilde{x}_5(t) \), provided that

i) \( \delta \tilde{x}_2(t) \) vanishes at the collocation nodes and,

ii) the value of the derivative of \( \delta \tilde{x}_2(t) \) at the collocation nodes equals the value of \( B_{1,1}^{-1} \cdot A_{1,2} \cdot \delta \tilde{x}_5(t) \) at the collocation nodes.

We restrict attention to the \( i \)-th interval \([t_i, t_{i+1}]\) and satisfy both of these conditions by letting each component of \( \delta \tilde{x}_2(t) \) equal an appropriate multiple of the shifted Legendre polynomial \( Q_m(N(t - t_i)) \) and by choosing the time-dependent polynomial coefficient in \( \delta \tilde{x}_5 \) accordingly. We note that continuity of \( \delta \tilde{x}_2 \) across interval boundaries can be ensured by the appropriate selection of the nullvector of \( B_{1,2} \) in each interval. By the symmetry properties of the Legendre polynomials, the resultant function then satisfies the periodicity condition if and only if \( Nm \) is even.

We conclude from the above discussion that the collocation formulation in terms of Gauss-Legendre collocation nodes again results in a singular continuation problem when \( Nm \) is even, even with the use of reduced-order polynomials for \( \tilde{x}_3(t) \), \( \tilde{x}_4(t) \), and \( \tilde{x}_5(t) \). As an alternative, we consider instead the use of periodic, continuous, piecewise-polynomial functions of degree \( m \) for all components of \( \tilde{x} \), and the imposition of the differential-algebraic system of equations on each interval at points corresponding to the Radau collocation nodes \([64]\), i.e., the roots of the polynomial \( Q_m + Q_{m+1} \) on \([0, 1]\). By the asymmetry of these nodes on each interval, and the boundary values of \( Q_m + Q_{m+1} \) (0 and 2, respectively), it follows that no nonzero function that vanishes at the collocation nodes can also be continuous across interval boundaries. The degeneracy identified above is thus suitably eliminated.

4.3.3 Spatial equivariance

We return to the index-one differential-algebraic equations and recall the existence of a vector \( \tilde{v} \), such that arbitrary translations of the \( \tilde{x}_1 \) component of a solution along \( \tilde{v} \) parametrize a
continuous family of solutions. Equivalently, we note the existence of a vector

$$\hat{v} := \begin{pmatrix} \bar{v}^T & 0 & 0 & 0 \end{pmatrix}^T$$  \hspace{1cm} (4.88)

such that

$$R(\tilde{x} + k\hat{v}) = R(\tilde{x})$$  \hspace{1cm} (4.89)

for \(i = 1, \ldots, 5\) and arbitrary \(k\). To every solution \(\tilde{x}^*\) of these equations, there thus exists a one-parameter family of solutions obtained by applying the elements of the continuous group of translations along \(\hat{v}\) to \(\tilde{x}^*\) (see Fig. 4.6). It follows that the discretization of the spatiotemporal boundary-value problem results in a degenerate continuation problem in the case of clamped-clamped boundary conditions with \(w_1\), \(w_2\), and \(w_3\) vanishing on the boundaries. In our previous publication, this was one reason for restricting attention to the case of hinged-hinged boundary conditions, for which the singularity does not occur. As we show below, however, the singularity in the continuation problem related to the translational symmetry may be removed using either of two approaches.

In the first approach we append an additional unknown and an additional equation to the system of algebraic equations obtained from the time discretization of Eq. (4.80) to ensure a nonsingular Jacobian and a unique solution for each choice of continuation parameters. To this end, consider the differential-algebraic system of equations given by

$$\begin{pmatrix} B_{1,1} & 0 & 0 \\ 0 & B_{1,1} & 0 \\ 0 & 0 & 0 \end{pmatrix} \cdot \dot{\tilde{x}} = TR(\tilde{x}) + \beta \hat{v}$$  \hspace{1cm} (4.90)

for some yet-to-be-determined scalar \(\beta\). In this case, periodic solutions of the original system may be obtained only for \(\beta = 0\), whereas nonzero values result in a constant drift along \(\hat{v}\). Suppose that

$$\tilde{n}^T \cdot \tilde{v} \neq 0$$  \hspace{1cm} (4.91)
for some constant column matrix \( n \) and let

\[
\hat{n} = \left( \hat{n}^T 0 0 0 \right)^T
\]  

(4.92)

We obtain a nonsingular continuation problem by appending the condition that

\[
\hat{n}^T \cdot \tilde{x}(0) = d
\]  

(4.93)

for some scalar \( d \) to the time discretization of Eq. (4.90). In this case, continuation produces a unique family of periodic solutions parameterized by system parameters, for which \( \beta = 0 \) and, without loss of generality,

\[
\hat{n}^T \cdot \tilde{x}(0) = d,
\]  

(4.94)

i.e., for which \( \tilde{x}(0) \) lies on a hyperplane with normal vector \( \hat{n} \) through the point \( d\hat{n}/\|\hat{n}\|^2 \) (as suggested in Fig. 4.6). The Jacobian of the discretized boundary-value problem corresponding to Eq. (4.90) consists of the Jacobian of the discretization of the original boundary-value problem Eq. (4.80) bordered by an additional column and row that ensure invertibility.

As an alternative to the augmented approach, a reduction in dimensionality may be obtained by considering the non-invertible variable transformation

\[
\hat{x} := \tilde{x} - \frac{\hat{n}^T \cdot \tilde{x} - d}{\hat{n}^T \cdot \hat{v}} \hat{v} = \tilde{x} - \frac{\hat{n}^T \cdot \tilde{x}_1 - d}{\hat{n}^T \cdot \hat{v}} \hat{v}
\]  

(4.95)

as inspired by principles of symmetry reduction (e.g., [65]) and the observation that

\[
R(\tilde{x}) = R(\hat{x})
\]  

(4.96)

for \( i = 1, \ldots, 5 \). It follows immediately that

\[
\hat{x}_i = \tilde{x}_i
\]  

(4.97)

for \( i = 2, \ldots, 5 \), and that

\[
\hat{n}^T \cdot \hat{x} = d
\]  

(4.98)
Figure 4.6: Equivariance results in a symmetry group orbit of periodic solutions shifted along $\hat{v}$. A unique solution is obtained by appending a boundary condition at $t = 0$.

i.e., that $\hat{x}$, as obtained from Eq. (4.95), parametrizes a hyperplane with normal vector $n$ through the point $d\hat{n}/\|\hat{n}\|^2$. Since $\hat{n} \neq 0$, we can eliminate one of the components of $\hat{x}_1$ by using Eq. (4.98). A reduced continuation problem now results from differentiation of the remaining components of Eq. (4.95) with respect to time and suitable discretization. As the rate of change with respect to time of the coefficient of $\hat{v}$ in Eq. (4.95) depends only on $\hat{x}$, it follows that the original solution may be reconstructed by integration and inversion of Eq. (4.95).
4.4 Numerical results

While the continuation paradigm was successfully applied to the mixed weak form of the governing partial differential equations in [56] in the case of hinged-hinged boundary conditions, preliminary observations were made therein regarding singularities in the continuation formulation for the case of clamped-clamped boundary conditions. The discussion in the previous sections has sought to identify and resolve these singularities, in order to allow for a comprehensive analysis independently of boundary conditions.

We proceed to investigate numerically the different discretizations in space and time proposed above and, in the case of uniform harmonic excitation, to show the results of parameter continuation obtained with COCO [5] for nondimensional excitation amplitudes equal to multiples of $\Lambda_0 = 0.075$, and, unless otherwise noted, with $\alpha = 1.2 \times 10^4$ and $\zeta = 0.02$.

4.4.1 An equilibrium problem

We calibrate our expectations of the finite-element discretization schemes discussed previously, by first considering their application to the analysis of time-independent solutions of Eq. (4.1), or the equivalent mixed strong form, in the absence of damping and for the case that $f$ is a constant, not-necessarily-uniform load and $\alpha = 0$. As such solutions may be obtained explicitly by integration, they offer an initial study of the rate of convergence of the different schemes with respect to the spatial discretization orders $M$ and $p$ (cf. [66]).

It follows from the analysis in Appendix A that the case of clamped-clamped boundary conditions with $w_1$, $w_2$, and $w_3$ vanishing on the boundary yields a noninvertible coefficient matrix with an image space that, generically, fails to contain the inhomogeneity of the linear finite-element discretization. This discretization scheme cannot be used to find equilibrium solutions of the clamped-clamped beam.

In contrast, as shown in Appendix A, each of the other two mixed form discretizations yield invertible coefficient matrices. Let the corresponding discretization errors $\varepsilon_{\text{mixed},h}$ and $\varepsilon_{\text{mixed},c}$ be given by the $L_2$ norm of the difference between the exact displacement and that computed by each discretization, and let $\varepsilon_{\text{scalar},h}$ and $\varepsilon_{\text{scalar},c}$ denote the equivalent error for
<table>
<thead>
<tr>
<th>error</th>
<th>$p = 3$</th>
<th>$p = 4$</th>
<th>$p = 5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon_{\text{scalar},h}$</td>
<td>4.2</td>
<td>5.1</td>
<td>6.1</td>
</tr>
<tr>
<td>$\varepsilon_{\text{scalar},c}$</td>
<td>4.0</td>
<td>5.0</td>
<td>6.0</td>
</tr>
<tr>
<td>$\varepsilon_{\text{mixed},h}$ and $\varepsilon_{\text{mixed},c}$</td>
<td>4.2</td>
<td>4.0</td>
<td>6.2</td>
</tr>
</tbody>
</table>

Table 4.1: Values of the negative slope to two significant digits obtained by fitting a straight line to log-log pairs of values of the order $M$ and the corresponding errors $\varepsilon$ for each of the four discretization methods applicable to the case of time-independent solutions to the undamped, linear beam problem with constant load. the corresponding $R^2$-values equal 0.999 to three decimals.

the finite-element discretization of Eq. (4.20). Then, with

$$f(s) = -8\pi^4 \cos 2\pi s,$$  \hspace{1cm} (4.99)

numerical experiments with $p = 3$, 4, and 5, and regression using data for $M = 4$, 8, 16, and 32, yield the convergence rates shown in Table 4.1, i.e., the exponents when fitting a straight line to log-log pairs of discretization orders $M$ and corresponding errors $\varepsilon$.

4.4.2 Linear response

We next explore the combination of spatial and temporal discretization in the analysis of periodic steady-state responses to uniform harmonic excitation for the linear case with $\alpha = 0$. Here, the solution may be obtained as an explicit time-dependent expansion in the corresponding mutually orthogonal spatial eigenfunctions given by

$$\phi(s) = \cos \sqrt{\lambda}s - \cosh \sqrt{\lambda}s$$

$$+ \frac{\cosh \sqrt{\lambda} - \cos \sqrt{\lambda}}{\sin \sqrt{\lambda} - \sinh \sqrt{\lambda}} \left( \sin \sqrt{\lambda}s - \sinh \sqrt{\lambda}s \right)$$  \hspace{1cm} (4.100)

where

$$\cosh \sqrt{\lambda} \cos \sqrt{\lambda} = 1$$  \hspace{1cm} (4.101)
in the case of clamped-clamped boundary conditions, and

$$\phi(s) = \sqrt{2} \sin \sqrt{\lambda} s$$  \hspace{1cm} (4.102)

where $\sin \sqrt{\lambda} = 0$, in the case of hinged-hinged boundary conditions. Substitution of the term-by-term ansatz

$$v(t, s) = a \cos (\omega t + \varphi) \phi(s)$$  \hspace{1cm} (4.103)

into Eq. (4.1) with $\alpha = 0$, followed by multiplication by $\phi(s)$ and integration over the spatial domain then yields an ordinary differential equation, to which the method of undetermined coefficients may be applied to obtain

$$a = \frac{k_2 \Lambda}{\sqrt{\omega^4 + (c^2 - 2k_1) \omega^2 + k_1^2}}$$  \hspace{1cm} (4.104)

$$\varphi = \arccos \left( \frac{k_1 - \omega^2}{\sqrt{\omega^4 + (c^2 - 2k_1) \omega^2 + k_1^2}} \right)$$  \hspace{1cm} (4.105)

where

$$k_1 = \int_0^1 \phi(s) \phi_{sss}(s) \, ds, \quad k_2 = \int_0^1 \phi(s) \, ds$$  \hspace{1cm} (4.106)

We proceed to consider the case that the excitation frequency $\omega$ equals the lowest root $\lambda_1 \approx 22.3733$ of the characteristic equation for the case of clamped-clamped boundary conditions, for which we expect the response to be dominated by the first spatial eigenmode, and let $\Lambda = 5\Lambda_0$. Throughout this section, we rely on the projection onto this first eigenmode as an initial solution guess for the root-finding algorithms implemented in coco. We consider, at first, the discretization error defined as the $L_2$ norm of the difference between the exact solution (obtained from a 20-term truncation of the corresponding modal expansion) at an instance of maximum displacement (vanishing velocity) of the mid-point of the beam and that obtained at $t = 1$ using the corresponding discretization scheme for the case of the scalar beam model as well as the mixed formulation with $w_1$, $w_2$, and $w_5$ vanishing on the boundaries. Numerical experiments with $N = 12$, $m = 5$, $p = 3, 4, \text{ and } 5$, and regression using data for $M = 4, \ldots, 8$, yield the convergence rates shown in Table 4.2.
Table 4.2: Values of the negative slope to two significant digits obtained by fitting a straight line to log-log pairs of values of the order $M$ and the corresponding errors $\varepsilon$ for each of the two discretization methods applicable to the case of periodic solutions to the linear beam problem with harmonic excitation, for which the original continuation problem is nonsingular. the corresponding $R^2$-values equal 0.99 to two decimals.

<table>
<thead>
<tr>
<th>error</th>
<th>$p = 3$</th>
<th>$p = 4$</th>
<th>$p = 5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon_{\text{scalar,c}}$</td>
<td>4.7</td>
<td>5.0</td>
<td>5.9</td>
</tr>
<tr>
<td>$\varepsilon_{\text{mixed,c}}$</td>
<td>4.3</td>
<td>4.1</td>
<td>6.0</td>
</tr>
</tbody>
</table>

In the case of the mixed formulation with $w_1$, $w_2$, and $w_3$ vanishing on the boundaries, we choose to rely on the method described in conjunction with Eq. (4.90) with $N = 12$ and $m = 5$ and the additional (somewhat arbitrary, but reasonable) condition that the slope of the interpolant for $u_1$ vanish at $t = s = 0$. Table 4.3 shows the corresponding convergence rates for $p = 3$, 4, and 5, using data obtained for $M = 4$, 8, 16, and 32. Here, the discretization error $\varepsilon_1$ is again the $L_2$ norm of the difference between the exact solution (obtained from a 20-term truncation of the corresponding modal expansion) at an instance of maximum displacement of the mid-point of the beam and that obtained at $t = 1$ using the discretization scheme. As an alternative, we first compute the (peak-to-peak) difference between the spatial beam shapes at moments of maximum and minimum displacement (vanishing velocity) of the midpoint and then define the discretization error $\varepsilon_2$ by the $L_2$ norm of the difference between the functions obtained using discretization and from the exact solution, respectively. Finally, the discretization error $\varepsilon_3$ is the $L_2$ norm of the difference between the spatial interpolant for $u_2$ obtained at a moment of maximum value of this interpolant at $s = 1/2$ using the discretization scheme and the corresponding exact solution.

Finally, we recall the need to rely on the Radau points in the time-discretization of the index-two differential-algebraic equations obtained for the case of $w_1$, $w_2$, and $w_5$ vanishing on the boundaries. To investigate the implications to the rate of convergence with respect to the discretization order $N$, numerical experiments were performed on the mixed discretizations with $m = 3$, $p = 5$, and with $M = 32$ for the equivariant, index-one formulation (with Gauss collocation nodes) and $M = 8$ for the index-two formulation (with Radau collocation nodes).
Table 4.3: Values of the negative slope to two significant digits obtained by fitting a straight line to log-log pairs of values of the order $M$ and the corresponding errors $\varepsilon$ for the discretization method applicable to the case of periodic solutions to the linear beam problem with harmonic excitation, for which the original continuation problem is equivariant. The corresponding $R^2$-values equal 0.99 to two decimals.

<table>
<thead>
<tr>
<th>error</th>
<th>$p = 3$</th>
<th>$p = 4$</th>
<th>$p = 5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon_1$</td>
<td>4.0</td>
<td>3.8</td>
<td>5.1</td>
</tr>
<tr>
<td>$\varepsilon_2$ and $\varepsilon_3$</td>
<td>4.3</td>
<td>4.1</td>
<td>6.0</td>
</tr>
</tbody>
</table>

Here, the discretization error was defined as the difference between the displacement of the mid-point of the beam at a moment of maximum displacement (coinciding with a temporal mesh point, for which we expect superconvergence) and the exact solution. In this case, the convergence rates obtained by regression applied to data obtained for $N = 4, 8, 16,$ and $32$ were 5.6 for the equivariant, index-one formulation and 4.9 for the index-two formulation, consistent with the expected loss of convergence rate associated with the Radau collocation nodes. In both cases, the corresponding $R^2$ values equal 0.99 to two decimals.

4.4.3 Frequency-response curves

We conclude the numerical study by reporting on frequency-response curves and associated data obtained using numerical continuation along the solution manifold of periodic orbits under variations in the excitation frequency for $\Lambda = 10\Lambda_0$ and for nonzero damping and nonlinearity. We seed the root-finding algorithm with the periodic response of the linear, harmonically excited beam, as in the previous section.

As in the case of the hinged-hinged beam, the frequency-response curve shown in Fig. 4.7 for the clamped-clamped beam exhibits the hardening behavior induced by the nonlinearity and the existence of two fold bifurcations delimiting a region of bistability. As shown in the inset, the discretization scheme obtained using the mixed formulation with $w_1$, $w_2$, and $w_5$ vanishing on the boundaries and $N = 12$, $m = 5$, and $p = 4$ converges rapidly as $M$ increases. We characterize the rates of convergence of the two mixed formulations with respect to the discretization order $M$ by considering the application of Richardson extrapolation to the
Figure 4.7: Frequency-response curves obtained using the index-2 discretization (dashed) for $M = 2, 4,$ and $6$ and $p = 4$ compared to that obtained using the discretized scalar model (solid) for $M = 8$ and $p = 4$.

excitation frequency and peak-to-peak response amplitude of the mid-point of the beam at the upper fold bifurcation point. Here, using data obtained for $M = 4, 6,$ and $8$, the estimated discretization errors decay exponentially to the extrapolated values at rates of $7.9$ and $3.7$ for the index-two boundary-value problem and rates of $3.4$ and $3.5$ for the index-one boundary-value problem. In each case, the $R^2$ values are $1.00$ to two decimals.

4.5 Conclusions

The above enumeration of discretization methods in space and time, and associated singularities, leaves one with some uncertainty as to the most suitable choice for the simple
application at hand. Although convergence is observed numerically, we are not aware of theoretical results that guarantee such convergence for either of the mixed formulations (but see [67, 68] for a general treatment of convergence and stability in mixed finite-element methods and, e.g., [69, 70] for the use of mixed formulations in the analysis of beam deformations). Although not entirely unexpected, one even notes critical values for the model parameters, corresponding to the static equilibrium problem, where discretization may yield an unrecoverably singular boundary-value problem, even as convergence is observed in the case of steady-state responses for harmonic excitation.

There is no clear \textit{a priori} distinction between the two choices of boundary conditions on the test functions in the mixed formulation of the clamped-clamped beam. Indeed, one can imagine many other choices for boundary conditions, as well as alternative mixed formulations for which the auxiliary fields $\tilde{u}_3$, $\tilde{u}_4$, and $\tilde{u}_5$ are redefined in such a way as to eliminate the equivariance found above, yielding nonsingular continuation problems. On the other hand, recalling the variational origin of the weak formulation, one may be led to associate the test functions to virtual variations in the field variables in such a way as to ensure a consistent physical meaning across each of the five equations of the mixed weak form. Such a treatment lends credence to the imposition of zero boundary values for $w_1$, $w_2$, and $w_4$ in the case of hinged-hinged boundary conditions, and for $w_1$, $w_2$, and $w_5$ in the case of clamped-clamped boundary conditions.

Finally, we reflect on the implications of the equivariance in the mixed formulation with $w_1$, $w_2$, and $w_3$ vanishing on the boundaries to system design. As the original formulation fails to identify uniquely a member of the one-parameter family of periodic orbits, one should not assume that a unique solution obtained through either of the two methods for breaking the equivariance through some choice of $\hat{n}$ and $d$ is in any way superior to a solution obtained through a different choice of $\hat{n}$ and $d$. The choice corresponding to vanishing slope for the interpolant of $u_1$ at $t = s = 0$ is consistent with the strong formulation. Nevertheless, in the absence of a convergence proof specifically for this condition, we have restricted most of the numerical analysis to quantities that are invariant under the group symmetry, e.g., the critical excitation frequency and peak-to-peak amplitude of the mid-point of the beam at the fold bifurcation point.
5.1 Introduction

Axially\(^1\) restrained beams [57] have widespread applications in nano/micro electromechanical devices, from sensing [71] to detection [72] and communication systems [73]. These applications rely on high accuracy predictive models that can be used in robust design of such electromechanical devices. The complexity of the design methodology is even further increased at the nanoscale when the high noise-to-signal ratios in the measurements demand for reliable designs based on models that take into account all the phenomenological aspects that govern the dynamics of these devices. In particular, the complication arises from the intrinsic different nonlinear signatures of beams [74]. The first group corresponds to nonlinearities of hardening type within the context of axially restrained nonlinear planar beams. The hardening effect is due to geometric nonlinearities in which the beam’s axis is stretched and the ensuing tension couples with the curvature and becomes pronounced at high amplitudes of deformation. This effect has been studied extensively [75, 76], mostly inspired by the work of Mettler [77]. The second group of nonlinearity has a softening effect and is mainly associated with inertia effects which, in turn, are particularly affected by the mass distribution in the specific geometric design of the beam.

In addition to the intrinsic sources of nonlinearities in beams, the particular physics used in the design of electromechanical devices can also contribute to this nonlinear behavior. This is the case, for example, with the electrostatic actuation which depends in a strongly nonlinear fashion from the beam deflection. Both types of nonlinearities in beams have been

---

\(^{1}\)The material in this chapter is collaborative efforts with Prof. Harry Dankowicz and Prof. Walter Lacarbonara and based on a journal manuscript being prepared for submission.
utilized, favorably, in designing sensor [78, 79, 80], switch [81], amplifier [82] applications in which the nonlinear effects play a crucial role (for a review see [83]).

Despite the successful approach of exploiting the nonlinearities at the design stage of electromechanical devices, nonlinear behaviors are, in general, considered as undesirable due to added uncertainty in the system performance and, hence, in some instances their avoidance is carefully sought [84].

Within the context of axially restrained beams, the hardening effect is usually dominant which leads to a natural restriction of the deflection amplitudes to ensure the operation within the linear range. However, in many practical situations, including the applications of nano-resonators, the high levels of noise-to-signal ratios, further restrict this linear range (also known as the dynamic range). Consequently, designs that increase the dynamic range of nano-resonators are highly desirable. The idea of tuning the nonlinearities in microelectromechanical devices has been investigated in the literature [85, 86, 87]. More recently, nonlinear tuning has been used experimentally to increase the dynamic range of electrostatically driven resonators [88, 89, 90].

In all these applications, the softening effect was generated by the specific physics used in the design of the device, where the intrinsic softening effects in the specific design of the beam were neglected. It is the goal of this study to explore the effect of the specific geometry of the beam, in particular a bi-layer beam, on the magnitude of the softening effect and its subsequent influence on the dynamic range. To this end, in section 5.2, we first develop an accurate model of the beam dynamics. In section 5.3, we employ a perturbation method to obtain a low-order approximation of the dynamic response and use it to validate the results of our numerical scheme. The numerical method is based on the application of the Galerkin method to a finite-element discretization based on the mixed strong formulation of the original partial-differential equations of the beam. This method, together with a formulation that captures the dynamic range of the beam as a constraint, is used to explore the dynamic range of the beam for a wide variation of several design parameters.

A preliminary study on the modeling as well as the perturbation analysis has been published in Ref. [91] and the related materials in this chapter are an extended version of that work.
5.2 Model derivation

In this study, we restrict our attention to planar motions [92]. Let

\[ r^0(s) = s e_1 \]  

be the position vector describing the configuration of the reference line of a stress-free straight beam parameterized by the horizontal coordinate \( s \). Denote by

\[ r(s, t) = s e_1 + u(s, t) e_1 + v(s, t) e_2 \]  

the position of the corresponding material point in the deformed configuration. Furthermore, as shown in Fig. 5.1, introduce the rotated reference triad to represent the current orientation of the material cross section at \( s \) as

\[ b_1(s, t) = \cos \theta(s, t) e_1 + \sin \theta(s, t) e_2 \]  
\[ b_2(s, t) = -\sin \theta(s, t) e_1 + \cos \theta(s, t) e_2 \]

where \( \theta(s, t) \) thus represents the counterclockwise rotation angle of the corresponding beam cross section. Then,

\[ r_s = (1 + u_s) e_1 + v_s e_2 \overset{\text{def}}{=} \nu b_1 + \eta b_2 \]

implies that

\[ \nu = (1 + u_s) \cos \theta + v_s \sin \theta, \]  
\[ \eta = -(1 + u_s) \sin \theta + v_s \cos \theta. \]

Let

\[ \mu \overset{\text{def}}{=} \theta_s. \]
Figure 5.1: Schematics of the beam (top) and its cross-section (bottom). The dashed lines show the center surface of each layer.

Here $\nu - 1$, $\eta$, and $\mu$ denote the beam elongation, shear strain and flexural curvature, respectively. Moreover, the angular velocity vector is defined as

$$\omega = \dot{\theta} b_3$$ \hspace{1cm} (5.9)

The balance of the linear and angular momentum in local form implies the equations of motion [92]

$$\mathbf{n}_s + \mathbf{f} = \rho A \ddot{\mathbf{r}} + \mathbf{\omega} \times \dot{\mathbf{r}} + \mathbf{\omega} \times \mathbf{\omega} \times \mathbf{g}$$ \hspace{1cm} (5.10)

$$\mathbf{m}_s + \mathbf{r}_s \times \mathbf{n} + \mathbf{c} = \rho J \ddot{\mathbf{\omega}} + \mathbf{\omega} \times \dot{\mathbf{r}} + \mathbf{\omega} \times (\rho J \cdot \mathbf{\omega})$$ \hspace{1cm} (5.11)
where

\[ \mathbf{n} = N\mathbf{b}_1 + Q\mathbf{b}_2 \]  

(5.12)

is the contact force vector comprising the tension \( N \) and the shear force \( Q \), \( \mathbf{m} = \mathbf{M}\mathbf{b}_3 \) is the bending moment, \( \mathbf{f} \) is the distributed force per unit reference length, \( \mathbf{c} = \mathbf{c}\mathbf{b}_3 \) is the distributed couple per unit reference length, \( \varrho \) is the mass density of the beam, \( A \) is the cross-sectional area, \( J \) is the cross-sectional second area moment of inertia, and \( \varrho\mathbf{i} \) is the vector of first moment of inertia, i.e.,

\[ \varrho\mathbf{i} = \varrho\mathbf{I}\mathbf{b}_2 \]  

(5.13)

where \( I \) is the cross-sectional first area moment of inertia. The component form of the equations of motion becomes

\[
\begin{align*}
N_s - \mu Q + f_1 &= \varrho A (\ddot{u} \cos \theta + \dot{v} \sin \theta) + \varrho I \dot{\theta}, \\
Q_s + \mu N + f_2 &= \varrho A (\ddot{v} \cos \theta - \dot{u} \sin \theta) + \varrho I \dot{\theta}^2, \\
M_s + \nu Q - \eta N + c &= \varrho I \ddot{\theta} + \varrho I (\ddot{u} \cos \theta + \dot{v} \sin \theta).
\end{align*}
\]  

(5.14-5.16)

where \( f_1 \) and \( f_2 \) are the components of \( \mathbf{f} \) along the \( \mathbf{b}_1 \) and \( \mathbf{b}_2 \) directions, respectively. By imposing the Kirchhoff constraint \( \eta = 0 \) (i.e., shear undeformability) the following relationships are obtained

\[
\begin{align*}
\nu &= \sqrt{(1 + u_s)^2 + v_s^2}, \\
\cos \theta &= \frac{1 + u_s}{\nu}, \quad \sin \theta = \frac{v_s}{\nu}
\end{align*}
\]  

(5.17-5.18)

such that

\[
\theta = \arctan \frac{v_s}{1 + u_s}
\]  

(5.19)

and

\[
\mu = \frac{(1 + u_s) v_{ss} - v_s u_{ss}}{\nu^2}
\]  

(5.20)
5.2.1 Multi-layered beams

The equations of motion of a multi-layered beam are obtained in the form [57]

\[
N_s - \mu Q + f_1 = \left( \sum_i \rho_i A_i \right) (\ddot{u} \cos \theta + \ddot{v} \sin \theta) + \left( \sum_i \rho_i I_i \right) \dot{\theta} \tag{5.21}
\]

\[
Q_s + \mu N + f_2 = \left( \sum_i \rho_i A_i \right) (\ddot{v} \cos \theta - \ddot{u} \sin \theta) + \left( \sum_i \rho_i I_i \right) \dot{\theta}^2 \tag{5.22}
\]

\[
M_s + \nu Q + c = \left( \sum_i \rho_i J_i \right) \ddot{\theta} + \left( \sum_i \rho_i I_i \right) (\ddot{u} \cos \theta + \ddot{v} \sin \theta) \tag{5.23}
\]

where \(\rho A_i\), and \(\rho J_i\) are the mass per unit length, and the mass moment of inertia, respectively, for the \(i\)-th layer with \(i = \{1, \cdots, n_l\}\) and \(n_l\) is the number of layers in the beam. In this study, we restrict our attention to the analysis of bi-layer beam, i.e., \(n_l = 2\). Upon rescaling time by \(l^2 / \sqrt{\rho_1 A_1 / E_1 J_1}\), forces by \(E_1 J_1 / l^2\), and lengths by \(l\), we obtain

\[
N_s - \mu Q + f_1 = \left( 1 + \frac{\rho_2 A_2}{\rho_1 A_1} \right) (\ddot{u} \cos \theta + \ddot{v} \sin \theta) + \frac{\rho_2}{\rho_1} \frac{I_2}{A_1 l} \dot{\theta} \tag{5.24}
\]

\[
Q_s + \mu N + f_2 = \left( 1 + \frac{\rho_2 A_2}{\rho_1 A_1} \right) (\ddot{v} \cos \theta - \ddot{u} \sin \theta) + \frac{\rho_2}{\rho_1} \frac{I_2}{A_1 l} \dot{\theta}^2 \tag{5.25}
\]

\[
M_s + \nu Q + c = \lambda \ddot{\theta} + \frac{\rho_2}{\rho_1} \frac{I_2}{A_1 l} (\ddot{u} \cos \theta + \ddot{v} \sin \theta) \tag{5.26}
\]

in which

\[
\lambda = \frac{1}{\alpha^2} \left( 1 + \frac{\rho_2 A_2}{\rho_1 A_1} \left( \frac{h_2}{h_1} \right)^2 + 3 \left( 1 + \frac{h_2}{h_1} \right)^2 \right) \tag{5.27}
\]

where

\[
\alpha = l \sqrt{A_1 / J_1}. \tag{5.28}
\]

Here, we have assumed that the first layer is the beam substrate with non-zero dimensions throughout the beam span. Note that the horizontal axis in Fig. 5.1 passes through the center line of the substrate such that the corresponding moment of inertia equals zero. We also have

\[
I_2 = -A_2 (h_1 + h_2) / 2. \tag{5.29}
\]
By considering the strain-displacement relationship for the elongation

$$\epsilon(z) = \nu - 1 + z \mu$$

(5.30)

and the linearly elastic constitutive

$$\sigma_i = E_i \epsilon(z)$$

(5.31)

for the longitudinal stress in the $i$-th layer, we then obtain for the tension [93]

$$N = \int_D E(\nu - 1 + z\mu)dA = \sum_i (E_i A_i) (\nu - 1) + \mu \int_D zE(z) \ dA,$$

$$= \sum_i (E_i A_i) (\nu - 1) + \mu E_2 I_2,$$

(5.32)

in which $E_i$ and $b_i$ are Young’s modulus and the width of the $i$-th layer, respectively, and $D$ is the cross-sectional domain. Rescaling of forces yields

$$N = K_{11} (\nu - 1) + K_{12} \mu$$

(5.33)

where

$$K_{11} = 12 \left(\frac{h_1}{l}\right)^{-2} \left(1 + \frac{E_2}{E_1} \frac{A_2}{A_1}\right)$$

(5.34)

$$K_{12} = -6 \left(\frac{h_1}{l}\right)^{-1} \frac{E_2}{E_1} \frac{A_2}{A_1} \left(1 + \frac{h_2}{h_1}\right)$$

(5.35)

Similarly, we obtain

$$M = \int_D zE(\nu - 1 + z\mu)dA = (\nu - 1) E_2 I_2 + \mu \sum_i (E_i I_i)$$

(5.36)

After rescaling

$$M = K_{12} (\nu - 1) + K_{22} \mu$$

(5.37)
in which
\[ K_{22} = 1 + \frac{E_2 A_2}{E_1 A_1} \left( \frac{h_2}{h_1} \right)^2 + 3 \left( 1 + \frac{h_2}{h_1} \right)^2 \] (5.38)

The coupled equations of motion are then obtained as
\[ K_{11}\nu_s + K_{12}\mu_s - \mu\Pi + \bar{I}\ddot{\theta} + f_1 = \bar{m} (\ddot{u} \cos \theta + \ddot{v} \sin \theta) \] (5.39)
\[ \Pi_s + K_{11}\mu (\nu - 1) + K_{12}\mu^2 + \bar{I}\dot{\theta}^2 + f_2 = \bar{m} (\dot{v} \cos \theta - \dot{u} \sin \theta) \] (5.40)

with
\[ \bar{m} = 1 + \frac{E_2 A_2}{E_1 A_1}, \] (5.41)
\[ \bar{I} = \frac{1}{2} \frac{E_2 A_2 h_1}{E_1 A_1} \left( 1 + \frac{h_2}{h_1} \right), \] (5.42)

and
\[ \Pi = \frac{\lambda\ddot{\theta} - \bar{I} (\ddot{u} \cos \theta + \ddot{v} \sin \theta) - K_{12}\nu_s - K_{22}\mu_s - c}{\nu} \] (5.43)

When compared to the equations for a mono-layer beam, we observe that coupling effects (i.e. \( K_{12} \)) giving rise to quadratic terms appear in the bi-layer beam due to the breaking of the elastic symmetry of the cross section. This quadratic nonlinearity can partially cancel the effects of geometric nonlinearity [87].

5.3 Perturbation analysis

In this section, we use a perturbation analysis to find the free response of the beam under study. The results of this analysis will be used to validate the numerical tools developed in the proceeding sections. To this end, we suppose
\[ c = f_u = f_v = 0 \] (5.44)

and consider the beam with \( n_l = 1 \), i.e., mono-layer beam. We then introduce a small non-dimensional parameter \( \varepsilon \ll 1 \), such that \( \alpha^2 \mapsto \varepsilon^{-1} \alpha^2 \) (long, slender beam), \( \nu \mapsto \varepsilon \nu \), and
\( u \mapsto \varepsilon^2 u \). It follows that \( \theta = \mathcal{O}(\varepsilon) \), \( \nu - 1 = \mathcal{O}(\varepsilon^2) \), \( \mu = \mathcal{O}(\varepsilon) \), and (after division with \( \varepsilon \))

\[
\varepsilon^{-2} \alpha^2 \nu_s - \varepsilon^{-1} \mu \left( \varepsilon \alpha^{-2} \theta_{tt} - \mu_s \right) = \varepsilon \ddot{u} \cos \theta + \ddot{v} \sin \theta, \tag{5.45}
\]

\[
\varepsilon^{-1} \left( \varepsilon \alpha^{-2} \theta_{tt} - \mu_s \right) + \varepsilon^{-2} \alpha^2 (\nu - 1) = \ddot{v} \cos \theta - \varepsilon \ddot{u} \sin \theta. \tag{5.46}
\]

We will seek an oscillatory response of the two coupled PDEs. To this end, suppose

\[
u (s,t) = \tilde{v}(s,t_0,t_1,\ldots,t_k,\ldots) \quad \text{and} \quad v (s,t) = \tilde{v}(s,t_0,t_1,\ldots,t_k,\ldots) \tag{5.48}
\]

where \( t_i = \varepsilon^i t \), such that

\[
\partial_t u = \left( \sum_{k=0}^{\infty} \varepsilon^k \partial_k \right) \tilde{u} \tag{5.49}
\]

and

\[
\partial_t^2 u = (\partial_0 + \varepsilon \partial_1 + \cdots + \varepsilon^k \partial_k + \cdots)^2 \tilde{u} = \left( \sum_{k=0}^{\infty} \varepsilon^k \sum_{i=0}^{k} \partial_i \partial_{k-i} \right) \tilde{u} \tag{5.50}
\]

and similarly for \( v \). Furthermore, let

\[
\tilde{u}(s,t_0,t_1,\ldots,t_k,\ldots) = \sum_{l=0}^{\infty} \varepsilon^l U_l (s,t_0,t_1,\ldots,t_k,\ldots), \tag{5.51}
\]

\[
\tilde{v}(s,t_0,t_1,\ldots,t_k,\ldots) = \sum_{l=0}^{\infty} \varepsilon^l V_l (s,t_0,t_1,\ldots,t_k,\ldots) \tag{5.52}
\]

We proceed to substitute these expansions for the unknown functions and the differential operators into the pair of coupled PDEs.

**Order \( \varepsilon^0 \).** To this order, we obtain

\[
U_0'''' + V_0'V_0''' = 0 \tag{5.53}
\]

and

\[
V_0'''' + \partial_0^2 V_0 = 0. \tag{5.54}
\]
Denote by $\phi_n^{(1)}$ the $n$th spatial eigenfunction of the clamped-clamped Euler-Bernoulli beam corresponding to the natural frequency $\omega_n$, i.e.,

$$\phi_n^{(1)} = \cos \sqrt{\omega_n} s - \cosh \sqrt{\omega_n} s$$

$$+ \frac{\cosh \sqrt{\omega_n} - \cos \sqrt{\omega_n}}{\sin \sqrt{\omega_n} - \sinh \sqrt{\omega_n}} (\sin \sqrt{\omega_n} s - \sinh \sqrt{\omega_n} s),$$

where

$$\cosh \sqrt{\omega_n} \cos \sqrt{\omega_n} = 1.$$  \hspace{1cm} (5.56)

Since we are seeking an asymptotic approximation of the $n$th mode of the microbeam, we assume the generating solution of Eq. (5.54) as

$$V_0 = A_n (t_1, t_2, \ldots) e^{i \omega_n t_0} \phi_n^{(1)} (s) + c.c.,$$

where $c.c.$ denotes complex conjugate terms.

Using the clamped-clamped boundary conditions $U_0 (0) = U_0 (1) = 0$, it follows from (5.53) that

$$U_0 = A_n^2 e^{2i \omega_n t_0} \psi_n^{(1)} + A_n \bar{A}_n \psi_n^{(1)} + cc,$$

where

$$\psi_n^{(1)} = \frac{1}{2} \mathcal{G}_1 \left( \left( \phi_n^{(1)} \right)^2 \right),$$

where the linear operator $\mathcal{G}_1$ is given by

$$\mathcal{G}_1 (f) (s) = s \int_0^1 f (\xi) d\xi - \int_0^s f (\xi) d\xi.$$

For later reference, note that

$$\left( \mathcal{G}_1 (f) \right)' = \int_0^1 f (\xi) d\xi - f$$

and

$$\left( \mathcal{G}_1 (f) \right)'' = - f'.$$
Order $\varepsilon^1$. To this order, we obtain

$$
\alpha^2 U''_1 + \alpha^2 V''_1 V''_0 + \alpha^2 V'_1 V''_0 = \partial_0^2 U_0 + V'_0 \partial_0^2 V_0 - V''_0 V''_0
$$
\tag{5.63}
$$
\text{and}
$$
V''''_1 + \partial_0^2 V_1 = \alpha^2 U''_1 V''_0 + \alpha^{-2} \partial_0^2 V''_0 + \frac{1}{2} \alpha^2 (V'_0)^2 V''_0 - 2 \partial_0 \partial_1 V_0.
$$
\tag{5.64}

Here, the right-hand side of Eq. (5.64) equals

$$
\alpha^2 A^3_n \theta^{(1)}_n e^{3i\omega_n t_0} + (\alpha^2 A^2_n \bar{A}_n \theta^{(2)}_n + \alpha^{-2} A_n \theta^{(3)}_n - 2i\omega_n \partial_1 A_n \phi^{(1)}_n) e^{i\omega_n t_0} + c.c.,
$$
\tag{5.65}

where

$$
\theta^{(1)}_n = \frac{1}{2} \phi^{(1)''}_n \int_0^1 \left( \phi^{(1)'}_n \right)^2 d\xi,
$$
\tag{5.66}
$$
\theta^{(2)}_n = \frac{3}{2} \phi^{(1)''}_n \int_0^1 \left( \phi^{(1)'}_n \right)^2 d\xi,
$$
\tag{5.67}
$$
\theta^{(3)}_n = -\omega_n^2 \phi^{(1)''}_n.
$$
\tag{5.68}

By the Fredholm alternative, there exists a solution to Eq. (5.64) if and only if the right-hand side lies in the nullspace of the integral operator

$$
\mathcal{I}(f) = \frac{\omega_n}{2\pi} \int_0^{2\pi/\omega_n} \int_0^1 f(s,t) \phi^{(1)}_n(s) e^{-i\omega_n t_0} ds dt_0,
$$
\tag{5.69}

since

$$
\mathcal{I}(V''''_1 + \partial_0^2 V_1) = 0.
$$

It follows that

$$
\partial_1 A_n = -\frac{i}{2\omega_n} \alpha^2 A^2_n \bar{A}_n \int_0^1 \phi^{(1)}_n \theta^{(2)}_n d\xi - \frac{i}{2\omega_n} \alpha^{-2} A_n \int_0^1 \phi^{(1)}_n \theta^{(3)}_n d\xi.
$$
\tag{5.70}

We now have

$$
V''''_1 + \partial_0^2 V_1 = \alpha^2 A^3_n \theta^{(1)}_n e^{3i\omega_n t_0} + (\alpha^2 A^2_n \bar{A}_n \theta^{(4)}_n + \alpha^{-2} A_n \theta^{(5)}_n) e^{i\omega_n t_0} + c.c.,
$$
\tag{5.71}

113
where

\[ \theta_n^{(4)} = \theta_n^{(2)} - \phi_n^{(1)} \int_0^1 \theta_n^{(1)} \theta_n^{(2)} d\xi, \quad (5.72) \]

\[ \theta_n^{(5)} = \theta_n^{(3)} - \phi_n^{(1)} \int_0^1 \theta_n^{(1)} \theta_n^{(3)} d\xi. \quad (5.73) \]

It follows that

\[ V_1 = \alpha^2 A_n^3 \phi_n^{(2)} e^{3i\omega_n t_0} + \left( \alpha^2 A_n^2 \bar{\bar{A}}_n \phi_n^{(3)} + \alpha^{-2} A_n \phi_n^{(4)} \right) e^{i\omega_n t_0} + c.c., \quad (5.74) \]

where \( \phi_n^{(2)} \), \( \phi_n^{(3)} \), and \( \phi_n^{(4)} \) are particular solutions to

\[ \phi_n^{(2)\prime\prime\prime\prime} - 9 \omega_n^2 \phi_n^{(2)} = \theta_n^{(1)}, \quad (5.75) \]

\[ \phi_n^{(3)\prime\prime\prime\prime} - \omega_n^2 \phi_n^{(3)} = \theta_n^{(4)}, \quad (5.76) \]

\[ \phi_n^{(4)\prime\prime\prime\prime} - \omega_n^2 \phi_n^{(4)} = \theta_n^{(5)}. \quad (5.77) \]

with vanishing function values and derivatives at the boundaries.

It is useful to note that

\[ \int_0^1 \phi_n^{(1)} \phi_n^{(2)} d\xi = -\frac{1}{8\omega_n^2} \int_0^1 \phi_n^{(1)} \theta_n^{(1)} d\xi. \quad (5.78) \]

In contrast, note that \( \phi_n^{(3)} \) and \( \phi_n^{(4)} \) are determined from Eqs.(5.76-5.77) only up to the addition of an arbitrary multiple of \( \phi_n^{(1)} \). It follows that the integrals

\[ \int_0^1 \phi_n^{(1)} \phi_n^{(3)} d\xi \quad \text{and} \quad \int_0^1 \phi_n^{(1)} \phi_n^{(4)} d\xi \quad (5.79) \]

are as of yet undetermined.

We now have

\[ \alpha^2 U_1'' + \alpha^2 V_1' V_0' + \alpha^2 V_0' V_1' = A_n^2 e^{2i\omega_n t_0} \theta_n^{(6)} + A_n \bar{\bar{A}}_n \theta_n^{(7)} + c.c., \quad (5.80) \]
where

\[
\theta_n^{(6)} = -\phi_n^{(1)''}\phi_n^{(1)'''} - \omega_n^2\phi_n^{(1)'\prime} - 4\omega_n^2\psi_n^{(1)},
\]

\[
\theta_n^{(7)} = \theta_n^{(6)} + 4\omega_n^2\psi_n^{(1)},
\]

and, consequently,

\[
U_1 = G_1 \left( \alpha^{-2} \int_\xi^1 \left( A_n^2 e^{2i\omega_n t_0} \theta_n^{(6)} + 4A_n\tilde{A}_n\theta_n^{(7)} + c.c. \right) d\tau + V'_0(\xi)V'_1(\xi) \right),
\]

which implies that

\[
U_1 = \alpha^2 A_n^4 e^{4i\omega_n t_0} \psi_n^{(2)} + \alpha^{-2} A_n^2 e^{2i\omega_n t_0} \psi_n^{(3)} + \alpha^2 A_n\tilde{A}_n e^{2i\omega_n t_0} \psi_n^{(4)}
\]

\[
+ \alpha^{-2} A_n\tilde{A}_n \psi_n^{(5)} + \alpha^2 A_n^2 \tilde{A}_n^2 \psi_n^{(6)} + c.c.,
\]

where

\[
\psi_n^{(2)} = G_1 \left( \phi_n^{(1)'}\phi_n^{(2)''} \right),
\]

\[
\psi_n^{(3)} = G_1 \left( \phi_n^{(1)'}\phi_n^{(4)''} \right) + G_1 \left( \int_\xi^1 \theta_n^{(6)} d\tau \right),
\]

\[
\psi_n^{(4)} = G_1 \left( \phi_n^{(1)'}\phi_n^{(3)''} \right) + G_1 \left( \phi_n^{(1)'}\phi_n^{(2)''} \right),
\]

\[
\psi_n^{(5)} = G_1 \left( \phi_n^{(1)'}\phi_n^{(4)''} \right) + G_1 \left( \int_\xi^1 \theta_n^{(7)} d\tau \right),
\]

\[
\psi_n^{(6)} = G_1 \left( \phi_n^{(1)'}\phi_n^{(3)''} \right).
\]

**Order ε²** To this order, we obtain

\[
\alpha^2 U''_2 + \alpha^2 V''_2 V'_0 + \alpha^2 V''_0 V''_2 = 2\partial_0\partial_1 U_0 + \partial_0^2 U_1 + 2V'_0\partial_0\partial_1 V_0
\]

\[
+ V'_1\partial_0^2 V_0 - V''_0 V''_1 - V''_0 V''_1
\]

\[
+ V'_0\partial_0^2 V_1 + \alpha^{-2} V''_0 \partial_0^2 V'_0
\]

\[
+ \frac{1}{2}\alpha^2 U''_0 (V'_0)^2 + \alpha^2 U'_0 V'_0 V''
\]

\[
+ \frac{1}{2}\alpha^2 (V'_0)^3 V''_0 - \alpha^2 V''_0 V'_1
\]
and

\[ V_2''' + \partial^2_0 V_2 = U_0'''V_0' + V_0'\partial^2_0 U_0 + 3U_0'''V_0'' \\
+ 2(V_0''')^3 + 4U_0'''V_0''' + 7V_0''V_0''' \\
+ 2U_0''V_0''' + \frac{3}{2} (V_0'')^2 V_0''' + \frac{1}{2} (V_0'')^2 \partial^2_0 V_0 \\
- 2\partial_0 \partial_0 V_0 - \partial^2_0 V_0 - 2\partial_0 \partial_1 V_1 \\
+ 2\alpha^{-2} \partial_0 \partial_1 V_0'' + \alpha^{-2} \partial^2_0 V_1'' \\
+ \alpha^2 U_1'V_0' + \alpha^2 V_0''V_1' + \alpha^2 U_0''V_0'' + \frac{1}{2} \alpha^2 (V_0'')^2 V_1''. \]  

(5.91)

Here, the right-hand side of Eq. (5.91) equals

\[ \alpha^4 A_n^5 \theta^{(8)} e^{i\omega_n t_0} + (\alpha^4 A_n^4 A_n^\theta^{(10)} + A_n^3 \theta^{(10)}) e^{i\omega_n t_0} \\
+ (\alpha^4 A_n^3 A_n^3 \theta^{(11)} + A_n^2 \theta^{(12)} + \alpha^{-4} A_n \theta^{(13)} - 2i \omega_n \phi_n^{(1)} \partial_2 A_n) e^{i\omega_n t_0} + c.c., \]  

(5.92)

where

\[ \theta^{(8)}_n = \phi_n^{(1)'} \phi_n^{(1)''} \phi_n^{(2)'} + \frac{1}{2} (\phi_n^{(1)'})^2 \phi_n^{(2)''} + \phi_n^{(2)'} \psi_n^{(1)'} + \phi_n^{(1)''} \psi_n^{(2)'} \]  

(5.93)

\[ \theta^{(10)}_n = 2(\phi_n^{(1)''})^3 + 4\phi_n^{(1)'} \phi_n^{(1)''} \phi_n^{(1)'''} - 9\phi_n^{(2)} \int_0^1 \phi_n^{(1)''} \theta_n^{(3)'} ds + \phi_n^{(1)'} \phi_n^{(1)''} \phi_n^{(4)'} + \frac{1}{2} (\phi_n^{(1)')^2 \phi_n^{(4)''} + \phi_n^{(4)''} \psi_n^{(1)'} + 4\phi_n^{(1)''} \psi_n^{(1)''} + 3\phi_n^{(1)'} \psi_n^{(1)''} + \phi_n^{(1)''} \psi_n^{(3)'} - 9\omega_n^2 \phi_n^{(2)'} - 4\omega_n^2 \phi_n^{(1)'} \psi_n^{(1)} + 2\omega_n^2 \phi_n^{(1)'} \psi_n^{(1)'} \]  

(5.94)
\[ \theta_n^{(11)} = \phi_n^{(1)\mu} \phi_n^{(1)\nu} \phi_n^{(2)\mu} + \frac{1}{2} (\phi_n^{(1)})^2 \phi_n^{(2)\mu} \\
- \phi_n^{(3)} \int_0^1 \phi_n^{(1)} \theta_n^{(2)} \, d\xi + 3 \phi_n^{(1)} \phi_n^{(1)\nu} \phi_n^{(3)\nu} \\
+ \frac{3}{2} (\phi_n^{(1)})^2 \phi_n^{(3)\nu} + \phi_n^{(2)\nu} \phi_n^{(1)\nu} + 3 \phi_n^{(3)\nu} \phi_n^{(1)\nu} \\
+ \phi_n^{(1)\nu} \psi_n^{(4)\nu} + 2 \phi_n^{(1)\nu} \psi_n^{(6)\nu} + \frac{\phi_n^{(1)}}{4\omega_n^2} \left( \int_0^1 \phi_n^{(1)} \theta_n^{(2)} \, d\xi \right)^2 \] (5.96)

\[ \theta_n^{(12)} = \phi_n^{(1)\mu} \int_0^1 \phi_n^{(1)} \theta_n^{(2)} \, d\xi + 6 (\phi_n^{(1)})^3 + 12 \phi_n^{(1)} \phi_n^{(1)\nu} \phi_n^{(1)\mu} \\
- \phi_n^{(3)} \int_0^1 \phi_n^{(1)} \theta_n^{(3)} \, d\xi - \phi_n^{(4)} \int_0^1 \phi_n^{(1)} \theta_n^{(2)} \, d\xi + 3 \phi_n^{(1)} \phi_n^{(1)\nu} \phi_n^{(4)\nu} + \frac{3}{2} (\phi_n^{(1)})^2 \phi_n^{(4)\nu} \\
+ 3 \phi_n^{(4)\nu} \psi_n^{(1)\nu} + 12 \phi_n^{(1)\nu} \psi_n^{(4)\nu} + 9 \phi_n^{(1)\nu} \psi_n^{(1)\nu} + \phi_n^{(1)\nu} \psi_n^{(3)\nu} + 2 \phi_n^{(1)\nu} \psi_n^{(5)\nu} \\
- \omega_n^2 \phi_n^{(3)\nu} - 4 \omega_n^2 \phi_n^{(1)\nu} \psi_n^{(1)} + 6 \omega_n^2 \phi_n^{(1)\nu} \psi_n^{(1)\nu} \\
+ \frac{\phi_n^{(1)}}{2\omega_n^2} \left( \int_0^1 \phi_n^{(1)} \theta_n^{(2)} \, d\xi \right) \left( \int_0^1 \phi_n^{(1)} \theta_n^{(3)} \, d\xi \right) \] (5.97)

\[ \theta_n^{(13)} = \phi_n^{(1)\mu} \int_0^1 \phi_n^{(1)} \theta_n^{(3)} \, d\xi - \phi_n^{(4)} \int_0^1 \phi_n^{(1)} \theta_n^{(3)} \, d\xi - \omega_n^2 \phi_n^{(4)\mu} + \frac{\phi_n^{(1)}}{4\omega_n^2} \left( \int_0^1 \phi_n^{(1)} \theta_n^{(3)} \, d\xi \right)^2 \] (5.98)

By the Fredholm alternative, it again follows that

\[ \partial_2 A_n = -\frac{i}{\omega_n} \alpha A_n \bar{A}_n \int_0^1 \phi_n^{(1)} \theta_n^{(11)} \, d\xi - \frac{i}{\omega_n} A_n \bar{A}_n \int_0^1 \phi_n^{(1)} \theta_n^{(12)} \, d\xi \\
- \frac{i}{\omega_n} \alpha - A_n \bar{A}_n \int_0^1 \phi_n^{(1)} \theta_n^{(13)} \, d\xi. \] (5.99)

### 5.3.1 Reconstruction

From Eqs.(5.70,5.99), we can write the complex amplitude \( A_n \) as

\[ A_n = \frac{1}{2} a_n e^{i\nu_n} \] (5.100)
We the obtain

\[ v_n = t_1 \left( \frac{3\alpha^2 a_n^2}{16\omega_n} \int_0^1 (\phi_n^{(1)'(\xi)})^2 \, d\xi \right)^2 - \frac{\alpha^{-2}\omega_n}{2} \int_0^1 (\phi_n^{(1)'(\xi)})^2 \, d\xi \]

\[ t_2 \left( -\frac{\alpha^4 a_n^4}{32\omega_n} \int_0^1 \phi_n^{(1)(11)} \, d\xi - \frac{a_n^2}{8\omega_n} \int_0^1 \phi_n^{(1)(12)} \, d\xi - \frac{\alpha^{-4}}{2\omega_n} \int_0^1 \phi_n^{(1)(13)} \, d\xi \right) \]

(5.101)

Recalling \( t_k = \epsilon^k t \), it follows that the angular frequency of oscillation is given by

\[ \Omega_n = \omega_n + \epsilon \left( \frac{3\alpha^2 a_n^2}{16\omega_n} \int_0^1 (\phi_n^{(1)'(\xi)})^2 \, d\xi \right)^2 - \frac{\alpha^{-2}\omega_n}{2} \int_0^1 (\phi_n^{(1)'(\xi)})^2 \, d\xi \]

\[ + \epsilon^2 \left( -\frac{\alpha^4 a_n^4}{32\omega_n} \int_0^1 \phi_n^{(1)(11)} \, d\xi - \frac{a_n^2}{8\omega_n} \int_0^1 \phi_n^{(1)(12)} \, d\xi - \frac{\alpha^{-4}}{2\omega_n} \int_0^1 \phi_n^{(1)(13)} \, d\xi \right) \]

(5.102)

Assuming \( \alpha = 1 \) and after numerical calculation of the integrals in Eq. (5.102), we have

\[ \Omega_n = 22.37 + (-137.62 + 1.27a_n^2)\epsilon + (1227.67 - 88.36a_n^2 - 0.33a_n^4)\epsilon^2 \]

(5.103)

Apparently, considering only the first order of \( \epsilon \), there is an amplitude-independent negative shift in the oscillation frequency relative to \( \omega_n \). However, this shift is compensated by another term if we consider the next order of \( \epsilon \).

5.4 Mixed formulation

We proceed to develop a scheme for finding approximate solutions of Eqs. (5.39,5.40) for clamped-clamped boundary conditions. To this end, proper discretizations of both spatial and temporal domains are required. Before discretization, similar to our analysis of the
scalar beam, we proceed to rewriting Eqs (5.39,5.40) in their mix form, i.e.,

\[ A = (\ddot{m}u + c_\nu \dot{u} - f_u) \cos u_3 + (\ddot{m}v + c_\nu \dot{v} - f_v) \sin u_3 - \ddot{I}\theta \] (5.104)

\[ B = - (\ddot{m}u + c_\nu \dot{u} - f_u) \sin u_3 + (\ddot{m}v + c_\nu \dot{v} - f_v) \cos u_3 - \ddot{I}\theta^2 \] (5.105)

\[ 0 = u_3 - \arctan \frac{\partial u_2}{\partial s} \] (5.106)

\[ 0 = u_4 + K_{12} (\nu - 1) - K_{22} \frac{\partial u_3}{\partial s} \] (5.107)

\[ 0 = u_5 - \frac{\partial u_4}{\partial s} \] (5.108)

where

\[ A = \left( K_{11} - \frac{K_{12}^2}{K_{22}} \right) \nu s - \frac{K_{12}}{K_{22}} u_5 - \left( \frac{u_4}{K_{22}} + \frac{K_{12}}{K_{22}} (\nu - 1) \right) \Pi \] (5.109)

\[ B = \Pi s + \left( \frac{K_{11}}{K_{22}} - \frac{2 K_{12}^2}{K_{22}^2} \right) (\nu - 1) u_4 + \frac{K_{12}}{K_{22}} \left( K_{11} - \frac{K_{12}^2}{K_{22}} \right) (\nu - 1)^2 \]

\[ - \frac{K_{12}}{K_{22}^2} u_4^2 + \frac{u_4 + K_{12} (\nu - 1)}{K_{22}} N_x \] (5.110)

and

\[ \Pi = \frac{\lambda \theta_{tt} - u_5}{\nu} \] (5.111)

The reasoning behind the mixed formulations follows the same arguments as in Chapter 4. Here, again, the order of the spatial derivatives has reduced. Moreover, the last three variables in this formulation have a physical meaning, i.e., slope, moment, and the shear force, respectively. In terms of the auxiliary variables, the clamped-clamped boundary conditions are in Dirichlet form. Note that, in the numerical analysis in Section 5.6, we ignored the contribution of \( \ddot{I} \).

5.4.1 Discretization

The weak form is obtained by multiplying the equations in strong mixed form by a test function \( w(s) \) and integrating over \( \Omega = [0, 1] \). For both trial (denoted by \( v(s) \)) and test functions, we choose functions in the form of continuously differentiable, piecewise-polynomial
functions in $H^2$ space with arbitrary basepoint values on a spatial mesh given by the uniform partition $0 < s_1 < \cdots < s_{M+1} = 1$ of the domain $\Omega$ and the uniform subpartition $0 = \xi_1 < \cdots < \xi_{p+1} = 1$ of the $i$-th interval $\Omega_i$ in terms of the local variable

$$\xi := \frac{s - s_i}{s_{i+1} - s_i}$$

(5.112)

We again let $M$ denote a positive integer and consider the uniform partition

$$0 = s_1 < \cdots < s_{M+1} = 1.$$  

(5.113)

On the $i$-th interval $[s_i, s_{i+1}]$, we write

$$u_k(s,t) = u^{(i)}_k (\xi, t) := \sum_{l=1}^{p+1} \mathcal{L}_l(\xi) u^{(i)}_k(\xi_l, t)$$

(5.114)

$$w_k(s) = w^{(i)}_k (\xi) := \sum_{l=1}^{p+1} \mathcal{L}_l(\xi) w^{(i)}_k(\xi_l)$$

(5.115)

where $\xi$ is given by the transformation (5.112), the shape functions $\mathcal{L}_l$ are polynomials in Lagrange form, and $\{\xi_l\}_{l=1}^{p+1}$ is a uniform partition of the interval $[0, 1]$.

Following the analysis shown in [55], we choose to have the test functions $w_1, w_2, w_5$ vanish on the boundaries for the clamped-clamped beam. This choice will result in a non-degenerate set of differential-algebraic equations of index-2. Note that the continuity conditions for $u_3$, $u_4$, and $u_5$ imply the compatibility conditions for the slope, moments, and shear forces, respectively [94, 95]. We then use the collocation method with $N$ piecewise polynomials of degree $m$ in which the residual is imposed to vanish at Radau points along with the continuity condition at mesh points [5].

We look for a solution to two types of dynamic behavior of the beam under study, namely free and forced vibrations. The free vibration response gives the backbone of the frequency response. In the numerical continuation context, however, this is a degenerate problem as there exist infinitely many periodic solutions for a given set of model parameter values. As discussed in Ref. [62], this issue can be resolved by introducing an auxiliary parameter, $\beta$, ...
which adds a deviation transverse to the periodic trajectory and, hence, a periodic solution, assuming that it exists, is achieved only for $\beta = 0$. The second type of solution is the forced response in which $f_u = 0$ and $f_v = \Lambda \cos \omega t$ where $\Lambda$ is the amplitude of the excitation and $\omega$ is the corresponding frequency. Similar to our analysis in Chapter 4, an autonomous pair of equations are appended to this set to act as the forcing term in order to avoid discontinuous resetting of the phase in the non-autonomous equations.

5.5 Performance of the numerical method

In this section, we consider the mono-layer beam and compare the results obtained from the numerical approach with a second-order perturbation approximation.

5.5.1 Comparison with perturbation analysis

We proceed to compare the linear natural frequency of the beam obtained using the perturbation analysis as well as the numerical method. Table 5.1 shows sample results for three different values of $\epsilon$, where $\omega_n$ and $\omega_p$ denote the linear natural frequencies obtained using the numerical and perturbation methods, respectively.

| $\epsilon$ | $\omega_n$ | $\omega_p$ | $|\omega_n - \omega_p|$ |
|-----------|-------------|-------------|----------------------|
| $10^{-1}$ | 14.91799447 | 8.61078554  | 630.721              |
| $10^{-2}$ | 21.10903928 | 20.9970355  | 1120                 |
| $10^{-3}$ | 22.23689184 | 22.2356604  | 1231.1               |

Table 5.1: comparison of the numerical versus perturbation analysis results

We note that the linear frequency shift gets smaller as $\epsilon$ becomes smaller, as predicted by the perturbation analysis. Moreover, the last column of the table reports values of similar orders which implies that the order of the difference between the natural frequencies obtained from these two methods is indeed order two.

Figure (5.2) shows the backbone curve for $\epsilon = 10^{-3}$ using the perturbation method (first order: Red, second order: Green) along with the backbone obtained using the numerical method (in Blue). The vertical axis is the scaled displacement. We also observe that in the
Figure 5.2: Comparison of the perturbation results with the numerical method. The blue curve shows the numerical results while the first and second order perturbation results are shown in red and green, respectively. The inset shows the backbone curve emanating from the horizontal axis where, apparently, the zeroth-order shift in the frequency is not captured in the first-order perturbation analysis.

vicinity of the linear regime, the red curve is to the left of the blue cure, however, for large amplitudes, the red curve is to the right. This suggests that the actual behavior is softer than that predicted by the low-order perturbation analysis, as expected.

5.6 Parametric study of the dynamic range

The geometric nonlinearities due to large-amplitude deflections result in three co-existing responses to excitations within certain frequency ranges. Although this nonlinear behavior might be desirable in some applications, a linear response is often sought. Therefore, the larger the excitation amplitude at which there exists a single solution in the frequency response, the larger the range of desired performance of the nano/micro beam resonator. The largest response amplitude at the transition to nonlinear behavior is called the critical
amplitude and denoted by \( a_c \). In nano-resonators the range of performance is often further limited as the noise-to-signal ratios are high. The difference between \( a_c \) and the noise level is called the dynamic range of the resonator [84, 88, 96, 89].

5.6.1 Computational method

We start with a frequency-response curve as shown in Fig. 5.3, in which the maximum response amplitude is close to \( a_c \). This frequency-response curve is obtained using \( \Lambda = 0.02, c_u = c_v = 2\zeta \omega_n, M = 12, p = 3, N = 10, \) and \( m = 5 \) where \( \omega_n \) is the natural frequency and \( \zeta = 0.02 \). Apparently, this solution is in nonlinear regime where more than one stable solution exist for a certain range of excitation frequency, specified by two fold points. This implies that by reducing the excitation amplitude, we can decrease the nonlinear frequency range until the two fold points coincide and merge to give a cusp point. This cusp point is a codimension-2 bifurcation point and can be used as an indicator point in the parameter space that distinguishes between linear and nonlinear regimes.

In order to explore the solution space of this co-dimension 2 bifurcation point, we proceed to define four instances of the boundary-value problem at the parameter values shown by the red points in Fig. 5.3. The pairs in the right and left have the same frequency and we denote their amplitude difference by \( v_l \) and \( v_r \), respectively. By choosing a relatively small value for \( v_l \) and \( v_r \), \( \omega_l \) and \( \omega_r \) will represent the frequencies of the two fold points. We then define \( \omega_\delta = \omega_r - \omega_l \) and decrease it to a relatively small value while the amplitude of the excitation is free to vary. This solution corresponds (approximately) to the cusp point and we denote the amplitude of the upper right point by \( a_c \).

It is worth mentioning that an alternative approach would be to have only three instances of the solution (instead of four) with identical frequencies. Apparently, this approach reduces the size of the computational problem. We also note that in neither of these approaches, the solution instances used to represent the cusp point should be allowed to coalesce as it will lead to an ill-condition (or even singular) problem.
Figure 5.3: Frequency response curve close to the cusp point. The red points indicate the four instances of the boundary-value problem used in this study.

5.6.2 Beam configuration

Several sources can be identified that contribute to the overall stiffness of the beam in the transverse direction. Although the equations are coupled in a complicated way, we may note that $K_{22}$ is the bending linear stiffness of this beam. $\lambda$ is the coefficient of the contributions due to inertial effects and tends to decrease the overall stiffness. The hardening effect caused by the geometric nonlinearities stems from the contributions of the term with $K_{11}$ as its coefficient. Finally, $K_{12}$ which is the coupling effect introduced by the asymmetrical cross-section adds a quadratic term with softening effects.

5.6.3 Numerical study

We proceed to study the effect of beam’s geometry as well as the external applied axial force on the critical amplitude of the beam using the numerical approach. Figure 5.4 shows that $a_c$ increases with respect to $h_1/l$ for a mono-layer beam. We note that this relation is linear and, apparently, the increase is due to the change in the aspect ratio of the beam which increases the softening contributions of the inertia effects.
Figure 5.4: Critical amplitude \( (a_c) \) increases linearly with respect to the thickness to length ratio \( (h_1/l) \) of a mono-layer beam.

We choose three parameter values \( h_1/l = (0.01, 0.02, 0.03) \) and examine, numerically, how the critical amplitude varies as we apply an external force along the axial direction of the beam. Figure 5.5 shows that in all these cases the critical diameter decreases by applying the axial force. We then, again for three values of \( h_1/l \), add a uniform layer on top of the beam’s substrate. We repeat this experiment for two different cases at which the added layer is non-uniform and has discontinuities in its thickness. Particularly, for a beam’s span divided uniformly into three sections, in the first case the added layer covers the left and right sections (i.e., the middle section remains mono-layer) while in the second case only the middle section is bi-layer. We refer to these two cases by ‘YNY’ and ‘NYN’, respectively, and the case where the top layer covers the entire span by ‘YYY’. Figure 5.6 shows the relationship between \( h_2/h_1 \) and \( a_c \) in all these cases. From left to right, the three panels in this figure show the results for \( h_1/l = (0.01, 0.02, 0.03) \), respectively. We note that there is a qualitative similarity for all three values of \( h_1/l \). However, in each panel, the red curve corresponds to the beam with the top layers at the two ends which behave differently from
the other two cases. We note that the blue curve (corresponding to the uniform bi-layer beam) has the lowest $a_c$ and together with the green curve (corresponding to the beam with the top layer in the middle) exhibit an initial decrease in $a_c$ as we increase $h_2/h_1$ and reach a local minimum after which $a_c$ increases with $h_2/h_1$. On the other hand, the red curve undergoes a slight decrease initially and reaches its local minimum for relatively small values of $h_2/h_1$ after which it monotonically increases with $h_2/h_1$. In order to investigate the causes of this behavior, we show in Fig. 5.7 the nonlinear normal mode of this beam as well as its corresponding values of $u_3$, $u_4$ and $u_5$ for $h_1/l = 0.01$. The black curves in these figures represent the values corresponding to $h_1/h_2 = 0$ (i.e., a mono-layer beam), while the other colors represent the same beam configurations as in Fig. 5.6. We observe that only in the case where the added top layers are placed at the two ends of the beam, the beam softens and exhibits higher deflections compared to the mono-layer beam. Note that in Fig. 5.7, the oscillation in the graph of $u_5$ likely due to the low order of the spatial discretization.

Finally, we proceed to investigate the variation of the dynamic range with respect to the length of the top layer. Toward this end, we again consider two cases; NYN and YNY. However, the span of the beam is no longer divided into three uniform segments. Instead, the length of the top layer take distinct values as the top layer is added symmetrically around the mid-span of the beam. For example, the left panel in Fig. 5.8 shows the NYN where $d$ denotes the total length of the top layer. We observe that a mono-layer beam (i.e. $d = 0$) has the largest dynamic range and two local maximum exist; one at $d \approx 0.5$ and the other
one at $d = 1$, where $d = 1$ correspond to a fully bi-layer beam. In the right panel of Fig. 5.8, the results for the YNY case are shown. Again, $d$ denotes the total length of the top layer. We observe that the mono-layer and fully bi-layer cases correspond to local minimums of the dynamic range and the maximum occurs at $d \approx 0.5$. We also observe a $10 - 25\%$ variations in the dynamic range depending on the specific placement of the top layers and the YNY case always gives a larger dynamic range compared to the NYN case.

5.7 Conclusions

An accurate model was derived that governs the spatiotemporal dynamics of a beam in both longitudinal and transverse direction. The resulting partial differential equations are in implicit form with high-degree derivatives in the spatial domain.

A perturbation method was used to obtain a low-order solution to the free response problem of the beam as a validation tool for the numerical scheme. In the numerical method, discretization in time and space domains were performed using orthogonal collocation and the finite-element method, respectively. The free response of the beam was then obtained in order to compare with the results from the perturbation method, and the comparison was used to validate an amplitude-independent shift in the frequency response as well as the low-order softening in the backbone of the frequency response.

A numerical continuation method was employed to explore the variations of the dynamic
Figure 5.7: The corresponding values of $v$, $v_s$, $v_{ss}$, and $v_{sss}$ for $h_1/l = 0.01$ (blue:YYY, green:NYN, red:YNY, black:mono-layer).

range of the beam with respect to several design parameters. In particular, it was shown that in a mono-layer beam, the dynamic range increases linearly with the slenderness of the beam. Moreover, in a bi-layer beam where the top layer may cover only a certain part of the beam span, the relative thickness of the layers as well as the longitudinal placement of the top layer can affect the dynamic range of the beam.

The developed methods in this study can be used as tools for designing beams in nano-scale where the application demands for operations in a wide range of linear regime. In particular, with several applications of nano-beam in bio-sensors in which a functional layer is often integrated on top of the nano-resonators, the importance of design strategies that can take the dynamic range into account are significant.
Figure 5.8: Variation of dynamic range with respect to the length of the top layer. Left and right panels show NYN and YNY cases, respectively.
6.1 Introduction

Numerous studies have been conducted to analyze the performance of collocation methods for the approximate solutions of differential equations. The literature on this subject is mainly focused on both the error analysis of the collocation method and the algorithms for mesh selection in adaptive schemes. The former research is mainly on finding error bounds of specific forms of boundary value problems and conditions for achieving the superconvergence at the mesh points. The latter research, on the other hand, seeks for algorithms that can combine the analytical results on the error analysis with heuristic approaches in order to find a mesh selection strategy that suits the specific problem under study.

The asynchronous collocation algorithm introduced in Appendix B, however, has not yet been rigorously analyzed for its convergence performance. It is the goal of this chapter to initiate this study and examine a powerful approach used in the error analysis of synchronous collocation and extend it to address the asynchronous problem.

In this chapter, we strictly follow the ideas used in the error analysis of a high-order scalar differential equation in Ref. [97] and, more specifically, the analysis of a set of first-order differential equations in Ref. [98].

6.2 Collocation solution

Consider a system of first order nonlinear differential equations in explicit form

\[ D\mathbf{y}(t) = \mathbf{y}'(t) = \mathbf{F}(t, \mathbf{y}(t)) \]  

(6.1)
subject to $n$ linear boundary conditions

$$l_iy = \beta_i, \quad 1 \leq i \leq n$$  \hfill (6.2)

Here, the $l_i$’s are continuous linear functionals on $C[a, b]$ and $\beta_i$’s are constants. In Eqs. (6.1) and (6.2), $F$ and $y$ are $(n \times 1)$ vectors. We assume that each $F_i(t, y(t))$ is continuous on $[a, b]$ with possible discontinuities of derivatives at certain points. We also assume that a unique solution, $y$, exists for this problem.

An approximate solution to this problem using the collocation approach satisfies the differential equations at certain points as well as the boundary conditions. Although in practice a collocation solution is obtained using these conditions, de Boor and Swartz [97] formulated the collocation problem as a nonlinear operator equation to find the convergence rate of the collocation method with synchronous mesh. In this chapter, we employ the same idea and extend its results to study the collocation problem with asynchronous mesh. In the next section, we briefly discuss the main results by Russell [98] on the convergence of synchronous collocation.

### 6.3 Synchronous collocation

Consider a mesh, $\Delta$, defined as

$$\Delta : a = t_0 < t_1 < \cdots < t_N = b$$  \hfill (6.3)

We say that a vector-valued function $g(t)$ is an element of $C^{(s)}_\Delta[a, b]$ if each component is in $C^{(s)}[a, b]$ except possibly at points of $\Delta$ in which jump discontinuities can occur. We also define for each mesh $\Delta$

$$h = \max_{0 \leq j \leq N-1} h_j = \max_{0 \leq j \leq N-1} (t_{j+1} - t_j)$$  \hfill (6.4)

Finally, we say a function $p(t) \in \mathcal{P}_{m,\Delta}$ if every component of this function, on each interval of $\Delta$, is a polynomial of degree less than $m$.

Throughout this study, we extensively make use of the following lemma, as discussed in
Lemma 1. Any \( y \in \mathcal{C}_\Delta^{(1)}[a, b] \) satisfies

\[
y(t) = c + \int_a^b G(t, s) [Dy(s)] \, ds \tag{6.5}
\]

where \( G(t, s) \) is the Green’s function to the problem

\[
Dy(t) = f(t), \quad l_i y = 0 \tag{6.6}
\]

for \( 1 \leq i \leq n \), and \( c \) is a unique constant vector such that

\[
l_i c = l_i y = \beta_i, \quad 1 \leq i \leq n \tag{6.7}
\]

Using Lemma 1, we can define the nonlinear mapping \( T \) by

\[
Tv = F(\cdot, c + \int_a^b G(\cdot, s) v(s) \, ds) \tag{6.8}
\]

By identifying \( v = Dy \), it can be argued that finding \( v \in \mathcal{C}_\Delta[a, b] \) such that

\[
v = Tv \tag{6.9}
\]

is equivalent to solving problem defined by Eqs. (6.1,6.2).

The analysis then proceeds to defining \( Q \) as a linear projection from \( \mathcal{C}[-1, 1] \) onto \( \mathcal{P}_m \).

Similarly, the projection \( Q_\Delta \) is defined from \( \mathcal{C}_\Delta[a, b] \) such that each interval is first transformed, through a norm preserving transformation, to \([-1, 1]\) and then onto the space of polynomial function of degree less than \( m \), i.e., the image of \( Q_\Delta \) is in \( \mathcal{P}_{m, \Delta} \).

There are infinitely many ways to define \( Q \). Here, we define \( Q \) to be a polynomial interpolation at the points

\[
-1 < z_1 < z_2 < \cdots < z_m < 1 \tag{6.10}
\]

where \( z_i \)'s are called \textit{collocation nodes}. The collocation problem is equivalent to finding the
solutions $v_\Delta \in \mathcal{P}_{m,\Delta}$, such that

$$v_\Delta = Q_\Delta T v_\Delta \quad (6.11)$$

Equation (6.11) is the basis of the analyses in Refs. [97, 98]. It is then shown that, under some conditions and when Gauss points are used as the collocation nodes, the following convergence rates hold.

$$|y(t_j) - y_\Delta(t_j)| \leq \text{const } h^{2m} \quad (6.12)$$
$$\|y - y_\Delta\| \leq \text{const } h^{m+1} \quad (6.13)$$

for $0 \leq j \leq N$. Equation (6.12) shows the superconvergence at the mesh points, while Eq. (6.13) shows the convergence rate of the uniform error. In the next section, we briefly discuss the extension of the results for the convergence rate of the uniform error in the case of asynchronous collocation.

6.4 Asynchronous collocation

We restrict our attention to a system of two coupled nonlinear differential equations. Hence, we have

$$y_r' = F_r(t, y_1(t), y_2(t)), \quad r = 1, 2 \quad (6.14)$$

Similar to the analysis of the synchronous collocation, consider two linear projectors, $Q_1$ and $Q_2$, such that $Q_1 x, x \in C[-1,1]$, is in $\mathcal{P}_m$ and $Q_2 x, x \in C[-1,1]$, is in $\mathcal{P}_m$. Moreover, associate with each $Q_r$, a linear projector $Q_{\Delta r}$ on $C_{\Delta r}[a,b]$ in which a norm preserving linear transformation is used to map every interval in $C_{\Delta r}$ onto $C[-1,1]$. We, again, let $Q_r$ be a polynomial interpolation at the points

$$-1 < z_1 < z_2 < \cdots < z_{m_j} < 1 \quad (6.15)$$
Correspondingly, for \( g(t) \in \mathcal{C}_\Delta [a,b] \), \( Q_\Delta g(t) \in \mathcal{P}_{m_\Delta,\Delta} \) where \( \mathcal{P}_{m_\Delta,\Delta} \) is the space of piecewise polynomials of degree less than \( m_\Delta \) on partition \( \Delta \) defined as

\[
\Delta : a = t_0^q < t_1^r < \ldots < t_{N_r}^r = b
\]  

(6.16)

Following [97], we define the collocation problem as finding a solution \( v_{i,\Delta} \in \mathcal{P}_{m_\Delta,\Delta} \), such that

\[
v_{i,\Delta} = Q_\Delta T_i v_\Delta
\]  

(6.17)

for \( i = 1, 2 \), where \( T_1 \) and \( T_2 \) are components of the nonlinear map \( T \), and

\[
v_\Delta = \begin{cases} v_{1,\Delta_1} \\ v_{2,\Delta_2} \end{cases}
\]  

(6.18)

The following theorem aims at extending part of the results presented within Theorem 3 in Ref. [98] which, itself, was originally given in Ref. [97] for a scalar high-order differential equation.

**Theorem 1.** Assume that the nonlinear operator \( T \) is Fréchet differentiable at a solution \( v \) of (6.8), i.e., for every \( \Delta_i \) there exists a linear map \( T'_i \) on \( \mathcal{C}_\Delta \), such that

\[
E_i (w) = T_i w - T_i v - T'_i (w - v) = o (\|w - v\|)
\]  

(6.19)

Then

\[
\|v_1 - v_{1,\Delta_1}\| \leq K_{11} \text{dist}_\infty (v_1, \mathcal{P}_{m_1,\Delta_1}) + K_{12} \|v_2 - v_{2,\Delta_2}\|
\]  

(6.20)

\[
\|v_2 - v_{2,\Delta_2}\| \leq K_{22} \text{dist}_\infty (v_2, \mathcal{P}_{m_2,\Delta_2}) + K_{21} \|v_1 - v_{1,\Delta_1}\|
\]  

(6.21)

where \( K \)'s are constants.

**Proof.** From Eq. (6.19), with \( w = v_\Delta \), the collocation equations can be rewritten as

\[
v_{i,\Delta} = Q_{\Delta_i} (v_i + T'_i (v_\Delta - v) + E_i (v_\Delta))
\]  

(6.22)
where we have also used
\[ v_i = T_i v \] (6.23)

Since \( T' \) is a linear operator, we have
\[
\begin{align*}
v_{1,\Delta_1} &= Q_{\Delta_1} (v_1 + T'_{11} (v_{1,\Delta_1} - v_1) + T'_{12} (v_{2,\Delta_2} - v_2) + E_1 (v_\Delta)) \\
v_{2,\Delta_2} &= Q_{\Delta_2} (v_2 + T'_{22} (v_{2,\Delta_2} - v_2) + T'_{21} (v_{1,\Delta_1} - v_1) + E_2 (v_\Delta))
\end{align*}
\] (6.24) (6.25)

Then, if we subtract Eq. (6.24) and Eq. (6.25) from \( v_1 \) and \( v_2 \), respectively, we obtain
\[
\begin{align*}
v_1 - v_{1,\Delta_1} &= P_{11} v_1 + P_{12} (v_2 - v_{2,\Delta_2}) + (1 - Q_{\Delta_1} T'_{11})^{-1} Q_{\Delta_1} E_1 (v_\Delta) \\
v_2 - v_{2,\Delta_2} &= P_{22} v_2 + P_{21} (v_1 - v_{1,\Delta_1}) + (1 - Q_{\Delta_2} T'_{22})^{-1} Q_{\Delta_2} E_2 (v_\Delta)
\end{align*}
\] (6.26) (6.27)

where
\[
\begin{align*}
P_{11} &= (1 - Q_{\Delta_1} T'_{11})^{-1} (1 - Q_{\Delta_1}) \\
P_{12} &= (1 - Q_{\Delta_1} T'_{11})^{-1} Q_{\Delta_1} T'_{12} \\
P_{21} &= (1 - Q_{\Delta_2} T'_{22})^{-1} Q_{\Delta_2} T'_{21} \\
P_{22} &= (1 - Q_{\Delta_2} T'_{22})^{-1} (1 - Q_{\Delta_2})
\end{align*}
\] (6.28) (6.29) (6.30) (6.31)

Note that the invertibility of the operators are shown in Ref. [97] to be independent of the mesh. We have also used, from the same reference paper, the inequality
\[
\| f - Q_\Delta f \| \leq \| 1 - Q \| \text{dist}_\infty (f, \mathcal{P}_{m,\Delta})
\] (6.32)

\[ \square \]

**Corollary 1.** For \( v \in C^{(p)}[a,b] \), we have
\[
\| v - v_\Delta \| \leq \text{const.} |\Delta_1|^{\min(p,m_1)} + \text{const.} |\Delta_2|^{\min(p,m_2)}
\] (6.33)

**Proof.** The proof is immediate following the conclusion of Theorem 1 as well as Theorem 2
This result indicates that for \( p \) sufficiently large, convergence rate of a coupled set of differential equations is ruled by the lowest order when different discretizations are employed in the collocation scheme.

6.5 Conclusions

Error analysis of asynchronous collocation algorithm was briefly discussed in this chapter and some preliminary results were obtained. It was shown that the approach employed in [97] can be extended to the asynchronous case for the uniform error analysis. The results suggest that in case of two variables, the error convergence rate is no better than the worst of the two different discretizations. This analysis can be further developed to consider the superconvergence at the mesh points.
In this dissertation, we focused on the development of novel computational algorithms for the analysis of large-scale dynamical systems. After setting forth a brief study on the application of numerical continuation to power system networks and the implementation of a MATLAB-code, the dissertation extensively discusses the development of an algorithm for asynchronous collocation. The asynchronous collocation is basically an extension of the standard collocation paradigm with the capability of independent discretization for collocation nodes in which the mesh points do not necessarily coincide. This natural extension proves to be very beneficial in terms of the reduction gained in the size of a large problem with slow-fast dynamics as was examined in the analysis of a large group of aerosol particles with transient growth.

The growth dynamics of the aerosol particles was then investigated through the analysis of a suitable boundary value problem. In particular, a phenomenon called kinetic limitation was studied in order to evaluate the magnitude of the error introduced due to overestimation of the number of particles that form cloud droplets that is a result of simplified assumptions in the conventional techniques. A relatively low-cost algorithm was proposed and the amount of overestimation was shown to be significant.

The dissertation also contributes in the analysis of singularities that arises in the formulation of the beam structures. The sources of degeneracy are found in the spatial as well as the temporal discretization. Several techniques are proposed to successfully arrive at a non-degenerate formulation for the beam dynamics. This study is followed by a numerical analysis to validate the analytical finding and, also, showing the convergence rate of the modified formulations. This study is then expanded to set the computational framework of a more complicated beam model in which both the transverse and longitudinal displacements
are considered and rotary inertia effects are taken into account. An amplitude-independent
shift in the frequency response curve as well as a higher order softening effect is observed in
both the numerical and perturbation analyses. This study ends with the development of an
algorithm for parametric study of the dynamic range in nonlinear beams. It was shown that
several system parameters as well as the geometry of layered beams have significant effects
on the size of the dynamic range. It may be useful to take this observation into account in
the design of novel nano/micro-resonators.

Finally, a brief study of convergence rate is performed to show that the asynchronous
collocation algorithm, under certain conditions, can achieve high orders of convergence.

There are several lines of research that can be pursued to extend the range of findings
in this dissertation. The singularity analysis of the mixed formulation of beams shows that
this specific form of equation is a valid candidate for the analysis of the coupled beam.
The analysis of the dynamic range, however, may be used to evaluate many different design
choices, including the parameterization of the structure topology. The analytical study can
be complemented by experiments to validate the numerical results.

The results from the analysis of the cloud condensation problem demonstrated the need
for a thorough study of the growth dynamics of aerosol particles using accurate parameters
obtained from experimental observations. This can lead to a systematic investigation of the
errors in current estimates of the concentrations of cloud droplets in the atmosphere. It
might also be interesting to explore non-uniform distributions of probability mass among
the discretized particles to achieve a low-cost, yet sufficiently accurate, framework for the
analysis of cloud condensation.

The rigorous convergence analysis of the asynchronous collocation algorithm can also be
further investigated through the study of superconvergence at the mesh points and possible
conditions on the selection of mesh points in order to achieve the highest rate of convergence.
APPENDIX A

SINGULARITIES IN BEAM FORMULATIONS

A.1 A singular, skewsymmetric, persymmetric matrix

Let $p$ be a positive integer. Recall the definition of the square matrix $W_1$ in Eq. (4.65) in terms of the row matrix $\mathcal{L}$ of the Lagrangean shape functions in Eq. (4.26) for a uniform partition

$$0 = \xi_1 < \cdots < \xi_{p+1} = 1 \quad (A.1)$$

Integration by parts then yields

$$W_1 + W_1^T = \mathcal{L}^T(\xi) \cdot \mathcal{L}(\xi)|_0^1 \quad (A.2)$$

Let $\delta_{i,j}$ equal 1 for $i = j$ and zero otherwise. The $(i,j)$-th element of the matrix on the right-hand side of Eq. (A.2) is then given by $\delta_{i,p+1} \delta_{j,p+1} - \delta_{i,1} \delta_{j,1}$, i.e., all elements of $W_1 + W_1^T$ are zero except for the $(1,1)$ element which equals $-1$ and the $(p+1,p+1)$ element which equals 1. It follows that $W_1$ may be written as a sum of a skewsymmetric matrix $\hat{W}_1$ and a matrix whose elements all equal zero except for the $(1,1)$ element which equals $-1/2$ and the $(p+1,p+1)$ element which equals $1/2$.

Equation (4.26) implies that

$$\mathcal{L}_k(\xi) = \mathcal{L}_{p+2-k}(1 - \xi) \quad (A.3)$$

for all $k = 1, \ldots, p+1$, since $\xi_l = 1 - \xi_{p+2-l}$ for $l = 1, \ldots, p+1$. Variable substitution then

\footnote{The material in this appendix is collaborative efforts with Prof. Harry Dankowicz and accepted for publication in the ASME Journal of Computational and Nonlinear Dynamics [55].}
yields
\[ \int_0^1 L_k(\xi) \frac{dL_l(\xi)}{d\xi} d\xi = \int_0^1 L_{p+2-k}(\xi) \frac{dL_{p+2-l}(\xi)}{d\xi} d\xi \] (A.4)
which, by Eq. (A.2), implies that
\[ \int_0^1 L_k(\xi) \frac{dL_l(\xi)}{d\xi} d\xi = \int_0^1 L_{p+2-l}(\xi) \frac{dL_{p+2-k}(\xi)}{d\xi} d\xi \] (A.5)
provided that \( k \neq l \). We conclude that the matrix \( \hat{W}_1 \) is persymmetric. In particular, if
\[
\begin{pmatrix}
0 & w_2 & \cdots & w_{p+1}
\end{pmatrix}
\] (A.6)
represents the first row of \( \hat{W}_1 \), then its last row is given by
\[
\begin{pmatrix}
-w_{p+1} & \cdots & -w_2 & 0
\end{pmatrix}
\] (A.7)

The one-dimensional nullspace of \( W_1 \) is spanned by the column matrix
\[
u := \begin{pmatrix} 1 & \cdots & 1 \end{pmatrix}^T
\] (A.8)
since \( \partial_\xi L(\xi) \cdot w \equiv 0 \). For every integer \( n \), denote by \( Q_n \) the shifted Legendre polynomial of degree \( n \) defined on the interval \([0, 1]\) and let
\[ q_k = Q_p(\xi_k) \] (A.9)
for \( k = 1, \ldots, p+1 \). Since
\[ Q_n \left( \frac{1 - \xi}{2} \right) = (-1)^n Q_n \left( \frac{1 + \xi}{2} \right) \] (A.10)
it follows that \( q_k = q_{p+2-k} \) for even \( p \) and \( q_k = -q_{p+2-k} \) for odd \( p \). By the orthogonality properties of the Legendre polynomials, it then follows that the column matrix
\[
q := \begin{pmatrix} q_1 & \cdots & q_{p+1} \end{pmatrix}^T
\] (A.11)
spans the one-dimensional nullspace of the transpose $W_1^T$, since

$$\int_0^1 y^T \cdot \partial_\xi L^T(\xi) \cdot L(\xi) \cdot q \, d\xi = 0 \quad (A.12)$$

for all $y$.

Suppose henceforth that $Mp \geq 2$ and recall the definition of the $M(p+1) \times (Mp-1)$ and $M(p+1) \times (Mp+1)$ matrices $\tilde{P}_1$ and $\tilde{P}_2$ whose columns are a basis for the nullspaces $\tilde{C}_1$ and $\tilde{C}_2$, respectively. Let

$$\tilde{W}_1 := I_M \otimes W_1 \quad (A.13)$$

and denote by $A_{i,j}$ matrix product

$$\tilde{P}_i^T \cdot \tilde{W}_1 \cdot \tilde{P}_j \quad (A.14)$$

for $i, j = 1, 2$. It is again straightforward to see that the one-dimensional nullspace of the $(Mp+1) \times (Mp+1)$ matrix $A_{2,2}$ is spanned by the column matrix

$$\tilde{u} = \left(1 \ \cdots \ 1\right)^T \quad (A.15)$$

Now define the family of polynomials $\tilde{Q}_i(\xi) := Q_p(1-\xi)$ for odd $i$ and $\tilde{Q}_i(\xi) := Q_p(\xi)$ for $i$ even. The one-dimensional nullspace of the $(Mp+1) \times (Mp+1)$ matrix $A_{2,2}^T$ is then spanned by the column matrix $\tilde{q}$, where

$$\tilde{q}_1 = \tilde{Q}_1(\xi_1) \quad (A.16)$$

and

$$\tilde{q}_{(i-1)p+j} = \tilde{Q}_i(\xi_j) \quad (A.17)$$

for $i = 1, \ldots, M$ and $j = 2, \ldots, p + 1$.

The $(Mp-1) \times (Mp-1)$ matrix $A_{1,1}$ is skewsymmetric and therefore singular if and only if $Mp$ is even. The corresponding nullspace (as well as that of the transpose $A_{1,1}^T$) is spanned by the column matrix $\tilde{v}$, where

$$\tilde{v}_{j-1} = \tilde{Q}_1(\xi_j) - 1 \quad (A.18)$$
for \( j = 2, \ldots, p \) and

\[
\tilde{v}_{(i-1)p+j-1} = \tilde{Q}_i(\xi_j) - 1 \tag{A.19}
\]

for \( i = 2, \ldots, M \) and \( j = 1, \ldots, p \).

Finally, we observe that the \((Mp - 1) \times (Mp + 1)\) rectangular matrix \( A_{1,2} \) has a two-dimensional nullspace spanned by \( \tilde{q} \) and \( \tilde{u} \). It follows that the column space of the \((Mp + 1) \times (Mp - 1)\) rectangular matrix \( A_{2,1} \) is the orthogonal complement to \( \{\tilde{q}, \tilde{u}\} \).

### A.2 A positive definite, symmetric matrix

Next recall the definition of the positive definite, symmetric matrix \( W_2 \) in Eq. (4.66). Let

\[
\tilde{W}_2 := I_M \otimes W_2 \tag{A.20}
\]

and

\[
B_{i,j} = \tilde{P}_i^T \cdot \tilde{W}_2 \cdot \tilde{P}_j \tag{A.21}
\]

for \( i, j = 1, 2 \). Since \( \tilde{W}_2 \) is positive definite and symmetric, and since the columns of \( \tilde{P}_1 \) and \( \tilde{P}_2 \), respectively, are linearly independent, it follows that the \((Mp - 1) \times (Mp - 1)\) matrix \( B_{1,1} \) and the \((Mp + 1) \times (Mp + 1)\) matrix \( B_{2,2} \) are invertible for arbitrary values of \( M \) and \( p \). It follows from the orthogonality properties of the Legendre polynomials that

\[
\tilde{q}^T \cdot B_{2,2} \cdot \tilde{u} = 0 \tag{A.22}
\]

By the analysis in the previous section, it follows that the symmetric matrix \( A_{1,2} \cdot B_{2,2}^{-1} \cdot A_{2,1} \) is negative definite and, consequently, invertible with a negative definite inverse.

Now consider the matrix product \( A_{2,2} \cdot B_{2,2}^{-1} \cdot A_{2,2} \), whose nullspace contains the nullvector \( \tilde{u} \) of \( A_{2,2} \). The existence of a second, linearly independent nullvector \( \tilde{w} \) follows from Eq. (A.22), since the equation

\[
B_{2,2} \cdot \tilde{u} = A_{2,2} \cdot \tilde{w} \tag{A.23}
\]

may be solved for \( \tilde{w} \) if and only if the left-hand side is orthogonal to the nullspace of \( A_{2,2}^T \).
Here, a solution is given by \( \tilde{w}_1 = 0 \) and \( \tilde{w}_{(i-1)p+j} = \xi_j + i - 1 \) for \( i = 1, \ldots, M \) and \( j = 2, \ldots, p+1 \). By orthogonality of the Legendre polynomials, it holds that

\[
\tilde{q}^T \cdot B_{2,2} \cdot \tilde{w} = 0 \tag{A.24}
\]

provided that \( p \geq 2 \).

The two-dimensional nullspace of \( A_{2,2} \cdot B_{2,2}^{-1} \cdot A_{2,2} \) intersects the image of \( B_{2,2}^{-1} \cdot A_{2,1} \) if and only if there exists a nonzero linear combination \( \alpha \tilde{u} + \beta \tilde{w} \) such that

\[
\begin{pmatrix}
\tilde{q}^T \\
\tilde{u}^T
\end{pmatrix} \cdot B_{2,2} \cdot (\alpha \tilde{u} + \beta \tilde{w}) = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{A.25}
\]

It follows by Eq. (A.22) and (A.24) that the first row of (A.25) is trivially satisfied and we conclude that, as long as \( p \geq 2 \), the matrix

\[
A_{2,2} \cdot B_{2,2}^{-1} \cdot A_{2,2} \cdot B_{2,2}^{-1} \cdot A_{2,1} \tag{A.26}
\]

has a one-dimensional nullspace, whose image under \( B_{2,2}^{-1} \cdot A_{2,1} \) is spanned by \( 2 \tilde{w} - M \tilde{u} \). In particular, a solution to the equation

\[
B_{2,2} \cdot (2 \tilde{w} - M \tilde{u}) = A_{2,1} \cdot \tilde{z} \tag{A.27}
\]

is given by

\[
\tilde{z}_{j-1} = (M - \xi_j) \xi_j \tag{A.28}
\]

for \( j = 2, \ldots, p \) and

\[
\tilde{z}_{(i-1)p+j-1} = (M + 1 - \xi_j - i)(\xi_j + i - 1) \tag{A.29}
\]

for \( i = 2, \ldots, M \) and \( j = 1, \ldots, p \).

When \( Mp \) is odd, the invertibility of \( A_{1,1} \) now implies that the matrix product

\[
A_{2,2} \cdot B_{2,2}^{-1} \cdot A_{2,2} \cdot B_{2,2}^{-1} \cdot A_{2,1} \cdot B_{1,1}^{-1} \cdot A_{1,1} \tag{A.30}
\]
has a one-dimensional nullspace spanned by the image of $\tilde{z}$ under $A_{1,1}^{-1} \cdot B_{1,1}$. In contrast, when $Mp$ is even, the one-dimensional nullspace of this matrix product coincides with that of $A_{1,1}$.

A.3 Matrix singularities

We proceed to investigate the invertibility of each of the $4Mp \times 4Mp$ matrices

$$V_1 := \begin{pmatrix} 0 & 0 & 0 & MA_{1,2} \\ MA_{2,1} & -B_{2,2} & 0 & 0 \\ 0 & MA_{1,2} & -B_{1,1} & 0 \\ 0 & 0 & MA_{2,1} & -B_{2,2} \end{pmatrix}$$  \hspace{1cm} (A.31)

$$V_2 := \begin{pmatrix} MA_{1,1} & -B_{1,1} & 0 & 0 \\ 0 & MA_{2,1} & -B_{2,2} & 0 \\ 0 & 0 & MA_{2,2} & -B_{2,2} \\ 0 & 0 & 0 & MA_{1,2} \end{pmatrix}$$  \hspace{1cm} (A.32)

$$V_3 := \begin{pmatrix} MA_{2,1} & -B_{2,1} & 0 & 0 \\ 0 & MA_{2,1} & -B_{2,2} & 0 \\ 0 & 0 & MA_{1,2} & -B_{1,2} \\ 0 & 0 & 0 & MA_{1,2} \end{pmatrix}$$  \hspace{1cm} (A.33)

whose bottom three rows correspond to the coefficient matrices in front of $\left( \begin{array}{c} \tilde{x}_1 \\ \tilde{x}_3 \\ \tilde{x}_4 \\ \tilde{x}_5 \end{array} \right)^T$ in Eqs. (4.71-4.73), (4.74-4.76), and (4.77-4.79), respectively, and whose first row is obtained from the discretization of the integral term $\int_0^1 \partial_s u_5(s)w_2(s) \, ds$ in Eq. (4.44).

By the invertibility of $B_{1,1}$ and $B_{2,2}$, it follows that $\left( v_1^T \ v_2^T \ v_3^T \ v_4^T \right)^T$ is a nullvector.
of $V_1$ only if

$$v_4 = MB_{2,2}^{-1} \cdot A_{2,1} \cdot v_3$$  \hspace{1cm} (A.34)
$$v_3 = MB_{1,1}^{-1} \cdot A_{1,2} \cdot v_2$$  \hspace{1cm} (A.35)
$$v_2 = MB_{2,2}^{-1} \cdot A_{2,1} \cdot v_1$$  \hspace{1cm} (A.36)

where $MA_{1,2} \cdot v_4 = 0$. It follows that $v_4 = \alpha \tilde{q} + \beta \tilde{u}$ for some scalars $\alpha, \beta$, for which

$$
\begin{pmatrix}
\tilde{q}^T \\
\tilde{u}^T
\end{pmatrix} \cdot B_{2,2} \cdot (\alpha \tilde{q} + \beta \tilde{u}) =
\begin{pmatrix}
0 \\
0
\end{pmatrix}
$$  \hspace{1cm} (A.37)

Since this only holds if $\alpha = \beta = 0$, it follows that $V_1$ is invertible for all $M$ and $p$.

By the invertibility of $B_{1,1}$ and $B_{2,2}$, it follows that \( \begin{pmatrix}
v_1^T \\
v_2^T \\
v_3^T \\
v_4^T
\end{pmatrix}^T \) is a nullvector of $V_2$ if

$$v_4 = MB_{2,2}^{-1} \cdot A_{2,2} \cdot v_3$$  \hspace{1cm} (A.38)
$$v_3 = MB_{2,2}^{-1} \cdot A_{2,1} \cdot v_2$$  \hspace{1cm} (A.39)
$$v_2 = MB_{1,1}^{-1} \cdot A_{1,1} \cdot v_1$$  \hspace{1cm} (A.40)

where $MA_{1,2} \cdot v_4 = 0$. From the Appendix A.1, we conclude that one such nullvector is obtained with $v_1 = \tilde{v}$ and $v_2 = v_3 = v_4 = 0$ when $M \rho$ is even. For odd values of $M \rho$, the results above show the existence of a nullvector with $v_4 = \tilde{u}$, $v_3 = \tilde{w} - M \tilde{u}/2$, $v_2 = \tilde{z}/2$ and $v_1 = A_{1,1}^{-1} \cdot B_{1,1} \cdot \tilde{z}/2$. In either case, the matrix $V_2$ is singular. It is straightforward to show that $v_1 \neq 0$ for every nullvector \( \begin{pmatrix}
v_1^T \\
v_2^T \\
v_3^T \\
v_4^T
\end{pmatrix}^T \) of the transpose $V_2^T$.

The column matrix \( \begin{pmatrix}
v_1^T \\
v_2^T \\
v_3^T \\
v_4^T
\end{pmatrix}^T \) is a nullvector of $V_3$ if

$$MA_{2,1} \cdot v_1 = B_{2,1} \cdot v_2$$  \hspace{1cm} (A.41)
$$MA_{2,1} \cdot v_2 = B_{2,2} \cdot v_3$$  \hspace{1cm} (A.42)
$$MA_{1,2} \cdot v_3 = B_{1,2} \cdot v_4$$  \hspace{1cm} (A.43)
where $MA_{1,2} \cdot v_4 = 0$. It follows that $v_4 = \alpha \tilde{q} + \beta \tilde{u}$ for some scalars $\alpha, \beta$, for which

$$B_{2,1} \cdot (A_{1,2} \cdot B_{2,2}^{-1} \cdot A_{2,1})^{-1} \cdot B_{1,2} \cdot (\alpha \tilde{q} + \beta \tilde{u}) \quad (A.44)$$

is orthogonal to $\tilde{q}$ and $\tilde{u}$. Since $B_{1,2} \cdot \tilde{q}$ and $B_{1,2} \cdot \tilde{u}$ are both nonzero, the negative definiteness of $(A_{1,2} \cdot B_{2,2}^{-1} \cdot A_{2,1})^{-1}$ implies that $V_3$ is invertible for all $M$ and $p$. 
In this section, we develop the formulation for asynchronous discretization of a system of ordinary differential equations.

### B.1 Discretization

Consider, in particular, the solution $y(t) \in \mathbb{R}^n$ on the interval $[0, T]$ to an ordinary differential equation of the form

$$\dot{y} = f(y, p) \quad (B.1)$$

in terms of a vector field $f$ and some choice of problem parameters $p \in \mathbb{R}^q$. Let $\{\{\kappa_{\alpha,j}\}_{j=1}^{N_{\alpha}}\}_{\alpha \in \mathcal{A}}$ denote a finite collection of sequences, such that

$$\sum_{j=1}^{N_{\alpha}} \kappa_{\alpha,j} = 1, \forall \alpha \in \mathcal{A}. \quad (B.2)$$

Each $\alpha \in \mathcal{A}$ corresponds to a partition

$$\pi_{\alpha} : \quad 0 = t_{\alpha,0} < \cdots < t_{\alpha,j} = T \sum_{k=1}^{j} \kappa_{\alpha,k} < \cdots < t_{\alpha,N_{\alpha}} = T \quad (B.3)$$

of the interval $[0, T]$ in terms of the mesh nodes $t_{\alpha,j}$, for $j = 1, \ldots, N_{\alpha}$. For each $i$, we associate $\pi_{\alpha}$, for some $\alpha \in \mathcal{A}$, to $y_i$ by introducing a corresponding continuous approximant,

---

1 The material in this appendix is collaborative efforts with Prof. Harry Dankowicz and Prof. Matthew West and based on a journal manuscript being prepared for submission.
defined for $t \in [t_{\alpha,j}, t_{\alpha,j+1}]$ in terms of the independent variable

$$\sigma := 2 \frac{t - t_{\alpha,j}}{T_{\kappa_{\alpha,j+1}}} - 1 \in [-1, 1] \quad \text{(B.4)}$$

the Lagrange polynomials

$$L_l(\sigma) := \prod_{p=0, p \neq l}^{m} \frac{\sigma - \sigma_p}{\sigma_l - \sigma_p}, \quad \text{(B.5)}$$

and the Lagrange interpolant

$$g_{i,j}(\sigma) := \sum_{l=1}^{m+1} y_{i,(m+1)(j-1)+l} L_l(\sigma), \quad \text{(B.6)}$$

parameterized by its values $y_{i,(m+1)(j-1)+l}$ at the $m + 1$ base points $\sigma_1, \ldots, \sigma_{m+1}$.

Given $\alpha$, there exists a unique integer $j \in \{1, \ldots, N_{\alpha}\}$ for each value of $t \in (0, T]$ such that

$$\frac{t - t_{\alpha,j}}{T_{\kappa_{j+1}}} \in (-1, 1] \quad \text{(B.7)}$$

If we associate $j = 1$ and $\sigma = -1$ with $t = 0$, then the corresponding map

$$\Sigma_\alpha : [0, T] \mapsto \{(j, \sigma) | 1 \leq j \leq N_{\alpha}, -1 < \sigma \leq 1 \text{ or } j = 1, \sigma = -1\} \quad \text{(B.8)}$$

is a bijection with inverse given by

$$t = t_{\alpha,j} + \frac{1 + \sigma}{2} T_{\kappa_{\alpha,j+1}}. \quad \text{(B.9)}$$

The discretization of the system of differential equations is now complete with the imposition of collocation conditions on the $i$-th equation in terms of a distribution of collocation nodes on the partition $\pi_\alpha$. In particular, we choose here to introduce nodes for the $j$-th mesh interval, defined in terms of the local variable $\sigma$ by the $m$ roots $z_1, \ldots, z_m$ of the $m$-th degree Legendre polynomial on the interval $[-1, 1]$. The maps $\{\Sigma_\alpha\}_{\alpha \in \mathcal{A}}$ and their inverses may then be used to identify the associated pairs $(j, \sigma)$ for each component of $y$ appearing in the $i$-th component of the vector field.
Now let the $1 \times N_\alpha(m + 1)$ matrix

$$y_i := \left( \cdots y_{i,(m+1)(j-1)+1} \cdots y_{i,(m+1)j} \cdots \right)$$ (B.10)

collect the base point values for all of the polynomial approximants of $y_i$, and define the column matrix

$$y_{bp} = \left( y_1 \cdots y_n \right)^T.$$ (B.11)

Continuity across each of the mesh nodes may then be expressed in terms of the linear condition $Q \cdot y_{bp} = 0$ for some (very sparse) matrix $Q$ whose nonzero entries equal 1 or $-1$. Similarly, the values of the polynomial approximants appearing in the $i$-th component of the vector field, as well as of their derivatives, evaluated on the collocation nodes associated with the corresponding partition may be obtained in terms of the matrix products $W_i \cdot y_{bp}$ and $W'_i \cdot y_{bp}$ for some mesh-dependent matrices $W_i$ and $W'_i$. In the special case of a synchronous mesh with $\mathcal{A} = \{\alpha\}$, let

$$W = \text{Id}_{nN_\alpha} \otimes L,$$ (B.12)

and

$$W' = \text{Id}_{nN_\alpha} \otimes L'$$ (B.13)

where the $(p,l)$-th entries of the $m \times (m + 1)$ matrices $L$ and $L'$ equal $L_l(z_p)$ and $L'_l(z_p)$, respectively. It follows that the matrix products $y_{cn} = W \cdot y_{bp}$ and $y'_{cn} = W' \cdot y_{bp}$ collect the values of all the polynomial approximants and their derivatives at the corresponding collocation nodes.

### B.2 Adaptivity

In an adapted mesh, the mesh $\pi_\alpha$ associated with a group of components of $y$ is chosen to equidistribute an estimated total interpolation error across the corresponding mesh intervals. Specifically, for each $i$, such that $y_i$ is associated with the mesh $\pi_\alpha$, consider the piecewise linear function $\phi_i$, with $\phi_i(0) = 0$, that interpolates the $m$-th derivative of the piecewise-polynomial approximant of $y_i$ at the mid-point of each mesh interval. The error distribution
function $\theta_\alpha$ associated with $\pi_\alpha$ is then defined by

$$\theta_\alpha(t) := \int_0^t \max_i |\phi_i'(\tau)|^{\frac{1}{m+1}} \, d\tau$$  \hfill (B.14)

and the number of mesh intervals is given by

$$N_\alpha = \theta_\alpha(T) \left( \frac{\hat{C}}{TOL} \right)^{\frac{1}{m+1}}$$  \hfill (B.15)

for some known constant $\hat{C}$ and desired tolerance TOL [7]. Let $\theta_{\alpha,j} := \theta_\alpha(t_{\alpha,j})$ for $j = 1, \ldots, N_\alpha$ denote the values of the error distribution function on the corresponding mesh nodes. The values of $\kappa_{\alpha,j}$ for $j = 1, \ldots, N_\alpha$ are then chosen to ensure that $\theta_{\alpha,j} - \theta_{\alpha,j-1} = \theta_\alpha(T)/N_\alpha$.

Adaptivity during continuation results in changes to the discretization parameters $N_\alpha$ and $\kappa_{\alpha,j}$ along the collection of computed solutions to the collocation problem. Such changes may result in changes to the embedding dimension of the continuation problem, as well as to the interpretation of individual unknowns. Continuation beyond each such adaptive change is accomplished by sampling a previously obtained piecewise-polynomial solution in order to compute an initial solution guess for the new collection of unknown base point values. Similar interpolation may also be applied to each tangent vector along the original solution manifold in order to estimate the tangent space to the new solution manifold.

The decision to associate the components of $y$ with individual meshes $\pi_\alpha$ for $\alpha \in \mathcal{A}$ may be rationalized in terms of a constrained minimization problem. To this end, let $z_\alpha$ denote the number of components of $y$ associated with the mesh $\pi_\alpha$, such that

$$\sum_{\alpha \in \mathcal{A}} z_\alpha = n.$$  \hfill (B.16)

The total number of mesh intervals is then given by

$$N = \sum_{\alpha \in \mathcal{A}} z_\alpha N_\alpha = \left( \frac{\hat{C}}{TOL} \right)^{\frac{1}{m+1}} \sum_{\alpha \in \mathcal{A}} z_\alpha \theta_\alpha N_\alpha.$$  \hfill (B.17)
Suppose that the set \( A \) is sorted such that the corresponding values of \( \theta_{\alpha,N} \) are an increasing sequence and let \( \Theta \) be a smooth, increasing function such that \( \Theta(0) = 0 \) and

\[
\Theta \left( \sum_{i=1}^{q} z_{\alpha,i} \right) = \theta_{\alpha q,N_{\alpha q}}. \tag{B.18}
\]

It follows that

\[
N = N \left( z_{\alpha 1}, \ldots, z_{\alpha Q} \right) := \left( \frac{\hat{C}}{TOL} \right)^{\frac{1}{m+1}} \sum_{q=1}^{Q} \left( z_{\alpha q} \Theta \left( \sum_{i=1}^{q} z_{\alpha,i} \right) \right) \tag{B.19}
\]

for some integer \( Q \). Let \( \bar{z}_\alpha \) for \( \alpha \in A \) denote a solution to the constrained optimization problem

\[
\min_{\sum_{q=1}^{Q} z_{\alpha q} = n} N \left( z_{\alpha 1}, \ldots, z_{\alpha Q} \right). \tag{B.20}
\]

For each \( r = 1, \ldots, Q \), it follows that

\[
\Theta \left( \sum_{i=1}^{r} z_{\alpha i} \right) + \sum_{q=r}^{Q} z_{\alpha q} \Theta' \left( \sum_{i=1}^{q} z_{\alpha i} \right) + \lambda = 0 \tag{B.21}
\]

in terms of the Lagrange multiplier \( \lambda \). Elimination of \( \lambda \) using the equation for \( r = Q \) then yields

\[
\Theta \left( \sum_{i=1}^{r} z_{\alpha i} \right) + \sum_{q=r}^{Q-1} z_{\alpha q} \Theta' \left( \sum_{i=1}^{q} z_{\alpha i} \right) = \Theta \left( n \right) \tag{B.22}
\]

for \( r = 1, \ldots, Q - 1 \). We approximate the derivatives in Eq. (B.22) using finite difference to obtain

\[
\Theta \left( \sum_{i=1}^{r} z_{\alpha i} \right) + \sum_{q=r}^{Q-1} \Theta \left( \sum_{i=1}^{q} z_{\alpha i} \right) - \Theta \left( \sum_{i=1}^{q-1} z_{\alpha q} \right) = \Theta \left( n \right) \tag{B.23}
\]

or, equivalently,

\[
\theta_{\alpha r,N_{\alpha r}} - \theta_{\alpha r-1,N_{\alpha r-1}} = \theta_{\alpha Q,N_{\alpha Q}} - \theta_{\alpha Q-1,N_{\alpha Q-1}} \tag{B.24}
\]

for \( r = 1, \ldots, Q - 1 \) and with \( \theta_{\alpha 0,N_{\alpha 0}} := 0 \). It follows that we should choose the integers \( z_{\alpha} \) so that the values \( \theta_{\alpha i,N_{\alpha i}} \) for \( i = 1, \ldots, Q \) are successive integer multiples of \( \theta_{\alpha 1,N_{\alpha 1}} \).

It is worth noting that equidistribution of the accumulated error when sorted in the
ascending order suggests that an optimal solution is obtained when two variables with close values of accumulated error tend to take similar mesh partitions.

B.3 Performance

We investigate the application of an adaptive asynchronous collocation scheme to the FitzHugh-Nagumo model [99]

\[
\dot{v} = c \left( v - \frac{1}{3} v^3 + w \right), \quad \dot{w} = - \frac{(v - a + bw)}{c},
\]

in order to compare the observed convergence rates in the estimated error with those found using a synchronous mesh with or without adaptation. Specifically, we seek continuous piecewise-polynomial approximants \( v(t) \) and \( w(t) \) to the periodic solution \( (v_{\text{per}}, w_{\text{per}}) \) obtained for \( a = 0, b = 0.8, \) and \( c = 3 \) and shown in Fig. B.1 (computed using a uniform, synchronous mesh with \( N = 10^6 \)).

Here, we examine three different cases, viz., (i) a uniform synchronous mesh, (ii) an adapted synchronous mesh, and (iii) an adapted asynchronous mesh. For each numerical experiment, we report on discretization errors in terms of the supremum norms \( \sup_t |v(t) - v_{\text{per}}(t)| \) and \( \sup_t |w(t) - w_{\text{per}}(t)| \) (in Figures B.2 and B.4), and on the maximum norms...
max_i |v(t_{v,i}) - v_{\text{per}}(t_{v,i})| and max_i |w(t_{w,i}) - w_{\text{per}}(t_{w,i})| (in Figures B.3 and B.5). For each mesh, we let \( h = T \max_i \kappa_i \) and estimate the convergence rate \( r \) associated with the asymptotic behavior \( \mathcal{O}(h^r) \) of each of the discretization errors. In each case, the appropriate mesh selection algorithm is used to generate an initial mesh with \( N = 10 \) corresponding to the \( h_0/h = 1 \) rows in each of the tables. Further rows are generated by halving each of the mesh intervals (without any further adaptation).

Figure B.2: Convergence of the uniform error. Blue, red, and green correspond to cases (i), (ii), and (iii), respectively. Panel a: error for \( v, m = 2 \); Panel b: error for \( v, m = 3 \); Panel c: error for \( w, m = 2 \); Panel d: error for \( w, m = 3 \).

As a reference, we recall from [98] the theoretical values of \( r = m + 1 \) and \( r = 2m \) for the corresponding convergence rates in case (i). The data in Figures B.2-B.3 confirms this theoretical prediction, shows the same rates of convergence in case (ii), and a lower rate of convergence in case (iii), especially for the maximal discretization error across the variable-
Figure B.3: Convergence of error at the mesh points. Blue, red, and green correspond to cases (i), (ii), and (iii), respectively. Panel a: error for $v$, $m = 2$; Panel b: error for $v$, $m = 3$; Panel c: error for $w$, $m = 2$; Panel d: error for $w$, $m = 3$.

specific mesh nodes. The latter is caused by the interpolation errors within a mesh interval of one variable propagating to the mesh-point errors of the other variable. This coupling of errors is further explored by the data in Figures B.4-B.5. Here, a fine uniform mesh is used for $w$ in order to ensure that the corresponding interpolation error is of the same order as the mesh-point discretization error associated with the adaptive asynchronous mesh for $v$. The resulting convergence rates again agree with the theoretical predictions for the synchronous case.
Figure B.4: Convergence of the uniform error. Panel a: error for $v, m = 2$; Panel b: error for $v, m = 3$; Panel c: error for $w, m = 2$; Panel d: error for $w, m = 3$. 
Figure B.5: Convergence of error at the mesh points. Panel a: error for $v$, $m = 2$; Panel b: error for $v$, $m = 3$; Panel c: error for $w$, $m = 2$; Panel d: error for $w$, $m = 3$. 
APPENDIX C
CODES FOR ASYNCHRONOUS COLLOCATION TOOLBOX

function opts = adaphs_isol2sol(opts, prefix, varargin)
% (c) Mehdi Saghafi, 2014

% This function calls the hybrid system toolbox with asynchronous
discretization. We have
% varargin = [fhan, (dfdx), (dfdp), segs, p0, fbc, (dfbcdx)]
% fhan is the function handle of the zero function
% dfdx is the function handle of Jacobian (wrt variables) of the zero function
% dfdp is the function handle of Jacobian (wrt parameters) of the zero function
% segs is an array of structures that contains the values at collocation
% p0 is the parameter values
% fbc is the function handle for boundary conditions
% dfbcdx is the Jacobian of the boundary conditions function

argidx = 1;
argidx = argidx + 2;
deridx = coco_get_id(prefix, 'adapcoll');
dat = coco_get(opts, deridx);

% N : number of intervals in the collocation discretization at current step
% N0: number of intervals in the collocation discretization at previous step
N = dat.NTST;
N0 = dat.NTST0;

% kappa : mesh density function (\kappa) in the current step
% kappa0: mesh density function (\kappa) in the previous step
kappa = dat.kappa;
else

157
kappa = cell(numel(segs),1);
end
if isfield(dat,'kappa0')
    kappa0 = dat.kappa0;
else
    kappa0 = cell(numel(segs),1);
end

% async_ind0: the indices of mesh groups assigned to each variable
if ~isfield(dat,'async_ind0')
    dat.async_ind0 = cell(numel(segs),1);
end

% In this for-loop, the asynchronous collocation toolbox is called for all
% the segments of the multi-segment boundary value problem.
for i=1:numel(segs)
    fid = coco_get_id(prefix, sprintf('adaphs%d', i));
    opts = coco_set(opts, ids, 'NTST', N{i,1}, 'NTST0', N0{i,1}, ...
                    'kappa', kappa{i}, 'kappa0', kappa0{i}, 'group', dat.group{i,1}, ...
                    'async_ind0', dat.async_ind0{i,1});
    opts = adapcoll_start(opts, fid, fhan, segs(i), p0);
end

data.nsegs = numel(segs);
data.fhan = varargin{argidx+3};
dbcdxhan = [];
if argidx+4<nargin-2 && ischar(varargin{argidx+4})
    dbcdxhan = varargin{argidx+5};
    argidx = argidx+2;
end

fid = coco_get_id(prefix, 'adaphs');
data = coco_merge(coco_get(opts, fid),data);
ParNames = {};
for i1 = 1:numel(p0)
    ParNames = [ParNames sprintf('PAR(%d)', i1)];
end
defaults.ParNames = ParNames;
data = coco_merge(defaults, data);
data = coco_merge(adaphs_system(opts, prefix, numel(segs)), data);
data.modes = [segs.mode];

% This assignment calls the actual zero function for the hybrid system
opts = adaphs_create(opts, data);
end
function [opts, argidx] = adapcoll_start(opts, varargin)
% (c) Mehdi Saghafi, 2014

% This function starts the construction of the asynchronous collocation problem.
% varargin = [prefix, fhan, (dfdxhan), (dfdphan), segs, p0]

argidx = 1;
prefix = varargin{argidx};
argidx = argidx + 1;

fhan = varargin{argidx};
if ~isa(fhan, 'function_handle')
    error('%s: argument %d must be a function handle', ...
          mfilename, argidx);
end

dfdxhan = [];
dfdphan = [];
dfdxname = sprintf('%s DFDX', func2str(fhan));
if any(exist(dfdxname, 'file') == [2 3])
    dfdxhan = str2func(dfdxname);
end
dfdpname = sprintf('%s DFDP', func2str(fhan));
if any(exist(dfdpname, 'file') == [2 3])
    dfdphan = str2func(dfdpname);
end

while ischar(varargin{argidx+1})
    switch lower(varargin{argidx+1})
    case 'dfdx'
        dfdxhan = varargin{argidx+2};
    case 'dfdp'
        dfdphan = varargin{argidx+2};
    otherwise
        error('%s: unknown option '''%s''', mfilename, varargin{argidx+1});
    end
    argidx = argidx + 2;
end
argidx = argidx + 1;

if isstruct(varargin{argidx})
data = varargin{argidx};
else
    error('%s: unknown option '''%s''', mfilename, varargin{argidx});
end
argidx = argidx + 1;
p0 = varargin{argidx};

% The required matrices/structures are formed through the following call
[data, x0] = adapcoll_system(data, p0);

data.fhan = fhan;
data.dfdxhan = dfdxhan;
data.dfdphan = dfdphan;
data.prefix = prefix;

% The zero function of the asynchronous collocation problem is
opts = adapcoll_create(opts, data, x0, p0);
end
function [data, x0] = adapcoll_system(data, p0)
% (c) Mehdi Saghafi, 2014

% This function construct the required interpolation matrices and other
% necessary structures for the asynchronous collocation toolbox

data.p = length(data.group);
% p is the number of unique sets of mesh

% In this part, async ind is generated. This vector assigns to each
% variable the corresponding mesh group for its discretization.
ind0 = cellfun(@length, data.group);
ind = cell2mat(data.group);
val = arrayfun(@(x,y) repmat(x, [1 y]), 1:length(ind0), ind0, ...
'UniformOutput', false);
val = cell2mat(val);
data.async ind(ind) = val;

% It is possible that mesh groups change as we proceed to next continuation
% step. In this case, the old grouping is passed to the next step through
% async ind0. Otherwise, it is assumed that the same mesh group is going to
% be used again.
if isfield(data,'async ind0')
    if isempty(data.async ind0)
        data.async ind0 = data.async ind;
    end
end

p = data.p;
data.dim = length(data.async ind);
dim = data.dim;

% In the asynchronous collocation toolbox, NTST0 is a vector that contains
% the number of intervals corresponding to the collocation discretization
% of the previous continuation step. The length of this vector is equal to
% the number of mesh groups in the previous continuation step.
% NTST is similar to NTST0 but for the current continuation step.
% NCOL is the number of collocation nodes.
% NTSTs is a vector with length equal to number of variables and contains
% the number of mesh intervals corresponding to each variable.
NTST0 = data.NTST0;
NTST = data.NTST;
NCOL = data.NCOL;
NTSTs = NTST(data.async ind);
data.NTSTs = NTSTs;

if ~isfield(data,'fun ind')
data.fun ind = (1:dim)';
end

% fun ind(i1)=i2 means that the i1th equation takes the mesh associated
% to the i2 variable. async ind(j1)=j2 means that j1th variable takes the
% mesh in group j2.
data.fun ind = data.async ind(data.fun ind);

% kappa is the mesh density. Here, we assume a uniform mesh if kappa is not
% provided.
if isempty(data.kappa)
data.kappa = 1./(2*NTST);
data.kappa = arrayfun(@(y1,y2) repmat(data.kappa(y1), [y2 1]), 1:p, ...
    NTST, 'UniformOutput', false);
data.kappa0 = data.kappa;
end

% tk is the basepoins for Lagrange form representation of the polynomials
% th are collocation nodes.
tk = linspace(-1, 1, NCOL+1);
[th, wt] = coll_nodes(NCOL);

data.th = th(:,);
data.tk = tk;

% In the asynchronous collocation toolbox, the time values at the
% basepoints as well as the collocation nodes are stored and used for
% imposing the collocation conditions.
% Time values at basepoints are sorted for each mesh group and concatenated
% vertically to form tbp. Similarly, tcp contains the time values of
% collocation nodes. The time values of all the variables are stored in bp.
% (unlike tbp which only contains the values for mesh groups).

% The following fields contain the indices of the time values. tbp_idx is a
% cell array and its size is equal to the number of mesh groups. Each
% component is a vector such that the i-th component contains the indices
% of the time values within tbp corresponding the i-th mesh group.
% Similarly, tcp_idx and bp_idx contain the indices of the time values
% corresponding to tcp and bp, respectively.

data.tbp_idx = mat2cell((1:(NCOL+1)*sum(NTST))', NTST*(NCOL+1), 1);
data.tcp_idx = mat2cell((1:NCOL*sum(NTST))', NTST*NCOL, 1);
data.bp_idx = mat2cell((1:(NCOL+1)*sum(NTSTs))', NTSTs*(NCOL+1), 1);

tt = cumsum(2*cell2mat(data.kappa'))

% tt = circshift(tt, 1);

% The indices of the basepoint values at the mesh points (i.e., end points
% of the segments) are stored in segindN0 and segindN for the previous and
% current continuation steps, respectively.
data.segindN0 = (0:sum(NTST0)-1)*(NCOL+1)+1;
data.segindN = (0:sum(NTST)-1)*(NCOL+1)+1;

% segind0s and segind1s are similar to segindN0 and segindN, respectively,
% except that the former fields correspond to values of all variables
% instead of the values of the mesh groups.
data.segind0s = (0:sum(NTSTs)-1)*(NCOL+1)+1;
data.segind1s = (1:sum(NTSTs))*(NCOL+1);

coll_deriv calculates a vector (equal in length to NCOL+1 whose inner
% product with the values at the basepoints evaluates a scalar value
% corresponding to the NCOL-th derivative of the approximating polynomial.

data.deriv = coll_deriv(tk);

% The initial solution is interpolated at the basepoints

if isfield(data, 't0') && ~isempty(data.t0) && (data.Iter==1)
    if data.async==1 && size(data.t0,2)==1
        data.t0 = repmat(data.t0, 1, p);
    end
end
t0 = data.t0;
x0 = data.x0;
T0 = t0(end) - t0(1);
t0 = (t0 - t0(1)) / T0;
x0a = zeros((NCOL+1)*sum(NTST(data.async.ind)), 1);
for i1 = 1:dim
    x0a(data.bp_idx{i1}') = interp1(t0(:,data.async.ind(i1)), ...
        x0(:,i1), data.tbp(data.tbp_idx{data.async.ind(i1)}), 'cubic');
end
data.x0 = x0a;
data.t0 = data.tbp;
else
T0 = data.t0(end) - data.t0(1);
data.t0 = data.t0 / data.t0(end);
data.x0 = data.x0(:);
end

%%
data.fun
ind
inverse gets the 'var ind' and gives the ind of the fun
associated to it. data.fun
ind is the vice versa.

 [~,data.fun
indinverse] = sort(data.fun
ind);

M
async = cell(p,1);
V
async = cell(p,1);

for e1 = 1:dim
    M
async{data.fun
ind(e1),1} = [ M
async{data.fun
ind(e1),1},
    data.async
ind{data.M
async{e1}}];
end
for e1 = 1:p
    M
async{e1} = unique(M
async{e1});
    V
async{e1} = unique(V
async{e1});
end

M
async_group
ind = arrayfun(@(e1) [e1*ones(length(M
async{e1}),1),
    M
async{e1}'], 1:p, 'UniformOutput', false)';
M
async_group
ind = cell2mat(M
async_group
ind);

data.rowSub = M
async_group
ind(:,1);
data.colSub = M
async_group
ind(:,2);
    [data.colSub,colSubind] = sort(data.colSub);
data.rowSub = data.rowSub(colSubind);
data.Pij = sub2ind([p p], data.rowSub, data.colSub);

M
async_f
ind = arrayfun(@(e1) [e1*ones(length(V
async{e1}),1),
    V
async{e1}'], 1:p, 'UniformOutput', false)';
M
async_f
ind = cell2mat(M
async_f
ind);

data.rowi = M
async_f
ind(:,1);
data.vlj = M
async_f
ind(:,2);
data.Wij = sub2ind([p dim], data.rowi, data.vlj);

data.df
ind = reshape(1:sum(data.NTST(data.async
ind)*data.NCOL)*
    data.dim, sum(data.NTST(data.async
ind)*data.NCOL), data.dim);
    data.df
ind = mat2cell(data.df
ind, data.NTST(data.fun
ind)*
    data.NCOL, ones(1,data.dim));
    data.df
ind = cell2mat(data.df
ind(df
ij));

cnt = 0;
funx_num = cumsum([0;histc(data.rowi,1:p)]);
V
async_f
ind = zeros(numeq(df_row,1));
for e1 = 1:dim
    [~,tmp] = intersect(data.M
async_f{e1}',V
async{data.fun
ind(e1)});
    V
async_f
ind(cnt+1:cnt+numel(tmp)) = tmp + funx_num(data.fun
ind(e1));
    cnt = cnt+numel(tmp);
end
%%
data.kappa0 = cell2mat(data.kappa0');

% In the asynchronous collocation case, async_interp is called to form two
% types of interpolating matrices. The first type is used for adaptation,
% i.e., the basepoint values of each variable is interpolated when the mesh
% changes. In the second type, the interpolation is conducted to find the
% values of the approximating polynomials at the collocation nodes
% corresponding to a mesh group different from the associated mesh group of
% a particular variable.

switch data.mesh
case 'async'
    data = async_interp(data);
    x1 = data.W_async.a * data.x0;
end

case 'sync'
    x1 = data.x0;
end

x0 = [x1 ; T0];

pdim = numel(p0);
data.wtt = repmat(wt, [1 sum(NTSTs)]);

data.x_idx = 1:sum(NTSTs)*NCOL+1 + 1;

data.p_idx = sum(NTSTs)*NCOL+1 + 1 + (1:pdim);

data.xbpidx = 1:sum(NTSTs)+NCOL+1;

data.Tidx = sum(NTSTs)*NCOL+1 + 1;

data.drows = (1:sum(NCOL*NTST(data.funind)))';

data.dcols = mat2cell(data.dcols(NCOL*NTST(data.rowi)),1);

data.dcols = data.dcols(1);

data.dcols = cell2mat(data.dcols);

ipidx = setxor(data.segind1s, data.x1idx);
epidx = setxor(data.segind0s, data.x0idx); %or simply ipidx+1

rows = [1:sum(NTSTs-1) 1:sum(NTSTs-1)];

cols = [ipidx epidx];

vals = [ones(1,sum(NTSTs-1)) -ones(1,sum(NTSTs-1))];
data.Q = sparse(rows, cols, vals,sum(NTSTs-1), (NCOL+1)*sum(NTSTs));

% W and Wp are interpolating matrices that are used to find the values of
% approximating polynomials at the collocation nodes using the basepoint
% values

data.W   = coll_L(tk, th);
data.Wp   = coll_Lp(tk, th);
data.Wpend = coll_Lp(tk, 1);
data.kappa0 = data.kappa;
data.async_ind0 = data.async_ind;
function data = async_interp(data)
% (c) Mehdi Saghaei, 2014

NTST = data.NTST;
NTST0 = data.NTST0;
NCOL = data.NCOL;
p = data.p;
NTSt = NTST(data.async)
in;
NTSt0 = NTST0(data.async)
in0;
sNTST = cumsum(NTST);

kappa = cell2mat(data.kappa');

%% Interpolation for each variable with its new kappa
new_groups = [data.async ind0; data.async ind];
[new_groups, ~, ng2] = unique(new_groups, 'rows');

off = 0.1;
% This is just to make sure that a point at t=1 sits in the last bin ...
% of that group!

tbp_p = data.tbp(cell2mat(data.tbp_idx(new_groups(:,2))));
shift_tbp = repel(NTST(new_groups(:,2)) * (NCOL + 1), ... 
{0:1:rownew_groups(:,1) - 1} * (1 + off));

tbp1 = tbp_p + shift_tbp;
segN0 = mat2cell(data.segindN0', NTST0, 1);
segN0 = cell2mat(segN0(new_groups(:, 1)));
edges1 = [{data.t0(segN0, :) + repel(NTST0(new_groups(:, 1)), ... 
{0:1:rownew_groups(:,1) - 1} * (1 + off)); Inf};

[ns, bin] = histc(tbp1, edges1);
k0 = mat2cell(kappa0, NTST0, 1);
k0 = cell2mat(k0(new_groups(:, 1)));

% sigma is vector of local coordinates for each mesh groups corresponding to
% the basepoints of other mesh groups. The local coordinates are [-1,1]
% within each segment of a particular mesh. The actual mesh index of these
% values are stored in ns.
sigma = ((tbp1 - edges1(bin))./k0(bin)) - 1;
sigma = sigma(1:end - 1);
bs = linspace(-1, 1, NCOL + 1)';
L = cell(L(bases, sigma));
L0 = mat2cell(L, NTST0, NCOL + 1);
L = L0(:, 3);
L = cell2mat(L);

rows = repmat({1:1:(NCOL + 1) * sum(NTSTs)}', [1 NCOL + 1]);
cols = reshape({1:1:(NCOL + 1) * sum(NTST0s)}, [NCOL + 1 sum(NTST0s)]');
cols = cols(ns, :);

% W_async.a is the adaptation interpolating matrix
data.W_async.a = sparse(rows, cols, L);

%%
% Similar to the adaptation interpolating matrix, an interpolating matrix is
% also generated for values at collocation nodes from a each mesh to other
% coupled mesh groups
edges2 = [{data.t0(data.segindN, :) + repel(NTST, {0:p - 1} * (1 + off))}; Inf];
tcp2 = repmat(data.tcp, [1 p]);
tcp2 = tcp2 + repmat({0:p - 1} * (1 + off), [sum(NTST) * (NCOL) 1]);
sigma = ((tcp2 - edges2(bin))/kappa(bin)) - 1;
sigma = mat2cell(sigma, NTST*NCOL, ones(1,p));
sigma = sigma(data.Pij);
sigma = cell2mat(sigma);
L = cell_L(bases, sigma);
rows = mat2cell((1:sum(NCOL*NTST))', NCOL*NTST, 1);
rows = rows(data.rowSub);
rows = cell2mat(rows);
rows = repmat(rows, [1,NCOL+1]);
bin = mat2cell(bin, NTST*NCOL, ones(1,p));
bin = cell2mat(bin(data.Pij));
cols = reshape(1:(NCOL+1)*sum(NTST), [NCOL+1 sum(NTST)]');
cols = cols(data.bin(:,:));
P = sparse(rows, cols, L);
Pp = sparse(rows, cols, Lp);
tmp2 = [0,sNTST*NCOL];
tmp3 = [0,sNTST*(NCOL+1)];
[tmp11,~,tmp13] = unique(tmp1);
[tmp21,~,tmp23] = unique(tmp2);
W_async_b = cell(numel(tmp11),numel(tmp21));
for a1 = 1:numel(NTST)
    for a2 = 1:numel(NTST)
        W_async_b{a1,a2} = sparse(tmp11(a1),tmp21(a2));
    end
end
W_async_b = W_async_b(tmp13,tmp23);
W_async_b = W_async_b;
Wind = sub2ind(size(W_async_b), (1:numel(data.rowi))', data.colvj);
data.W_async_b = cell2mat(W_async_b);
data.W_async_b(Wind) = Pn(data.Wij);
data.W_async_b = cell2mat(W_async_b);
data.W_async_b(Wind) = Ppn(data.Wij);
data.W_async_b = cell2mat(data.W_async_b);
function [x, w] = coll_nodes(n)

% (c) Frank Schilder and Harry Dankowicz, reproduced here with permission

% This function finds the Gauss collocation nodes

nn = 1:n-1;
ga = -nn.*sqrt(1./(4.*nn.^2-1));
J = zeros(n,n);
J(sub2ind([n n], nn, nn+1)) = ga;
J(sub2ind([n n], nn+1, nn)) = ga;

[w, x] = eig(J);

x = diag(x);
w = 2*w(1,:)'.^2;

end
function A = coll_L(tk, th)
% (c) Frank Schilder and Harry Dankowicz, reproduced here with permission
q = length(tk);
p = length(th);

ti = reshape(tk, [1 1 q]);
tk = reshape(tk, [1 q 1]);
th = reshape(th, [p 1 1]);

ti = repmat(ti, [p q 1]);
tk = repmat(tk, [p 1 q]);
th = repmat(th, [1 q q]);

tki = tk-ti;

thi = th-ti;

idx = find(abs(tki) <= eps);

thi(idx) = 1;
tki(idx) = 1;

A = thi./tki;
A = prod(A, 3);
A = reshape(A, [p q]);

end
function A = coll_Lp(tk, th)
% (c) Frank Schilder and Harry Dankowicz, reproduced here with permission

q = length(tk);
p = length(th);

j = reshape(tk, [1 1 q]);
j = repmat (j, [p q 1]);

i = reshape(tk, [1 1 1 q]);

k = reshape(tk, [1 q 1 1]);

h = reshape(th, [p 1 1 1]);

i = repmat(i, [p q q 1]);

k = repmat(k, [p 1 q q]);

h = repmat(h, [1 q q q]);

k = tk-ti;
j = tk(:,:,:,1)-tj;

h = th-ti;

id1 = find(abs(k)<eps);
id2 = find(abs(j)<eps);
id3 = find(abs(repmat(tj, [1 1 1 q])-ti)<eps);

k(id1) = 1;

k(id3) = 1;

h(id1) = 1;

h(id3) = 1;

j(id2) = 1;

j = 1.0 ./ j;

j(id2) = 0;

A = h ./ k;

A = prod(A, 4);

A = reshape(A, [p q q]);

A = tkj .* A;

A = sum(A, 3);

A = reshape(A, [p q]);

end
function A = coll_deriv(tk)

% (c) Frank Schilder and Harry Dankowicz, reproduced here with permission

q = length(tk);

ti = reshape(tk, [1 1 q]);

tk = reshape(tk, [1 q 1]);

ti = repmat(ti, [1 q 1]);

tk = repmat(tk, [1 1 q]);

tki = tk-ti;

idx = abs(tki) <= eps;

tki(idx) = 1;

A = 1./tki;

A = prod(A, 3);

end
function [data, y] = adapcoll_F(opts, data, xp)
% (c) Mehdi Sagafi, 2014
x = xp(data.x.idx);
p = xp(data.p.idx);
xbp = x(data.xbpidx);
T = x(data.Tidx);

% xx and yy correspond to the values and the derivates of the variables, % respectively. The interpolating matrices, i.e., W_async_b and Wp_async_b, % have been generated such that when multiplied by the vector of basepoint % values, xbp, result in a large vector of concatenated smaller vectors % each corresponding to the values of the approximating polynomials at points % of a particular mesh. This interpolation has been optimized to perform % the interpolation only when required, i.e., a coupling between the % variables exists. The resulting large vector is then extracted using the % indices stored in Wij. The final form of the values at the collocation % nodes which passed to the zero function, fhan, is a cell array of % vectors. The row of this cell array corresponds to the variable index and % the column corresponds to the mesh group. For example, {i,j} corresponds % to the values of the i-th variable at the collocations nodes associated % to the j-th mesh group.
switch data.mesh
  case 'sync'
    xx = data.W * reshape(xbp, data.NCOL+1, ... 
      sum(data.NTST(data.async_ind)));
    xx = reshape(xx, data.NCOL*sum(data.NTST(data.async_ind)), 1);
    yy = data.Wp * reshape(xbp, data.NCOL+1, ... 
      sum(data.NTST(data.async_ind)));
    yy = reshape(yy, data.NCOL*sum(data.NTST(data.async_ind)), 1);
  case 'async'
    xx = data.W_async_b * xbp;
    yy = data.Wp_async_b * xbp;
end

pp = p;

% The mesh density, kappa, is used in the formulation of the differential % equations. Refer to the appendix. Here, the corresponding kappa values % associated to yy are extracted to form a vector.
  kappa = data.kappa(data.rowi);
  kappa = kappa(:);
  kappa = cell2mat(kappa);
  kappa = kron(kappa, ones(data.NCOL, 1));
  yy = yy / T ./ kappa;
  xx = mat2cell(xx, data.NTST(data.rowi)*data.NCOL, 1);
  yy = mat2cell(yy, data.NTST(data.rowi)*data.NCOL, 1);
  XX = cell(data.p, data.dim);
  YY = cell(data.p, data.dim);
  XX(data.Wij) = xx;
  YY(data.Wij) = yy;

% The zero function in the implicit form of ODEs/DAEs is called
  [data, fode] = data.fhan(XX', YY', pp, data.async_ind, [], data);

% The continuity condition of the piecewise approximating polynomials are % imposed
  fcont = data.Q * x(data.xbpidx);
  y = [ fode ; fcont ];
end
function [data, J] = adapcollDFDX(opts, data, xp)
% (c) Mehdi Saghafi, 2014

x = xp(data.x_idx);
p = xp(data.p_idx);
xbp = x(data.xbp_idx);
T = x(data.T_idx);

% Function initiate performs some preamble calculations to find the
% interpolated values of the approximating polynomials as well as their
% corresponding derivatives
[data, xx, yy, pp] = initiate(data, xbp, p, T);

% This function evaluates the Jacobian of the zero function. dfode1 and
% dfode2 are the Jacobians with respect to the variables and thier
% derivatives, respectively.
[data, dfode1, dfode2] = data.dfdxhan(xx, yy, pp, data.async_ind, [], data);

kappa2 = cell2mat(data.kappa(data.fun_ind)');
kappa2 = kron(kappa2, ones(data.NCOL,1));
kappa2 = repmat(kappa2, [1 data.dim]);
dfode2t = dfode2;
dfode2 = dfode2 ./ kappa2;
dfode1 = dfode1(data.df_ind);
dfode2 = dfode2(data.df_ind);
dfode2t = dfode2t(data.df_ind);
dfode1 = sparse(data.dxrows, data.dxcols, dfode1);
dfode2 = sparse(data.dxrows, data.dxcols, dfode2);
dfode2t = sparse(data.dxrows, data.dxcols, dfode2t);

yyt = cell2mat(reshape(yy, [data.dim * data.p 1]));
[rows, cols, vals] = find(dfode);

% Differentiation wrt T is evaluated
fode = - (dfode2t * yyt) ./ T;

r = (1:sum(data.NTST(data.async_ind))*data.NCOL)';
c = repmat((data.NCOL+1)*sum(data.NTST(data.async_ind))+1, ...
    [sum(data.NTST(data.async_ind))*data.NCOL,1]);
rows = [rows ; r];
cols = [cols ; c];
vals = [vals ; fode];
off = sum(data.NTST(data.async_ind))*data.NCOL;

% Jacobian of the continuity equations is evaluated
[r, c, v] = find(data.Q);
rows = [rows ; off + r];
cols = [cols ; c];
vals = [vals ; v];

J1 = sparse(rows, cols, vals);

% This function evaluates the Jacobian of the zero function wrt to the
% parameters
[data, dfode] = data.dfdphan(xx, yy, pp, data.async_ind, [], data);
dfcont = sparse(size(data.Q,1), numel(p));
J2 = [ dfode ; dfcont ];
J = sparse([J1 J2]);
end
function [data, XX, YY, pp] = initiate(data, xbp, p, T)
% (c) Mehdi Saghafi, 2014

% This function prepares the values of the approximating polynomials as well as their corresponding derivatives at the desired collocation nodes through interpolation

switch data.mesh
  case 'sync'
    xx = data.W * reshape(xbp, data.NCOL+1, sum(data.NTSTs));
    xx = reshape(xx, data.NCOL*sum(data.NTSTs), 1);
    xx = mat2cell(xx, data.NCOL*data.NTSTs, 1);
    xx = xx(data.rowSub);
    xx = cell2mat(xx);
    yy = data.Wp * reshape(xbp, data.NCOL+1, sum(data.NTSTs));
    yy = reshape(yy, data.NCOL*sum(data.NTSTs), 1);
    yy = mat2cell(yy, data.NCOL*data.NTSTs, 1);
    yy = yy(data.rowSub);
    yy = cell2mat(yy);
  case 'async'
    xx = data.W async b * xbp;
    yy = data.Wp async b * xbp;
end

pp = p;

kappa1 = data.kappa(1, data.rowi)';
kappa1 = cell2mat(kappa1);
kappa1 = kron(kappa1, ones(data.NCOL, 1));
yy = yy / T ./ kappa1;

xx = mat2cell(xx, data.NTST(data.rowi)*data.NCOL, 1);
yy = mat2cell(yy, data.NTST(data.rowi)*data.NCOL, 1);
XX = cell(data.p, data.dim);
YY = cell(data.p, data.dim);
XX(data.Wij) = xx;
YY(data.Wij) = yy;
XX = XX';
YY = YY';

end
function opts = adapcoll_create(opts, data, x0, p0)
% (c) Mehdi Saghafi, 2014
% This function imposes the collocation conditions to the corresponding
% zero function
fid = coco_get_id(data.prefix, 'adapcoll_fun');
opts = coco_add_func(opts, fid, @adapcoll_F, @adapcoll_DFDX, data, ...
'zero', 'x0', [x0; p0]);
idx = coco_get_func_data(opts, fid, 'idx');
data.idx = idx;

fid = coco_get_id(data.prefix, 'reduced_adapcoll_save');
opts = coco_add_slot(opts, fid, @coco_save_data, [], 'save_reduced');
fid = coco_get_id(data.prefix, 'adapcoll_save');
opts = coco_add_slot(opts, fid, @coco_save_data, data, 'save_full');
end
function outarg = repel(reps, vals)
% (c) Mehdi Saghafi, 2014

reps = reps(:)';
vals = vals(:)';
tmp1 = repmat((1:max(reps))', [1 length(reps)]);
tmp1 = tmp1 - repmat(reps, [size(tmp1,1) 1]);
outarg = repmat(vals, [size(tmp1,1) 1]);
outarg = outarg(tmp1 < 0);
outarg = reshape(outarg, sum(reps), 1);
end
REFERENCES


[79] W. Zhang, R. Baskaran, and K. Turner, “Effect of cubic nonlinearity on auto-
parametrically amplified resonant MEMS mass sensor,” Sensors and Actuators A: Phys-

[80] V. Kumar, J. Boley, Y. Yang, H. Ekowaluyo, J. Miller, G. Chiu, and J. Rhoads,
“Bifurcation-based mass sensing using piezoelectrically-actuated microcantilevers,” Ap-

2004.

chanical duffing resonator via stochastic resonance,” Applied Physics Letters, vol. 90,

and nano-resonators,” Journal of dynamic systems, measurement, and control, vol. 132,

[84] H. Postma, I. Kozinsky, A. Husain, and M. Roukes, “Dynamic range of nanotube-and
223105, 2005.

[85] M. Agarwal, S. Chandorkar, R. Candler, B. Kim, M. Hopcroft, R. Melamud, C. Jha,

nonlinear stiffness coefficients [actuators],” Microelectromechanical Systems, Journal of,


range, and frequency of nanomechanical resonators,” Applied Physics Letters, vol. 88,

of nonlinear nanomechanical resonant cantilevers for highly sensitive nems gas/mass
sensor applications,” Journal of Micromechanics and Microengineering, vol. 20, no. 4,

[90] L. Shao, M. Palaniapan, and W. Tan, “The nonlinearity cancellation phenomenon in
micromechanical resonators,” Journal of Micromechanics and Microengineering, vol. 18,
no. 6, p. 065014, 2008.


